VILNIUS UNIVERSITY

Jurij Tekutov

MODEL OF DOMAIN KNOWLEDGE CONTENT UPDATING BASED ON MANAGEMENT INFORMATION INTERACTIONS

Summary of Doctoral Dissertation

Physical Sciences, Informatics (09 P)

Vilnius, 2013

Doctoral dissertation was prepared at the Vilnius University of Kaunas Faculty of Humanities in 2008–2012.

Scientific Supervisor:
Prof Dr Saulius Gudas (Vilnius University, Physical Sciences, Informatics – 09 P)

Scientific Consultant:
Prof Dr Vitalijus Denisovas (Klaipeda University, Physical Sciences, Informatics – 09 P)

The dissertation will be defended at the Council of Scientific Field of Informatics at Kaunas Faculty of Humanities of Vilnius University:

Chairman:
Prof. Habil. Dr. Gintautas Dzemyda (Vilnius University, Physical Sciences, Informatics – 09 P)

Members:
Prof. Dr. Rimantas Butleris (Kaunas University of Technology, Physical Sciences, Informatics – 09 P)
Assoc. Prof. Dr. Audrius Lopata (Vilnius University, Physical Sciences, Informatics – 09 P)
Prof. Habil. Dr. Edmundas Kazimieras Zavadskas (Vilnius Gediminas Technical University, Physical Sciences, Informatics – 09 P)
Prof. Habil. Dr. Rimvydas Simutis (Vilnius University, Physical Sciences, Informatics – 09 P)

Opponents:
Prof. Habil. Dr. Genadijus Kulvietis (Vilnius Gediminas Technical University, Physical Sciences, Informatics – 09 P)
Prof. Dr. Lina Nemuraitė (Kaunas University of Technology, Physical Sciences, Informatics – 09 P)

The dissertation will be defended at the public meeting of the Council of Scientific of Informatics in the auditorium number 10 at the Kaunas Faculty of Humanities of Vilnius University, at 4 p. m. on 26 June 2013.
Address: Muitinės St. 12, LT-44280 Kaunas, Lithuania.
The summary of the doctoral dissertation was distributed on 24 May 2013.
A copy of the doctoral dissertation is available for review at the Library of Vilnius University (Universiteto St. 3, LT-01122Vilnius, Lietuva).

VILNIAUS UNIVERSITETAS

Jurij Tekutov

DALYKINĖS SRITIES ŽINIŲ TURINIO ATNAUJINIMO VALDYMO INFORMACINIŲ SĄVEIKŲ PAGRINDU MODELIS

Daktaro disertacijos santrauka

Fiziniai mokslai, Informatika (09 P)

Vilnius, 2013

Disertacija rengta 2008–2012 metais Vilniaus universitete

Mokslinis vadovas:

prof. dr. Saulius Gudas (Vilniaus universitetas, fiziniai mokslai, Informatika – 09 P)

Konsultantas:

prof. dr. Vitalijus Denisovas (Klaipėdos universitetas, fiziniai mokslai, Informatika – 09 P)

Disertacija ginama Vilniaus universiteto Informatikos mokslo krypties taryboje:

Pirmininkas:

prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, fiziniai mokslai, Informatika – 09 P)

Nariai:

prof. dr. Rimantas Butleris (Kauno technologijos universitetas, fiziniai mokslai, Informatika – 09 P)

doc. dr. Audrius Lopata (Vilniaus universitetas, fiziniai mokslai, Informatika – 09 P)

prof. habil. dr. Edmundas Kazimieras Zavadskas (Vilniaus Gedimino technikos universitetas, fiziniai mokslai, Informatika – 09 P)

prof. habil. dr. Rimvydas Simutis (Vilniaus universitetas, fiziniai mokslai, Informatika – 09 P)

Oponentai:

prof. habil. dr. Genadijus Kulvietis (Vilniaus Gedimino technikos universitetas, fiziniai mokslai, Informatika – 09 P)

prof. dr. Lina Nemuraitė (Kauno technologijos universitetas, fiziniai mokslai, Informatika – 09 P)

Disertacija bus ginama viešame Informatikos mokslo krypties tarybos posėdyje 2013 m. birželio mėn. 26 d. 16 val. Vilniaus universiteto Kauno humanitarinio fakulteto 10 auditorijoje.
Adresas: Muitinės g. 12, LT-44280, Kaunas, Lietuva.
Disertacijos santrauka išsiuntinėta 2013 m. gegužės mėn. 24 d.
Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje (Universiteto g. 3, LT-01122 Vilnius, Lietuva).

INTRODUCTION

Research area and relevance of the problem. As of late, information systems (IS) development life-cycles are analysed using different approaches. One of the most promising is model-driven approach, which aims to utilize formal domain knowledge to perform, adjust and manage and engineering process. One of the earliest stages of IS development life-cycle is enterprise modelling, which becomes especially important in applying model-driven IS development methodology (MDE – Model Driven Engineering) and its methods. The aim of model enterprise modelling is to create an enterprise model of an organization, which would help to deal with issues of an analysed domain. The concept of Model Driven Architecture (MDA) defines IS development process as a three-level model (CIM – the Computation Independent Model, PIM – the Platform Independent Model, PSM – the Platform Specific Model) interaction, whose purpose is to directly link enterprise modelling and further stages of IS analysis, design and implementation. Thus, from enterprise modelling professionals' point of view MDA concept should be thought of as an effort to integrate enterprise modelling methods with full life-cycle Computer-Aided Systems Engineering (CASE) technologies characterized by the bottom-up approach, i.e. from engineering methods to enterprise modelling methods for IS engineering needs. Hence, modern IS development methodologies pay increasingly more attention not only (and not as much) to software and IS engineering methods, but rather organizations' enterprise modelling issues, i.e. domain knowledge mapping.

IS engineering methods (IS life-cycle stage models) based on intra-enterprise engineering, the basis of which is subject area (domain) pattern model, are called knowledge-based methods. The purpose of MDE-based computerized IS engineering enterprise model is to systematize and formalize knowledge on a certain analysed domain (subject area). IS engineering methods are emerging that seek to integrate knowledge about operational functions, processes, rules in CASE systems and use them in intellectualizing IS development process. There have also been attempts to apply domain knowledge content creation and updating methods from similar areas, such as e-learning, Business Intelligence systems, business rule modelling.

The creation of knowledge content specification includes the collection of domain knowledge, its preparation and analysis, verification and validation. In this dissertation domain knowledge is understood as a dynamic set of knowledge attributes, described according to requirements as a system of facts, principles and theories related to a specific professional activity. In other words, knowledge attribute set is an expression of a domain in current context. Enterprise modelling can be assigned to requirement collection and analysis stage. Then enterprise models are transformed into requirement specification for the system being developed. Systems (for example, domain knowledge content) and separate component specifications are produced during IS engineering process. The most important step in every effective domain knowledge management process is the writing down of the requirements identified during the selection. It helps to ensure accurate communication and management of the requirements during their development. Documents are an understandable form for requirement recording. These documents provide a basis for context identification and requirement collection, yet are not very useful in supplementing and modifying requirement information.

Meeting modern, constantly evolving enterprise needs requires enterprise integration with information technologies and computerized IS that would be able to adapt to organizations' enterprise changes. The development of these systems should be performed by focusing on Value Chain Model (VCM), since computerization involves enterprise management functions and processes, as well as interactions between organization's functional areas, which are revealed by the VCM. In computerizing enterprise, a research of informational interactions between enterprise management functions and processes is performed, requiring further decomposition of value chain elements, i.e. further VCM modification is performed, a detailed Value Chain Model (DVCM) is created. Hence, knowledge-based IS development and performed based on knowledge on enterprise management, which defines fundamental attributes of enterprise management information interactions. Enterprise management information interaction is specified as an Elementary Management Cycle (EMC), linking enterprise management function and process by a feedback loop.

This dissertation provides a proposed domain management knowledge modelling method, whose formal DVCM and EMC descriptions are provided in a BPMN notation. At the same time, it is not limited to a formal representation; the method is elaborated and demonstrated in a specific domain – higher education study programme design and updating. This domain has been chosen intentionally, on the basis of methodological terms. In academic field IS development typically starts with enterprise management function analysis, in which enterprise management information interactions are important. Analysed higher education study domain is multivariate, dynamic and requires constant knowledge content updating. Only by actualizing study programme structure and knowledge content in a timely manner is it possible to properly take into account changing needs and requirements from all stakeholders.

One of the pioneers of analysing study programme development as an IS engineering process were MOCURIS – Modern Curriculum Development in Information Systems at Master Level project promoters, who offered in their works a model study programme design method. They have introduced the notion of study programme engineering, defined study programme architecture and by analyzing a study programme as a system of courses and modules, they have applied the standard IS engineering process to develop a Master‘s study programme. Using this method, a large IS-related Master's programme specification document was created, which would act as a guideline for creators of certain programmes, courses and modules. However, already in the project fundamental flaws of this method became evident. Such programme development process could only be effectively implemented by the authors, meanwhile, the resultant artefact (textual specifications document) because of its technicality, complexity and size was unsuitable to perform the main function of requirement specification which is to ensure communication of all process participants and coordinate their work. After analysing requirement identification, localization and specification (granularity) issues, it was found that these limitations are inherent in traditional non-automated requirement engineering processes. Therefore, other researchers (including the author of this thesis) have suggested to automate this process in creating a study programme requirement engineering system on the basis of CASE measures. With a direct involvement of the author, such a system has been created and applied for the development of new study programmes. However, in a fast changing environment timely study programme modernization, quality assurance and knowledge content updating become especially relevant. It all requires effective feedback from all stakeholders. To solve the issues mentioned in this dissertation, a detailed Value Chain Model has been used, ensuring constant information feedback, which sends descriptive data (attributes) to the enterprise management function and retrieves management decisions to direct enterprise process. The method is implemented by creating an enterprise knowledge database in IS engineering CASE system, which is designed on the basis of a computerized domain enterprise model.

Scientific significance of the work. In traditional computerized IS engineering a system is created empirically, by first collecting, analysing and specifying consumer needs. In knowledge-based computerized IS engineering a system is created using an enterprise knowledge base and domain knowledge stored in it. It is believed that the main characteristic of knowledge-based IS engineering is a subsystem of domain knowledge, which is based on an organization's enterprise model and is designed to acquire knowledge on a computerized domain. Such knowledge acquisition subsystem should perform the functions of a storage of domain knowledge, required for the creation of IS project models. The following structure for a subsystem was proposed: enterprise model and enterprise metamodel (formalized enterprise management knowledge structure). Domain knowledge acquisition subsystem should become the essential component of a knowledge-based CASE measure, intellectualizing the whole process of IS development. The creation of such subsystem is one of the most urgent scientific problems addressed in this dissertation.

Another urgent problem – the evaluation of management information interactions in modelling organization's enterprise. Quite a few enterprise modelling methods and languages (DFD, IDEFx, UML extensions, BPMN, DVCM and others) have been proposed, however, only some of them measure domain management information interactions that are necessary to manage operations from a system theory perspective. The creation and updating of a domain model that includes data, knowledge and objective components necessary to manage domain operations is a relevant scientific issue. The identification and updating of the content of computerized domain information interactions in the areas of IS development and BI measures and knowledge management is and unformalized process that is difficult to formalize in general. Currently, by applying computerized measures for IS engineering, formalized methods of creating and updating content of domain management information interactions are still developing.

The dissertation provides a method for updating the knowledge method of the analyzed domain, hereafter referred to as "enterprise domain", based on enterprise modelling in terms of management information interactions when management information interactions between domain components are identified. The model of domain knowledge content updating is formally described and computerized process measures are proposed. The method is applied by creating an enterprise domain identification and updating CASE system that is approved by developing (updating) for higher education Bachelor's and Master's informatics study programmes.

Research subject. The identification, modelling and updating of domain knowledge content that encompasses enterprise domain management information interactions.

The Aim and Tasks. To develop domain knowledge modelling method based on enterprise management information interactions and to computerize domain knowledge model updating process.

To achieve the aim, it was necessary to solve the following tasks:

1. To perform the analysis of current domain modelling methods in terms of knowledge content modelling.
2. To develop domain knowledge modelling method based on management information interactions:
   2.1. to define the sequence of enterprise management functional relationships, comprising information feedback;
   2.2. to apply a modified Value Chain and Elementary Management Cycle model to describe and update domain knowledge content.
3. To develop domain management knowledge content identification and updating computerized process model and information system prototype.
4. To apply knowledge content identification and updating method to study programme requirement analysis and programme content updating, and to propose system application methodology.
5. To perform an experimental research of the designed method as well as the application of CASE measures, evaluating the quality of updated knowledge.

Scientific Novelty. The following results were derived:

- designed subject area (domain) management knowledge modelling method based on enterprise management information interactions;
- model of domain management knowledge content identification and knowledge updating computerized process, designed on the basis of a modified Value Chain and Elementary Management Cycle;
- in implementing this model and its algorithms, a subsystem of enterprise knowledge has been created in a knowledge-based CASE system, performing the function of a domain knowledge database;
- designed method and system were approved in a selected domain – higher education study programme creation and updating.

Personal contribution. Essential contributions of the author are:

a) definition of the sequence of management functional relationships, comprising informational feedback in knowledge modelling and management area;
b) a modified Value Chain and Elementary Management Cycle model was created to describe management knowledge updating process;
c) a computerized process model and knowledge-based CASE system were created.

Practical Significance. The following results of practical value were achieved:

- a computerized management knowledge measure was created that can be applied not only in chosen subject area updating, but other areas as well, such as: the creation and updating of business knowledge bases, the development of BI measures by implementing knowledge bases, the evaluation of the fullness of accumulated repositories' content;
- the method and the computerized measure were used in the examination and updating of study programme knowledge content;
- "study programme development / implementation management", "content knowledge model structuring management", "domain knowledge model formation management" and "knowledge model updating management" information DVCM models were created. The results obtained can be used in

subsequent studies of the domain process management and knowledge content refinement models;

- the results were used in developing and implementing study programme requirement analysis system in the project “Modernisation, development and assurance of mobility of Master study programmes in information technology area” (SFMIS code: BPD2004-ESF-2.5.0-03-05/0063, support contract No. ESF/2004/2.5.0-03-407/BPD-178);
- system application methodology was proposed, that allows to systematize, analyse, localize and evaluate requirements for higher education study programmes, modernize the structure as well as update knowledge content of these studies;
- informatics study programme structure and knowledge content were updated on the basis of a scientific method.

Hypotheses:

1. A method was designed, allowing to develop a two-level (granular) model for describing knowledge of domain management information interactions.
2. Proposed hierarchical modified Value Chain Model and Elementary Management Cycle model allows to structure domain knowledge identification and updating processes.
3. Computerized model of knowledge updating allows to develop a knowledge-based CASE system, which can be applied for updating quickly evolving domain (not only study programme) knowledge content.
4. The developed methodology for the application of study programme requirement analysis and knowledge content updating system enables to systematize, analyse, localize and evaluate requirements for developed higher education study programmes and flexible update their structure and knowledge content.

Research methodology. Theoretical and experimental research methods are applied in the dissertation. The investigation of domain knowledge content analysis includes a systematic analysis of scientific literature, the summarizing of the methods reviewed in enterprise modelling and knowledge management modelling areas, traditional enterprise modelling methods (the analysis of the properties of DFD, IDEF0, IDEF3, BPMN and other methods), the application of qualitative analysis (conceptual maps), knowledge based modelling methods and the application of comparative analysis and summarizing methods. For theoretical and experimental researches CASE measures (*IBM Rational RequisitePro™*, *MagicDraw™*, *MS Visio™*) and other computerized software was used: concept map tool (*Mindjet MindManager Pro ™*), database management systems (*MS Access ™*, *MySQL Server ™*).

Approbation of the research results. The main results of the dissertation were published in 12 scientific papers: 1 in a periodical scientific journal (ISI Web of Science), 1 in ISI indexed proceedings, 5 articles in the periodical scientific publications; 1 article in the reviewed scientific periodical publications, 1 article in the other International Scientific Conference Proceedings, 1 these published in international scientific conference publications, 2 articles in local conference proceedings. The main results of the work have been presented and discussed at 7 national and international conferences.

Dissertation scope and structure. The dissertation is written in Lithuanian. The dissertation consists of abbreviations and terms glossaries, preface, five chapters, findings, literature reference list, publication list and 16 annexes. Chapters are divided into parts, whereas, some parts have their own sections.

The *introductory section* discusses the research area and relevance of the problem, scientific significance of the work, also defines the research subject, formulates the research goal and objectives, provides scientific novelty of the dissertation, personal contribution, practical value of the research findings, and hypotheses, as well as describes research methodology. At the end of the introduction is provided the list of the author‘s publications and conference reports on the dissertation topic and dissertation structure.

First chapter reviews domain modelling methods in terms of knowledge content modelling. *Second chapter* introduces the created domain management knowledge modelling method, based on enterprise management information interactions. The application of a modified Value Chain and Elementary Management Cycle model for knowledge content description and updating is demonstrated. The created domain management knowledge content identification and knowledge updating computerized process model as well as information system prototype are laid out in *chapter three*. *Chapter four* introduces study programme requirement analysis and knowledge content updating system and its application methodology. *Chapter five* provides the approbation of the created domain management knowledge modelling method. The results from experimental research confirm method appropriateness. A summary and findings are formulated at the end of each chapter. At the end of the dissertation general findings and annexes are provided.

Volume of the dissertation is 177 pages (A4 format). Text of the dissertation has 13 tables and 105 figures. In the dissertation 214 refferences to other sources were used.

1. THE ANALYSIS OF CURRENT DOMAIN MODELLING METHODS IN TERMS OF KNOWLEDGE CONTENT MODELLING

Due to the fact that there have been many new enterprise modelling approaches, methodologies and methods created, a separate research area has occurred, called enterprise modelling.

Enterprise models are divided into the following: business process models, data flow models, data models, business objective models, control flow models, performance management process models, knowledge management processes. Widely known enterprise modelling methods are structural-functional modelling (Data Flow Diagrams or DFD), business process modelling languages, BPMN notation, IDEF set of standards and organization enterprise modelling language UEML, and business architecture modelling language UPDM. Well known enterprise modelling methodologies are CIMOSA, GERA, EPC, DODAF, MoDAF, MDA, standards (ISO 14258, ISO 15704). However, amongst this enterprise modelling approaches, methodologies and methods one can rarely find those that would model enterprise in terms of management, i.e. performance management information components and their interaction.

Therefore, one has to select the most convenient way of modelling management activities. In order to achieve the goal, similar models were analyzed and a thorough comparison was performed. Based on the domain modelling method (language, notation) analysis, a conceptual map was designed allowing to systematize the variety of these methods (Fig. 1).

When comparing modern enterprise modelling methods (languages, notations), it is becoming clearer that organization enterprise modelling needs to have capable methodologies because it has to be ensured that all necessary structures can be modelled. For this purpose, the above mentioned conceptual map can be used, that shows primary and systematized information and provides references to all other required sources of domain modelling methodologies. When analysing known enterprise modelling methods (notations, languages) in terms of performance management, it is becoming clearer that only a few methods can actually evaluate performance management information interactions that are necessary to manage performance from a theoretical perspective.

In organizational management practice *Value Chain Model* (herein referred to as VCM), created by Porter, is popular and has been used in many works. The structured VCM is used to identify the information transactions between *Management Functions* and *Enterprise Processes*. Detailed VCM embodies a procedural approach to enterprise consisting of: primary activities (operational processes) and support activities (operational functions). Formal detailed VCM description:

$$VCM = \{(F1,\ldots,Fi);(P1,\ldots,Pj)\}, \quad (1)$$

where: $(F1,\ldots Fi)$ – enterprise Management Functions, $(P1,\ldots Pj)$– Enterprise Processes.

The interrelationship between primary and secondary business processes explored in Gudas, Lopata identified a different nature of these 2 enterprise activities: secondary processes possess informational nature and are referred to as enterprise (management) *Functions*; while primary processes are concrete (non-informational) and are named enterprise (material) *Processes*. This dissertation presents more detailed content of *Function Fi* since it defines a sequence of definite types of interacting information activities directed to control *Process Pi*.

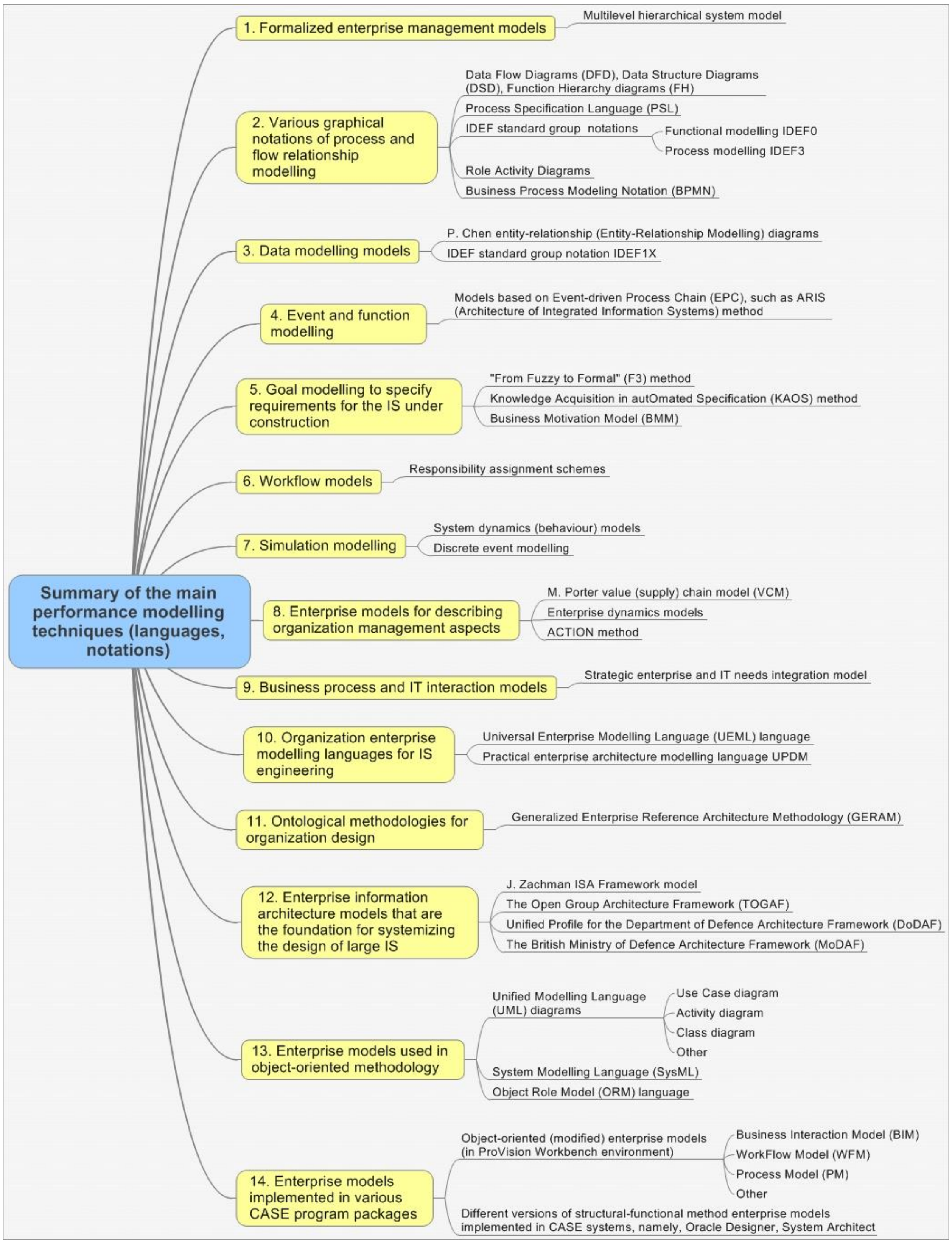


Fig. 1. Conceptual map of domain modelling methods (languages, notations)

In brief, the concept of *Elementary Management Cycle* (EMC) is a formalized description of the Enterprise management control as the interaction between the *Function* and the *Process* – as two core components of enterprise from the control point of view. On the basis of these findings, the *Value Chain Model* is *modified*. The interaction between the core elements a *Function* and a *Process* is formally assumed as a *Control Process* with the *Feedback Loop* between the *Function Fi* and the *Process Pj* (Fig. 2). A detailed model of the informational interactions between the *Function Fi* and *Process Pj (Fi x Pj)* can be defined by:

$$EMC(Fi,Pj) = (Pj(A,G) \rightarrow IN(A,B,G) \rightarrow DP(B,C,G) \rightarrow DM(C,D,G) \rightarrow RE(D,V,G) \rightarrow Pj(V,G)) \quad (2)$$

Where *A* – state attributes of the process *Pj*, needed in terms of *G*; *B* – systemized (interpreted) primary data, needed for the enterprise management function *Fi* in terms of *G*; *C* – enterprise data, formed by data processing *DP*, prepared for decision making *DM* and needed in terms of *G*; *D* – goal-congruent management decision formed by the decision making process *DM*; *V* – goal-congruent effects of management on the process *Pj*, formed by the realization process *RE*.

The *Elementary Management Cycle* (EMC) is the basic construct of *Enterprise Management* modelling, it refines the components of management (control) cycle as well as content of management information transformations. The interaction between different layers is considered as the control loop (information feedback) and it was formally described by Gudas as EMC *(Elementary Management Cycle)*. The mandatory steps (*Interpretation – IN, Data Processing – DP, Decision Making – DM, Realisation of Decision – RE*) of the EMC are defined as information transferring processes focused on the control of the content of the *Management Functions F*.

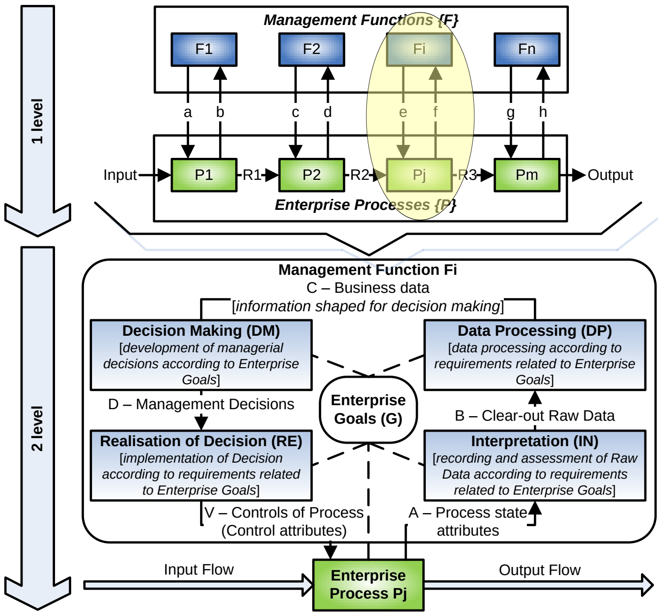


Fig. 2. The formalized model of the interaction between the *Function Fi* and *Process Pj* (Elementary Management Cycle – EMC) in *modified* (structured) VCM

The structure of an EMC is provided in Table 1.

Table 1. The composition of an Elementary Management Cycle (EMC)

EMC component	Description
Management *Function Fi*.	*Fi* identifies a specific management *Function* defined in EMC. Each element of a specific *EMC (i)* is defined in terms of this management *Function Fi*.
Pj (A, G), Pj (V, G) – technological *Process* (managed object).	Technological *Process*, its input (*I*) and output *(O)* are *Material Flows*. *Material Flows* are defined by state attributes of a specific *Process Pj*, which are necessary to perform a specific enterprise management *Function Fi* in combination with enterprise *Goals* (*G*).
IN (A, B, G) – *Interpretation* of the state of the managed object.	*Interpretation* (*IN*) is performed according to enterprise rules that depend on enterprise *Goals* (*G*). *IN* rules are a subset of organization‘s enterprise rules that forms systemized primary data *B*.
DP (B, C, G) – Data Processing.	*Data Processing* (*DP*) is the solution of enterprise problems and data processing according to enterprise rules, which depend on enterprise *Goals* (*G*). *DP* prepares data for the next EMC stage –

EMC component	Description
	process *DM*.
DM (C, D, G) – management *Decision Making*.	When making management decisions (*DM*), it is sought to fulfil enterprise goals and the requirements for controlling managed process *Pj* captured in them. *Decision Making DM* is carried out according to the rules that depend on enterprise *Goals* (*G*). *DM* output is a specific management decision intended to control the process *Pj*.
RE (D, V, G) – management *Decision Realisation*.	Management decisions are carried out according to *G* requirements (rules). Management *Decisions Realisation* (*RE*) output is the effects of management on process *Pj*, corresponding to the state attributes *A* of a particular process *Pj* and enterprise *Goals G*.
G – Enterprise *Goal* (subgoal).	Each enterprise *Goal* (subgoal) plans enterprise output requiring the implementation of a particular EMC in order to be achieved. Each subgoal is linked to the elements of a particular EMC.

Management Functional Dependency (MFD) – is internal reason for management information transformations which are implemented by management *Functions*. *Management Functional Dependence* (MFD) consists of sequence of *Functional Dependencies* (FD) of EMC stages:

$$MFD = \{FD1,\quad FD2,\quad FD3,\quad FD4,\quad FD5\}, \tag{3}$$

where:

- $FD1 = (Si \rightarrow IN(G) \rightarrow Sj)$ – *Functional Dependence* of the *Interpretation* step, Si – *Business Process* (technological process) state attributes ($Si = A$), Sj – output attributes of the *Interpretation* step ($Si = B$);
- $FD2 = (Sj \rightarrow DP(G) \rightarrow Sn)$ – *Functional Dependence* of the *Data Processing* step, Si – output attributes of the *Interpretation* step are an input ($Sj = B$), Sn – *Data Processing* output ($Sn = C$);
- $FD3 = (Sn \rightarrow DM(G) \rightarrow Sm)$ – *Functional Dependence* of the *Decision Making* step, Sn – output attributes of the *Data Processing* step are an input of *Decision Making* ($Sn = C$), Sm – *Decision Making* output ($Sm = D$);
- $FD4 = (Sm \rightarrow RE(G) \rightarrow Sk)$ – *Functional Dependence* of the *Decision Realisation* step, Sm – output attributes of the *Decision Making* step are an input of *Decision Realisation* ($Sm = D$), Sk – *Decision Realisation* output ($Sk = V$);
- $FD5 = (Sk \rightarrow P(G) \rightarrow Si)$ – *Business Process* (*technological process*) implementation step, modelled as a *Functional Dependence,* Sk – *Decision Realisation* step output, that is a managing effect for business (technological) process ($Sk = V$), Si – *Business Process* (*technological process*) state attributes ($Si = A$).

Accordingly, enterprise *Management Functional Dependence* (MFD) is a sequence of EMC elements‘ *Functional Dependencies* $FD1$, $FD2$, $FD3$, $FD4$ and $FD5$, that forms an information feedback loop which defines integrity requirements of EMC steps:

$$MFD = \{FD1 = (Si \rightarrow IN(G) \rightarrow Sj); FD2 = (Sj \rightarrow DP(G) \rightarrow Sn);$$
$$FD3 = (Sn \rightarrow DM(G) \rightarrow Sm);\ \ FD4 = (Sm \rightarrow RE(G) \rightarrow Sk); FD5 = (Sk \rightarrow P(G) \rightarrow Si)\} \tag{4}$$

Based on this general model, the author has proposed a specialised model for particular problem domain.

2. DOMAIN MANAGEMENT KNOWLEDGE MODELLING METHOD, BASED ON ENTERPRISE MANAGEMENT INFORMATION INTERACTIONS

Domain management knowledge modelling method, based on enterprise management information interactions, was created in order to identify the discrepancy between the existing knowledge model and the actual domain knowledge model. The resulting method is designed to create a two-level (granular) knowledge description model for the analyzed domain management information interactions, to form current knowledge and domain knowledge models and to perform the analysis of knowledge models. The two levels of knowledge description are:

a) first level of detail – DVCM that identifies information interactions;
b) second level of detail – a detailed model of each management information interaction by applying EMC.

BPMN notation – a principal scheme of domain management knowledge modelling method is provided in Figure 3. *MagicDraw*^{TM} software package was selected for its creation.

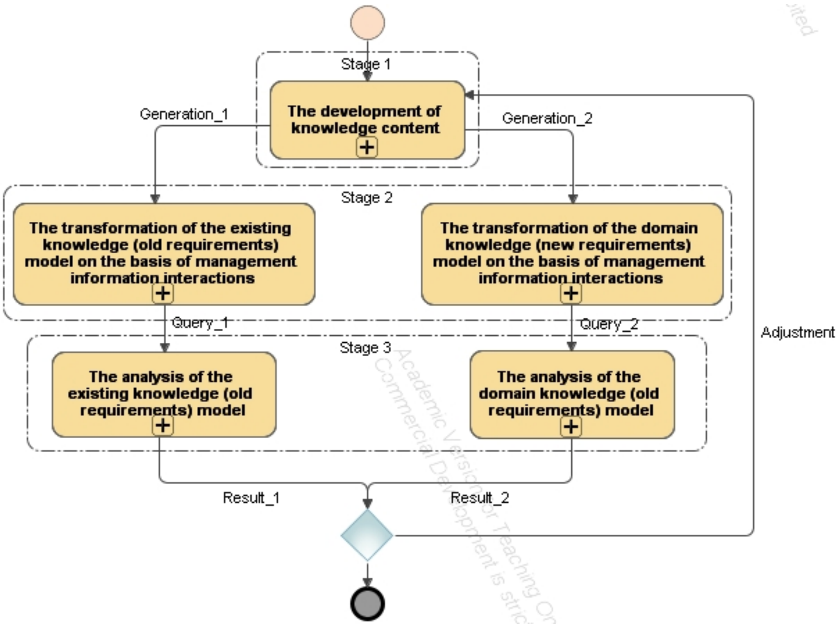


Fig. 3. A principal scheme of domain management knowledge modelling method (BPMN notation)

First stage includes one step:

- the development of knowledge content.

Second stage consists of two steps:

- the transformation of the existing knowledge (old requirements) model on the basis of management information interactions;
- the transformation of the domain knowledge (new requirements) model on the basis of management information interactions.

Third stage includes two steps:

- the analysis of the existing knowledge (old requirements) model;
- the analysis of the domain knowledge (old requirements) model.

The development of knowledge content (Fig. 4) involves structural, functional, quality and other requirements.

In terms of the current domain (for example, programme structure), the smallest structural component is a study module. Module composition is described in terms of attributes. To establish a system hierarchy, modules are combined into study blocks (subject groups). Main blocks are defined in the general requirements for study programmes. Functional (study content) requirements are defined by the purpose, stage and objectives of a study programme that are defined in regulatory documents and derived from other requirement sources.

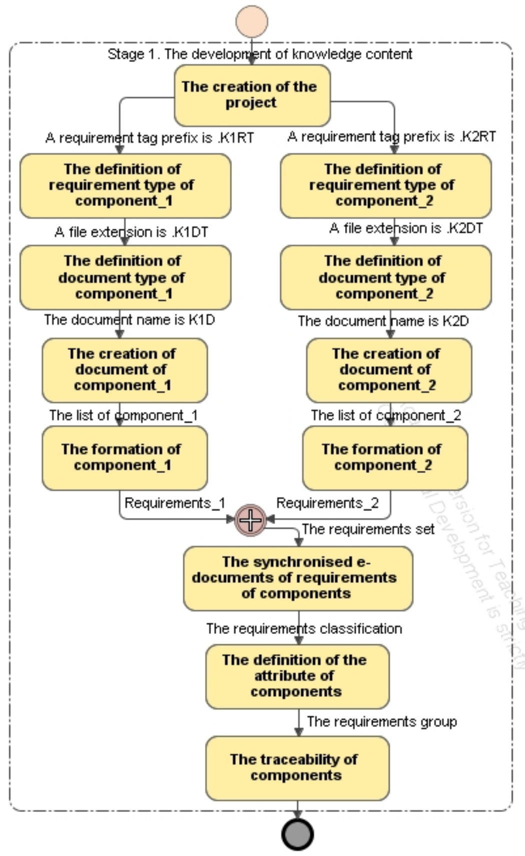


Fig. 4. Part of the method (Stage 1 "The development of knowledge content", BPMN notation)

Designing a knowledge base on the basis of DVCM and EMC. Management knowledge model based on DVCM and EMC is created in two stages (Fig. 5):

1. On the basis of the *detailed Value Chain Model* (DVCM) identifying *management Functions* and their information resources.
2. Each intersection between *management Function* and *Process* is modelled in detail as the managed process by creating their *Elementary Management Cycle* (EMC) models.

1st stage of designing a management knowledge model and a knowledge base. Designing a knowledge base on the basis of the *detailed Value Chain Model* (DVCM) requires to evaluate not only DVCM structure, but also all the informational attributes (data) of interactions between *management Functions* and *Processes*:

- analyze the general case of the *detailed Value Chain Model* (DVCM) which is the formalized DVCM structure;
- verify that *management Function* hierarchy exists;
- verify that *Process* hierarchy exists;
- verify that data (information attributes) related to each intersection between a *management Function* and a *Process* exists.

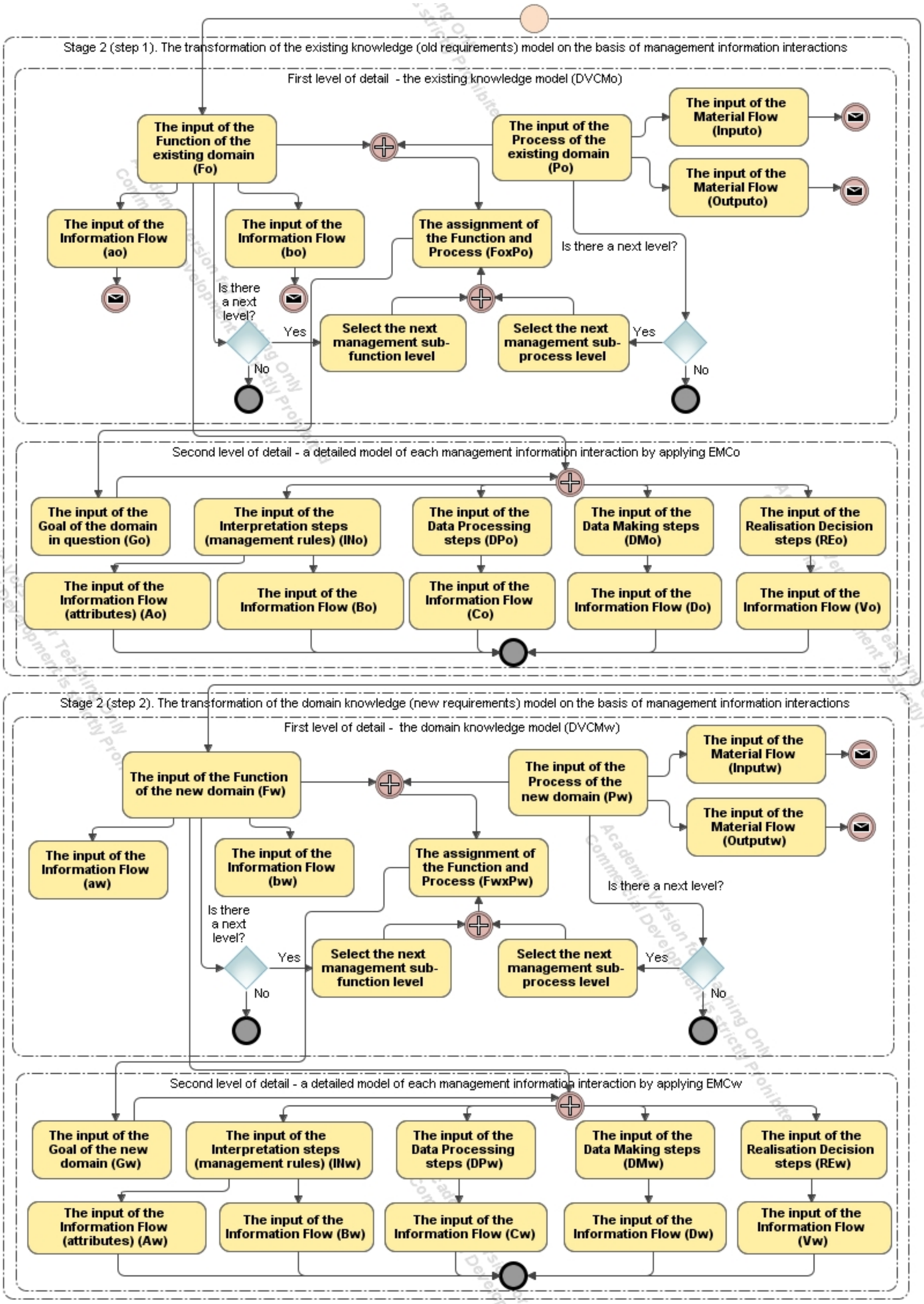


Fig. 5. Part of the method (Stage 2 "The transformation of the existing and the domain knowledge models on the basis of management information interactions", BPMN notation)

2nd stage of designing a management knowledge model and a knowledge base. The resulting management knowledge model does not assess the internal structure (informational transformations performed by the *enterprise management Function* itself) of *management Functions*, i.e. information interactions between the structural parts (components) of a *management Function*. Next, the internal structure of each *management Function* is modelled on the basis of *Elementary Management Cycle* (EMC) description:

- select a specific *management Function* and *Process* pair (intersection) identified with DVCM, that will further be modelled as a managed process (formally described EMC);
- create an *Elementary Management Cycle* (EMC) model of the selected *management Function* and *Process* intersection by identifying (naming) all EMC components: IN, DA, SP and RE processes, related goals G, information flows between EMC components;
- illustrate DVCM and EMC through domain entity classes model;
- develop a knowledge base prototype.

When requirements documents are created and linked together while the requirements themselves are kept in the Knowledge Base (KB), an opportunity appears to automate requirements analysis by using computerized systems (Fig. 6).

Within the selected CASE tool (*IBM Rational RequisitePro*™) special visual environments (so called views) for knowledge analysis are selected. It is possible to review knowledge (requirements) presented in different views in parallel by using various matrices or hierarchical structures (trees) where requirements with their attributes and/or traceability links between different requirement types are represented. For example, one of the problems of requirement analysis is the determination of requirements attributes.

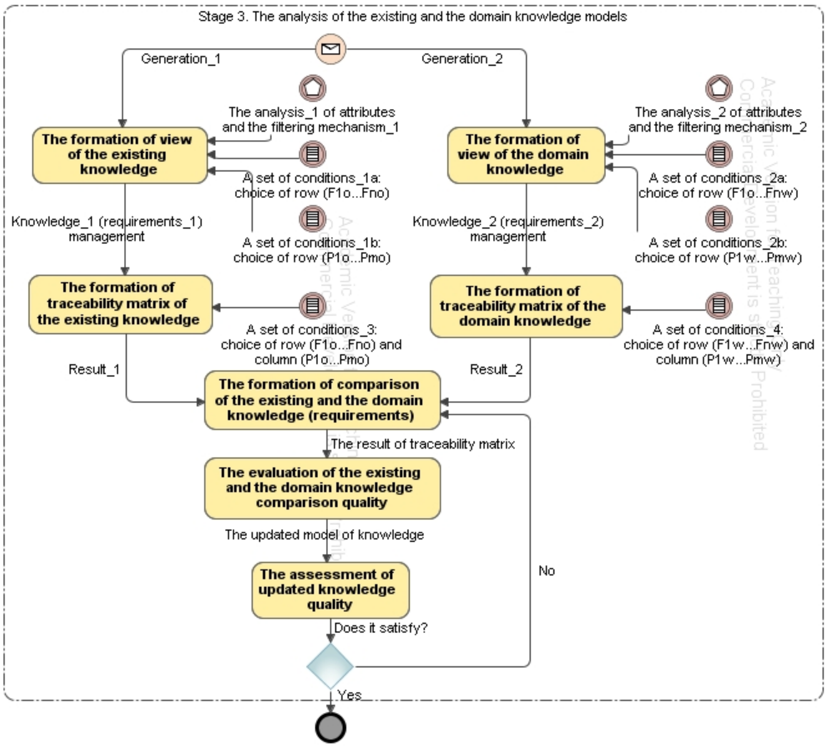


Fig. 6. Part of the method (Stage 3 " The analysis of the existing and the domain knowledge models", BPMN notation)

When applying the visual environment of *Attribute Matrix*, it is possible to comfortably revise all the requirements of a particular type and attributes related to them. When analyzing a curent domain (for example, study programme), the following actions with the matrix are performed: the establishment and editing of a requirement title, text, attributes and traceability links; the saving of matrix query; matrix printing. Filtering and sorting functions can also be applied to analyze requirements, thus maximizing the informational value of each requirement. By choosing one or several attributes and/or traceability signs (indicators), it is possible to perform requirement filtering and sorting, therefore, selecting the necessary requirements or forming new categories of them. In particular, it is used to design the architecture of a study programme and it's structural components (study blocks and modules).

The modification of a structured Value Chain Model to identify domain knowledge content. To reflect these needs the author has proposed a new problem domain life cycle model, which is based on *modified* VCM. This specialised model is presented in Figure 7.

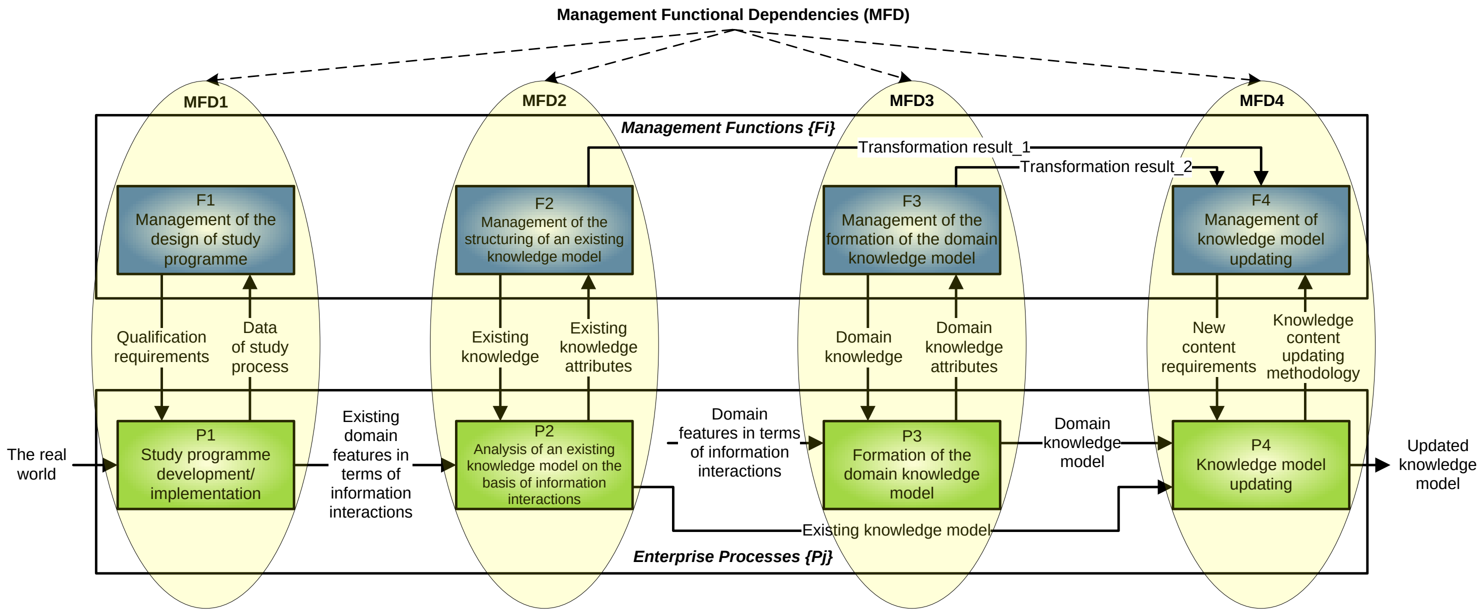


Fig. 7. Informational model of domain knowledge content identification process

This model allows to specify the requirements for the managed object (lowering down) and define its characteristics (requirement localization).

Each interaction between *management Function Fi* and *Process Pj* $(Fi \times Pj)$ identified with DVCM is seen as a *managed process* that implements a specific *Management Functional Dependence* (MFD).

For example, the formalised model of the interaction between the *Function F2 = "Management of the structuring of an existing knowledge model"* and the *Process P2 = "Analysis of an existing knowledge model on the basis of information interactions"* (see Fig. 7) is presented in Figure 8.

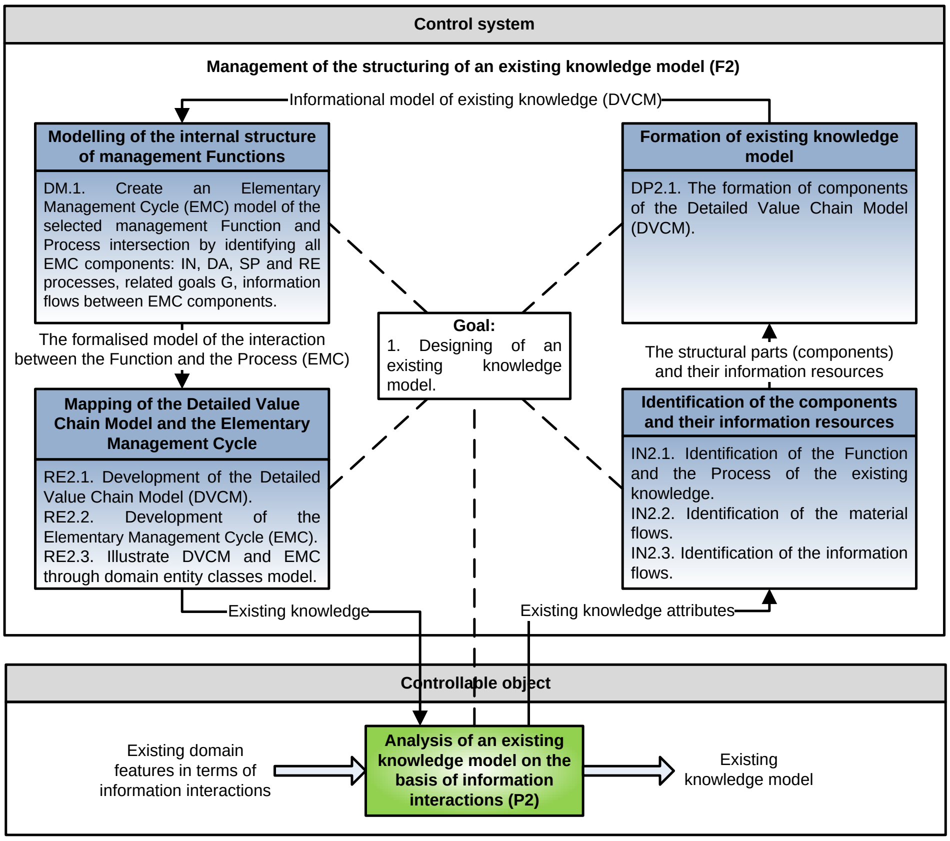


Fig. 8. The Formalised Model of the Interaction between the *Function F2 = "Management of the structuring of an existing knowledge model"* and the *Process P2 = "Analysis of an existing knowledge model on the basis of information interactions"*

It is important to mention that any specific EMC does not end, it continues with each new domain process cycle. Thus, the current domain is developed based upon the spiral principle with the renewal of the quality in general. In particular, the steps (rules) and the attributes (data) of the transformation of current domain management information were identified.

3. DOMAIN MANAGEMENT KNOWLEDGE CONTENT IDENTIFICATION AND KNOWLEDGE UPDATING COMPUTERIZED PROCESS MODEL

This chapter provides a principal scheme of knowledge content identification and domain knowledge updating model, on the basis of which a knowledge-based specification is performed. A particular enterprise *Management Functional Dependency* is the aggregate of the necessary information interactions among specific *enterprise management Function* components, specified by a particular *Elementary Management Cycle*.

Construction of a principal scheme of knowledge content identification and domain knowledge updating model. A principal knowledge content identification and domain knowledge updating model scheme is provided in Figure 9.

When analysing the principal knowledge model scheme, it can be stated that *management Function* model, in this case, becomes a scenario that defines mandatory information interactions, necessary to computerize all tasks that constitute *management Function*.

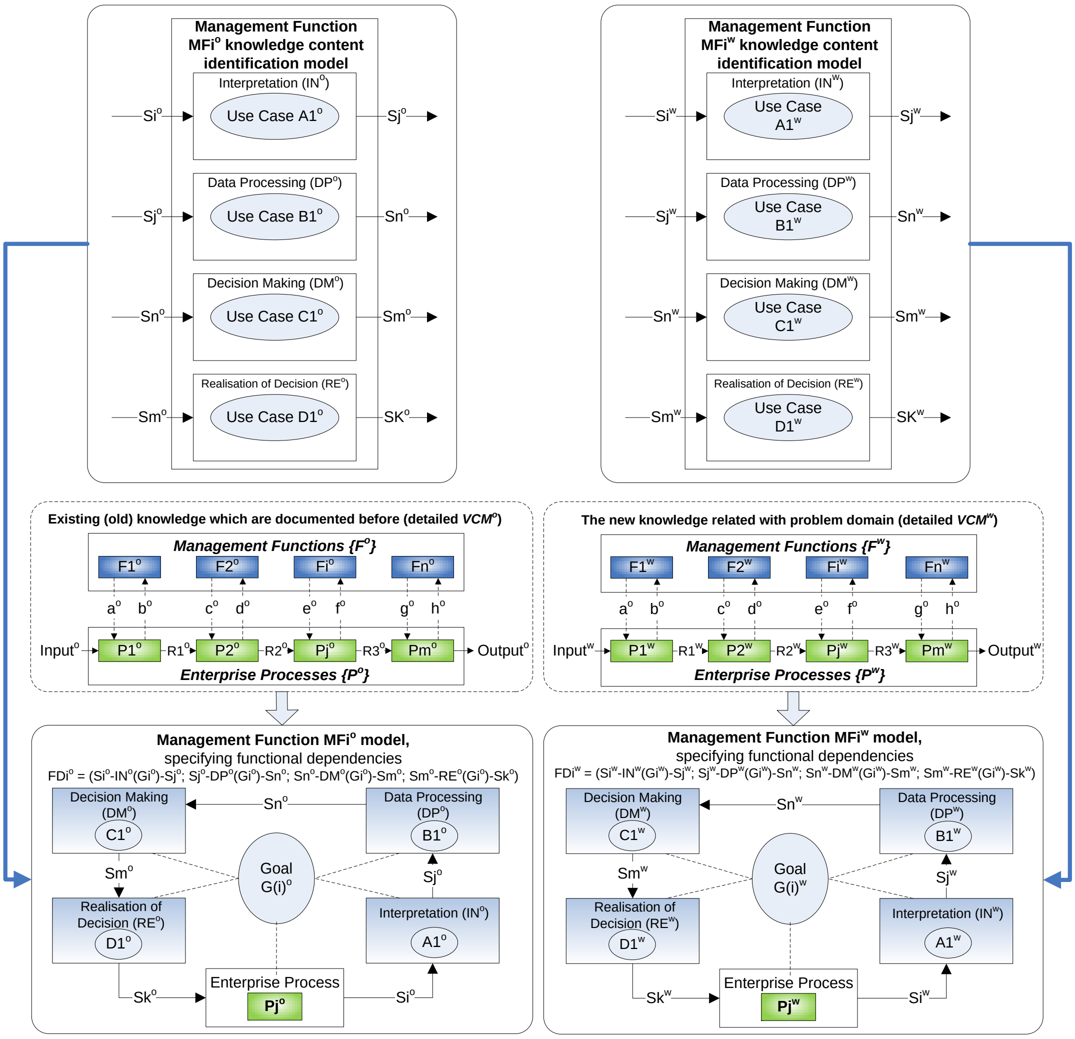


Fig. 9. A principal scheme of knowledge content identification and domain knowledge updating model (based on enterprise management function)

The identification of the existing (old) knowledge which are documented before and the new subject area knowledge related with problem domain is presented in two upper detailed VCM in Figure 10.

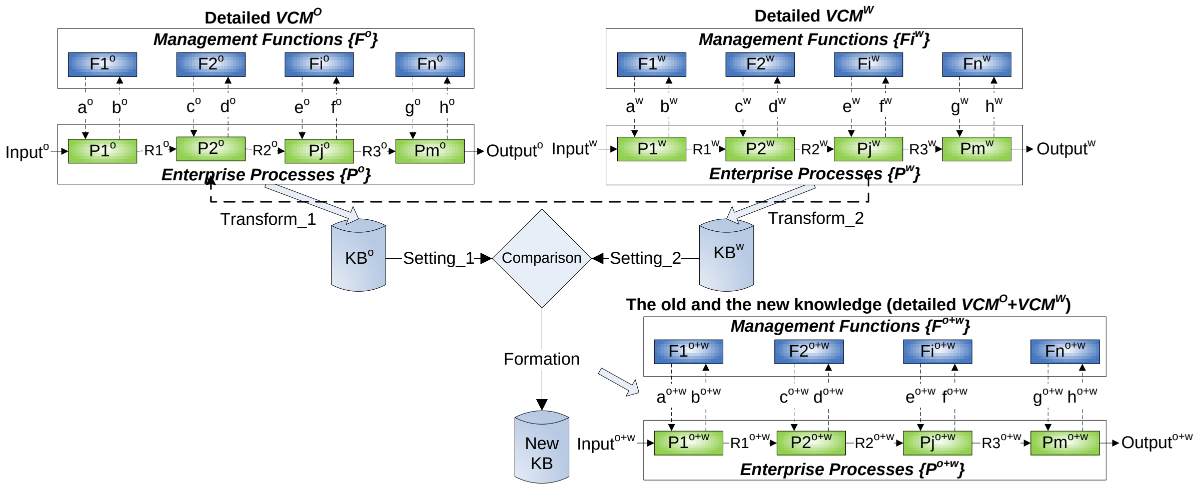


Fig. 10. The identification of the existing and the new knowledge (here: VCM^o – the old knowledge; VCM^w – the new knowledge)

Mapping and comparison of the old and the new knowledge allow to construct a *modified* VCM producing a set of updated knowledge.

A technique of knowledge analysis is used for comparison between detailed VCM^o and VCM^w. *Traceability Matrix* shows the relation of two knowledge sets (Fig. 11).

There are several situations that can be identified: 1) *the sameness* of knowledge; 2) *the supplementation* of existing knowledge with lacking new subject area knowledge; 3) *the addition* (*inclusion*) of new domain knowledge; 4) *the exclusion* of present knowledge from the current domain (no longer meets subject area requirements, competency becomes less important).

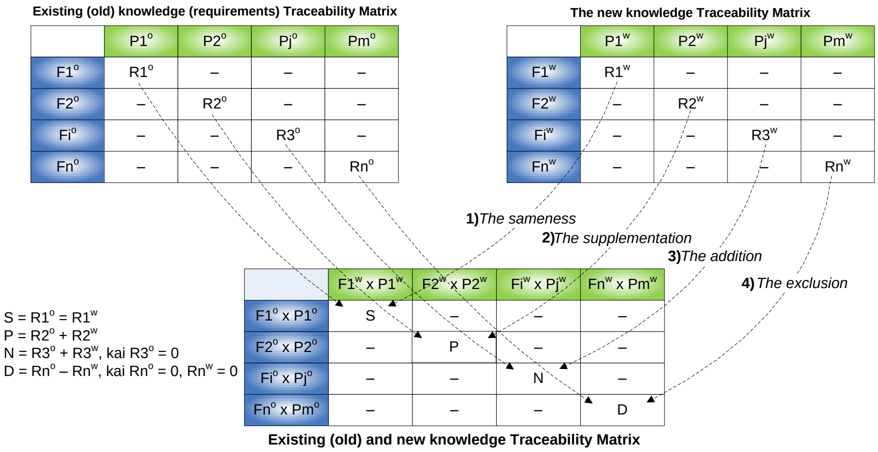


Fig. 11. The relations between existing (old) and the new knowledge (requirements)

On the basis of the fundamental content refinement process model scheme are provided current domain (study programme) and subject area knowledge (requirement) detailed VCM examples.

An example of a current study programme knowledge (old requirements) and subject area knowledge (new requirements) detailed VCM is described in the thesis.

Implementation of the knowledge repository of proposed model for current domain requirement improvement and knowledge content updating. Subject area entity class model (logical data model) is presented in Figure 12 (using *MagicDraw* CASE tool).

Requirement management is performed on the basis of *IBM Rational RequisitePro*^{TM} tool, meanwhile, the structure of a knowledge base is based on detailed Value Chain and Elementary Management Cycle.

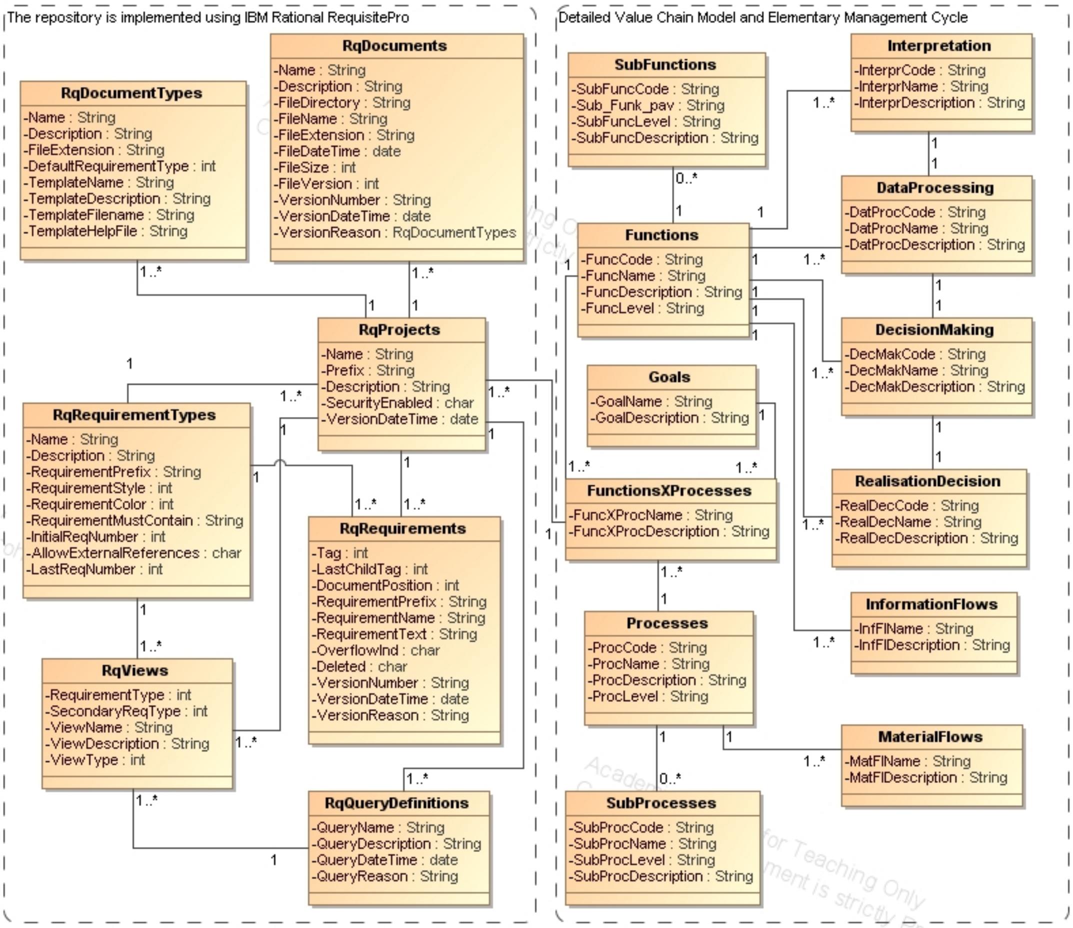


Fig. 12. Subject area entity class model (logical data model)

Subject area entity class model of the knowledge repository specifies *VCM* and *EMC* elements which can be a function (consisting of information activities: *Interpretation*, *Data Processing*, *Decision Making* and *Decision Realization*), a process, a material flow, etc. Thus, the current domain parameters derived from detailed *VCM* and *EMC* together with requirements are stored in the knowledge databases and can be used for solving specific problems of expected application areas.

4. APPLICATION METHODOLOGY FOR A STUDY PROGRAMME REQUIREMENT ANALYSIS AND KNOWLEDGE CONTENT UPDATING SYSTEM

A fundamental study programme requirement analysis system and knowledge content updating application methodology scheme (graphical illustration) is provided in Figure 13 that separates the stages of "Interpretation", "Data Processing", "Decision Making", "Realisation of Decision" procedures.

The following is a demonstration of the application methodology for a study programme requirement analysis and knowledge content updating system.

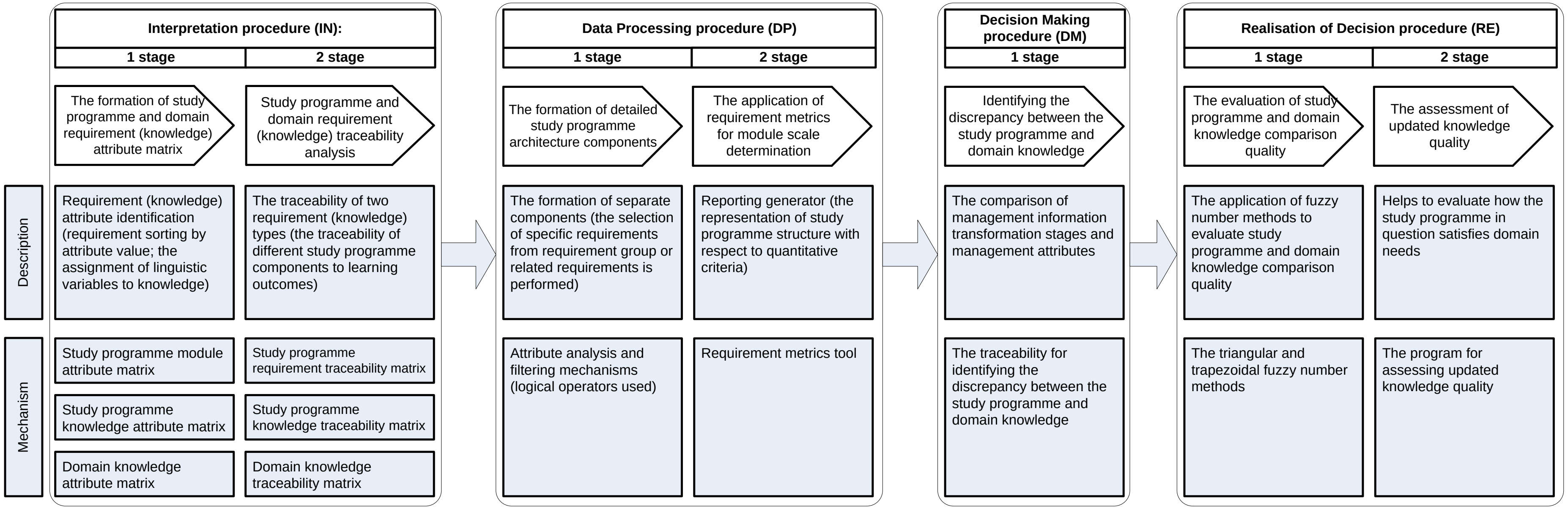


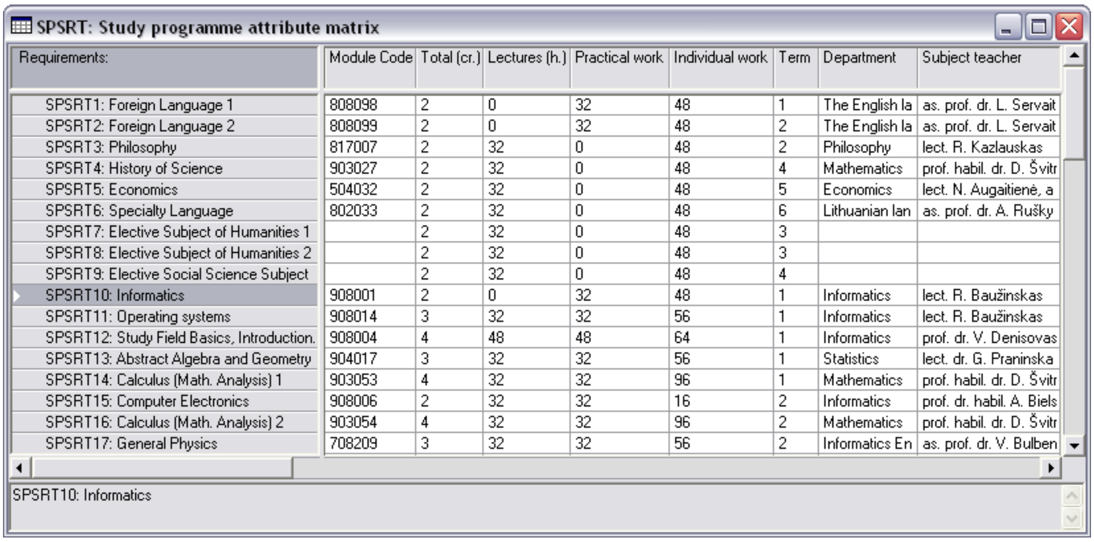
Fig. 13. A fundamental study programme requirement analysis and knowledge content updating system application methodology scheme (illustrated "Interpretation", "Data Processing", "Decision Making" and "Realisation of Decision" procedure stages)

Interpretation procedure (IN).

1 stage. The formation of study programme and domain requirement (knowledge) attribute matrix.

Study programme module attribute matrix. One of the problems of requirement analysis is the determination and traceability of requirements attributes. When applying the visual environment of Attribute Matrix, it is possible to comfortably revise all the requirements of a particular type and attributes related to them (Fig. 14).

In the matrix of study programme modules (components) attributes a list of subjects (modules) and attributes is provided: module code, scope in credits and hours, semester, component group, selection type, accounting form, etc. Standard attributes are assigned to the module list (reflecting study programme module structure adopted in the institution in question) while additional user-input attributes highlight the issues and specificity of a particular study programme. For example, in this matrix requirements can be temporarily sorted in different ways by the value of a selected attribute.



SPSRT: Study programme attribute matrix

Requirements:	Module Code	Total (cr.)	Lectures (h.)	Practical work	Individual work	Term	Department	Subject teacher
SPSRT1: Foreign Language 1	808098	2	0	32	48	1	The English la	as. prof. dr. L. Servait
SPSRT2: Foreign Language 2	808099	2	0	32	48	2	The English la	as. prof. dr. L. Servait
SPSRT3: Philosophy	817007	2	32	0	48	2	Philosophy	lect. R. Kazlauskas
SPSRT4: History of Science	903027	2	32	0	48	4	Mathematics	prof. habil. dr. D. Švitr
SPSRT5: Economics	504032	2	32	0	48	5	Economics	lect. N. Augaitienė, a
SPSRT6: Specialty Language	802033	2	32	0	48	6	Lithuanian lan	as. prof. dr. A. Rušky
SPSRT7: Elective Subject of Humanities 1		2	32	0	48	3		
SPSRT8: Elective Subject of Humanities 2		2	32	0	48	3		
SPSRT9: Elective Social Science Subject		2	32	0	48	4		
SPSRT10: Informatics	908001	2	0	32	48	1	Informatics	lect. R. Baužinskas
SPSRT11: Operating systems	908014	3	32	32	56	1	Informatics	lect. R. Baužinskas
SPSRT12: Study Field Basics, Introduction.	908004	4	48	48	64	1	Informatics	prof. dr. V. Denisovas
SPSRT13: Abstract Algebra and Geometry	904017	3	32	32	56	1	Statistics	lect. dr. G. Praninska
SPSRT14: Calculus (Math. Analysis) 1	903053	4	32	32	96	1	Mathematics	prof. habil. dr. D. Švitr
SPSRT15: Computer Electronics	908006	2	32	32	16	2	Informatics	prof. dr. habil. A. Biels
SPSRT16: Calculus (Math. Analysis) 2	903054	4	32	32	96	2	Mathematics	prof. habil. dr. D. Švitr
SPSRT17: General Physics	708209	3	32	32	56	2	Informatics En	as. prof. dr. V. Bulben

SPSRT10: Informatics

Fig. 14. Study programme module attribute matrix

Similarly, study programme and domain knowledge attribute matrices are constructed. A distinguishing feature of these matrices is an application of linguistic variables to knowledge attributes values (with possible value set: Excellent, Good, Satisfactory, Sufficient, Insufficient). For example, value assigned to "Assimilation of DFD, DSD, HF notations" is "Good".

2 stage. Study programme and domain requirements (knowledge) traceability analysis.

Study programme requirement traceability matrix. An alternative method of requirement analysis is Traceability Matrix that shows the relation of two requirement types. Traceability Matrix (Fig. 15) shows how different study programme components are linked to study outcomes (an arrow pointing from one requirement to another shows direct traceability).

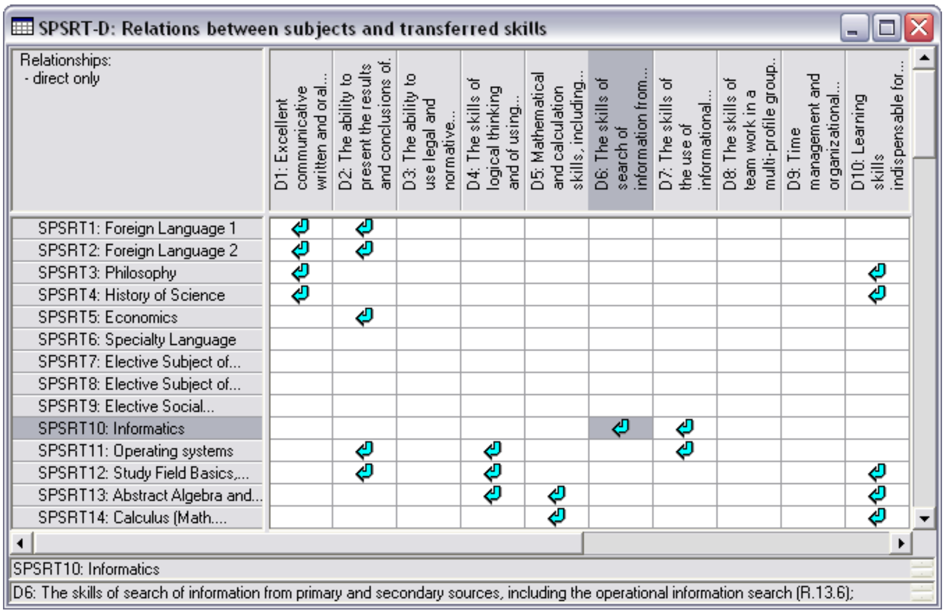


Fig. 15. Relations between subjects and learning outcomes (transferred skills)

Study programme requirement traceability matrix is used when applying decomposition principle called "top-down approach" by consistently dividing a selected component into subcomponents as many times as it makes sense in terms of a modeling goal.

Data Processing procedure (DP).

1 stage. The formation of detailed study programme structure components. According to informatics studies regulations, a study programme is composed of three target sections: general education section, study foundation field fundamentals section (forms the core of a study programme) and special (professional) section which provides deeper knowledge and skills oriented towards further professional or research enterprise. When creating and renewing informatics study programmes, the above sections should be elaborated on the levels of study areas (subgroups) and elements (subjects). The example (Fig. 16) illustrates how study field fundamentals section is formulated according to the existing regulations.

SPSRT: Study Field Fundamentals subjects

Requirements:	Module Code	Term	Department	Subject teacher	Group 1 - Flt:Y Srt:A	Subgroup
SPSRT10: Informatics	908001	1	Informatics	lect. R. Baužinskas	Study field fundamentals	Study Field General subject
SPSRT11: Operating systems	908014	1	Informatics	lect. R. Baužinskas	Study field fundamentals	Study Field General subject
SPSRT12: Study Field Basics, Introduction to...	908004	1	Informatics	prof. dr. V. Denisovas	Study field fundamentals	Study Field General subject
SPSRT13: Abstract Algebra and Geometry	904017	1	Statistics	lect. dr. G. Praninska	Study field fundamentals	Mathematics subjects
SPSRT14: Calculus (Math. Analysis) 1	903053	1	Mathematics	prof. habil. dr. D. Švitr	Study field fundamentals	Mathematics subjects
SPSRT15: Computer Electronics	908006	2	Informatics	prof. dr. habil. A. Biels	Study field fundamentals	Study Field General subject
SPSRT16: Calculus (Math. Analysis) 2	903054	2	Mathematics	prof. habil. dr. D. Švitr	Study field fundamentals	Mathematics subjects
SPSRT17: General Physics	708209	2	Informatics	as. prof. dr. V. Bulben	Study field fundamentals	Study Field General subject
SPSRT18: Programming and Web Pages Development	908005	2	Informatics	prof. dr. V. Denisovas	Study field fundamentals	Study Field General subject
SPSRT19: Mathematical Logic	908008	3	Informatics	as. prof. dr. D. Baziuk	Study field fundamentals	Theoretical informatics subj
SPSRT20: Differential Equations	903017	3	Mathematics	prof. habil. dr. D. Švitr	Study field fundamentals	Mathematics subjects
SPSRT21: Computer Graphics	908020	3	Informatics	prof. dr. habil. A. Biels	Study field fundamentals	Study Field General subject

SPSRT18: Programming and Web Pages Development

Fig. 16. Study field fundamentals subjects

Filtering and sorting functions can also be applied to analyze requirements, thus maximizing the informational value of each requirement. By choosing one or several attributes or traceability in the created views, it is possible to perform requirement filtering and sorting, therefore, selecting the necessary requirements or forming new categories.

The example (Fig. 17) illustrates how the list of subjects of the 3 semester of the study programme in question is formulated.

SPSRT: 3 semester subjects

Requirements:	Module Code	Lectures (h.)	Practical work	Individual work	Term 1 - Flt	Department	Subject teacher	Group
SPSRT7: Elective Subject of Humanities 1		32	0	48	3			General education
SPSRT8: Elective Subject of Humanities 2		32	0	48	3			General education
SPSRT19: Mathematical Logic	908008	32	32	56	3	Informatics	as. prof. dr. D. Baziuk	Study field fundamental
SPSRT20: Differential Equations	903017	32	32	56	3	Mathematics	prof. habil. dr. D. Švitr	Study field fundamental
SPSRT21: Computer Graphics	908020	32	32	56	3	Informatics	prof. dr. habil. A. Biels	Study field fundamental
SPSRT22: Theory of Algorithms	908027	32	32	56	3	Informatics	prof. dr. V. Denisovas	Study field fundamental
SPSRT23: Data Structures and Algorithms	908009	32	64	64	3	Informatics	prof. dr. V. Denisovas	Study field fundamental

SPSRT22: Theory of Algorithms

Fig. 17. The summary of the 3 semester subjects of the study programme

2 stage. The application of requirement metrics for module scale determination. Requirement Metrics are applied to form a final study programme structure by taking into account quantitative criteria (limitations) in requirements. By connecting various filters with *and*, *or* or *not* logical operators, it is possible to create structural components of all levels of study programme. System requirement metrics allow study programme compilers to prepare statistical reports and export those into

MS Excel[TM] spreadsheet (for a further statistical processing and results visualization). It is metrics that allow determining whether the created study programme satisfies the criteria defined in study regulations and other documents. An example of the query of requirement metrics report formulation is provided in Table 2.

Table 2. The attributes of the query of requirement metrics report formulation

Title	Criterion
2 credits	BASE = SPS: Study programme attribute matrix AND “Section” IN [Study Field Fundamentals] AND “Subject scale” IN [2]
3 credits	BASE = SPS: Study programme attributes matrix AND “Section” IN [Study Field Fundamentals] AND “Subject scale” IN [3]
...	...
5 credits	BASE = SPS: Study programme attributes matrix AND “Section” IN [Study Field Fundamentals] AND “Subject scale” IN [5]

The application of this metric also ascertains if the scope of a study programme is sufficiently balanced (in the study field fundamentals component group of the analyzed study programme an optimal-scope 3 and 4 credit subjects are the dominant ones).

Decision Making procedure (DM).

1 stage. Identifying the discrepancy between the existing and new knowledge. Traceability matrix is developed to identify the discrepancy between study programme knowledge (old requirements) and domain knowledge (new requirements) and is presented in Figure 18.

Existing (old) and new knowledge Traceability Matrix

	$Pj(G)^w$	A^w	$IN(G)^w$	B^w	$DA(G)^w$	C^w	$SP(G)^w$	D^w	$RE(G)^w$	V^w
$Pj(G)^o$	R^T	–	–	–	–	–	–	–	–	–
A^o	–	R^T	–	–	–	–	–	–	–	–
$IN(G)^o$	–	–	R^T	–	–	–	–	–	–	–
B^o	–	–	–	R^T	–	–	–	–	–	–
$DA(G)^o$	–	–	–	–	R^T	–	–	–	–	–
C^o	–	–	–	–	–	R^T	–	–	–	–
$SP(G)^o$	–	–	–	–	–	–	R^T	–	–	–
D^o	–	–	–	–	–	–	–	R^T	–	–
$RE(G)^o$	–	–	–	–	–	–	–	–	R^T	–
V^o	–	–	–	–	–	–	–	–	–	R^T

Fig. 18. Traceability matrix for identifying the discrepancy between the existing and new knowledge ($R^T \in \{S, P, N, D\}$)

Formal description of traceability matrix elements is as follows (where $R_1^o, R_2^o, ..., R_n^o$, $R_1^w, R_2^w, ..., R_n^w$ – the existing and new managed process results):

$$R^T = \begin{cases} S, & when \quad R_1^o = R_1^w, \quad R_1^o, R_1^w \\ P, & when \quad R_2^o \le R_2^w, \quad R_2^o + R_2^w \\ N, & when \quad R_3^o = 0, \quad R_3^o + R_3^w \\ D, & when \quad R_n^o = 0, R_n^w = 0, \quad R_n^o - R_n^w \end{cases}, \qquad (5)$$

It means that four cases are distinguished:

- equal requirements, knowledge remains *unchanged* (denoted by *S*);

- current knowledge only partially covers new requirements and should be *supplemented* (denoted by *P*);
- *the addition* of new domain knowledge (denoted by *N*);
- *the exclusion* of present knowledge from the current domain (no longer meets subject area requirements, competency becomes less important) (denoted by *D*).

Realisation of Decision procedure (RE).

1 stage. The evaluation of study programme and domain knowledge comparison quality. Current and domain knowledge are evaluated based on the created quality model by applying triangular and trapezoidal fuzzy number methods. The application of triangular and trapezoidal fuzzy number methods can be found in the experimental study of domain knowledge modelling method.

2 stage. The assessment of updated knowledge quality. Updated knowledge quality assessment software program is designed to help study programme developers and employees in higher schools, who prepare or refine study programmes, to evaluate how their study programmes satisfy domain (labour market) needs after knowledge content update (Fig. 19).

	A	B	C	D	E	F	G	H	I	J
1	The assessment Worksheet of updated knowledge quality									
2		Enter course name(s) in the columns to the right; cut/paste for additional columns or delete as needed.								
3		Enter 1 through 5 indicating the highest level in which your course addresses each knowledge.								
4		Refer to the "Definitions" tab in this worksheet for explanation of mastery levels								
5		○ 1	No Awareness	Course name 1	Course name 2	Course name 3	Course name 4	Course name 5	Average	Alignment w/knowledge
6		◔ 2	Awareness/Knowledge							
7		◑ 3	Comprehension/Application							
8		◕ 4	Analysis							
9		● 5	Synthesis/Evaluation							
10			Average score per course	1,00	1,00	1,00	1,00	1,00	1,00	
11			% of knowledge addressed	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
12	Atnaujintos žinios									
13		**1.** Studying of enterprise analysis and analyst's activity (P1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
14		**2.** Knowledge of enterprise modelling (DFD, DSD, FH) notations and analyst qualification requirements (A1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
15		**3.** Assimilation of DFD, DSD, FH notations and UML language notation (IN1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
16		**4.** Assimilated knowledge of DFD, DSD, FH notations and UML language notation (B1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
17		**5.** Activity domain modelling using structural-functional and object methods		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
18		**6.** Activity domain model (C1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
19		**7.** Verification of activity domain model (DM1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
20		**8.** Activity domain errors (D1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
21		**9.** Adjustment of activity domain model (RE1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	
22		**10.** Correctly designed enterprise model and process modelling methods (V1o+w).		○ 1	○ 1	○ 1	○ 1	○ 1	○ 1,00	

Fig. 19. Image of the form for updated knowledge quality evaluation (the form is blank)

5. METHOD APPROBATION

This chapter presents the experimental research of the domain knowledge modelling method for identifying the comparison between the current study programme, based on requirement management, detailed Value Chain and Elementary Management Cycle, and subject area knowledge, in order to examine method usability and development opportunities.

The experimental research consists of three stages: first stage involves demonstrating the development of knowledge content; second stage involves the development of study program and subject area knowledge models; third stage involves the analysis of the study program and subject area knowledge. The evaluation of a study programme and subject area knowledge is based on the application of triangular and trapezoidal fuzzy number methods.

GENERAL CONCLUSIONS

After conducting theoretical and experimental research, the following scientific and practical conclusions were drawn:

1. After analysing domain modelling methodologies and methods (languages, notations), it was found that a knowledge based enterprise modelling method for identifying organization's operational objectives, as well as the informational interaction between a controlled object and a controlling system forming control feedback, can be based on a detailed Value Chain Model.
2. A domain management knowledge modelling method was designed, allowing to create a two-level (granular) model for describing knowledge of domain management information interactions:
   2.1. the highest level modified Value Chain Model is further elaborated into a set of Elementary Management Cycle models;
   2.2. the chosen hierarchical structure allows to create new structural knowledge models and to update existing ones.
3. The created domain knowledge content identification and updating enterprise model has been formally described, IS engineering process measures for its realization have been proposed, a computerized CASE system has been implemented.
4. The work goes beyond formal presentation; the method is applied in a specific domain, which requires constant knowledge content updating – higher education study programme construction and updating.
5. The flexibility of the proposed solutions should be noted – the possibility to integrate the created domain knowledge base into a study programme requirement management system, by identifying knowledge management processes and implementing a knowledge subsystem in an existing CASE system environment.
6. Methodology for the application of this system has been proposed, while its approval provided results relevant and useful for the members, employees and professionals of academic community in higher education institutions, working on improving existing study programs and seeking to ensure their quality.
7. Actual application results (derived from updating KU informatics study programmes) allow to draw conclusions about the wide application possibilities of this methodology in developing and updating the study programmes in higher education institutions.
8. The proposed method has been approved by an experiment, which confirmed the correctness of the proposed solutions. The quality of updated domain knowledge has also been evaluated, taking into account requirements for professionals' knowledge, estimating the compliance with the requirements and summarizing subjective opinions of the experts based on the fuzzy number theory.
9. The resulting knowledge-based computerized CASE system may be further elaborated and applied not only in identifying and updating the knowledge content of the domain knowledge in question, but also in other domains

(creating and updating corporative knowledge bases, improving BI measures, implementing knowledge bases, evaluating the completeness of the content knowledge of accumulated in repositories). It may be objectives of further works.

LIST OF THE AUTHOR'S PUBLICATIONS ON THE SUBJECT OF THE DISSERTATION

Articles in journals from the list of Institute of Scientific Information (ISI) Web of Science database

1. **TEKUTOV, Jurij**; GUDAS, Saulius; and DENISOVAS, Vitalijus. The Refinement of Study Program Content Based on a Problem Domain Model. *Transformations in Business & Economics.* Vol. 11, No. 1 (25). Vilnius: Vilnius University Press, 2012, p. 199–212. ISSN 1648-4460. [ISI Web of Science (& Master Journal List) by Thomson Services], [Social Sciences Citation Index], [Social SciSearch], [Journal Citation Reports], [EBSCO], [IBSS], [EconLit], [e-JEL], [JEL on CD], [SCOPUS], [Cabell's Directory of Publishing Opportunities] (IF = 1.67).

Articles in other ISI indexed journals (ISI Proceedings)

2. **TEKUTOV, Jurij**; GUDAS, Saulius; DENISOVAS, Vitalijus; ir TEKUTOVA, Julija. The Study Programme Requirements Enhancement Based on a Problem Domain Model. *INTEL-EDU 2012: 3rd International Workshop on Intelligent Educational Systems and Technology-enhanced Learning (associated with Riga Technical University 53rd International Conference Dedicated to the 150th Anniversary and the 1st Congress of World Engineers and Alumni of RPI/RTU).* Ryga: JUMI Publishing House LTD, 2012, p. 108–120. ISBN 978-9984-30-210-2.

Articles in peer reviewed international and foreign journals and periodical publications

3. **TEKUTOV, Jurij**; GUDAS, Saulius; and DENISOVAS, Vitalijus. Study process management based on the modified value chain model. *Information Science.* Vol. 56. Vilnius: Vilnius University, 2011, p. 50–62. ISSN 1392-0561. [Central and Eastern European online Library – CEEOL], [Central & Eastern European Academic Source (EBSCO sąrašas)], [Library and Information Science Abstracts – LISA]. (In Lithuanian).
4. **TEKUTOV, Jurij**; GUDAS, Saulius; DENISOVAS, Vitalijus; and TEKUTOVA, Julija. Study programme requirements engineering system‘s: from automated document‘s management to model-based development process. *Journal of Management* No. 1 (Vol. 18). Klaipeda: Klaipeda University Press, 2011, p. 33–43. ISSN 1648-7974. [IndexCopernicus – IC], [Central and Eastern European online Library – CEEOL], [EBSCO Publishing, Inc. Business Source Complete]. (In Lithuanian).
5. DENISOVAS, Vitalijus; GUDAS, Saulius; and **TEKUTOV, Jurij**. The engineering method and information system of study programme requirements. *Information Science.* Vol. 53. Vilnius: Vilnius University, 2010, p. 106–126. ISSN 1392-0561. [Central and Eastern European online Library – CEEOL], [Central & Eastern European Academic Source (EBSCO)], [Library and Information Science Abstracts – LISA]. (In Lithuanian).
6. DENISOVAS, Vitalijus; GUDAS, Saulius; **TEKUTOV, Jurij**; TEKUTOVA, Julija; and GALDIKIENĖ, Sigita. Improvement of informatics non-university study programme using curriculum requirements management system. *Professional studies:*

Theory and Practice: scientific journal No. 6. Šiauliai: Šiauliai College, 2010, p. 239–251. ISSN 1822-3648. [IndexCopernicus – IC]. (In Lithuanian).
7. DENISOVAS, Vitalijus; GUDAS, Saulius; **TEKUTOV, Jurij**; TEKUTOVA, Julija; and BRAUKLYTĖ, Ilona. Application of requirements engineering methods for improvement of the informatics bachelor‘s study programme. *Journal of Management* No. 1 (Vol. 14). Klaipeda: Klaipeda University Press, 2009, p. 123–131. ISSN 1648-7974. [IndexCopernicus – IC], [Central and Eastern European online Library – CEEOL], [EBSCO Publishing, Inc. Business Source Complete]. (In Lithuanian).

Articles in the reviewed scientific periodical publications

8. DENISOVAS, Vitalijus; GUDAS, Saulius; **TEKUTOV, Jurij**; and TEKUTOVA, Julija. Match fixing of learning outcomes of informatics study programme and professional requirements. *Technology science works in Western Lithuania region VII.* Klaipeda: Klaipeda University Press, 2010, p. 154–159. ISSN 1822-4652. (In Lithuanian).

Articles in the other International Scientific Conference Proceedings

9. DENISOVAS, Vitalijus; GUDAS, Saulius; **TEKUTOV, Jurij**; and TEKUTOVA, Julija. Система управления требованиями учебных программ в области информатики. *The collection of works of international scientific-practical conference "Modern problems and ways of their solution in science, transport, production and education' 2010" on Technical sciences*. Vol. 2. Odesa: издательство Черноморье, 2010, p. 37–46. ISBN 966-555-157-4. (In Russian).

Theses published in international scientific conference publications

10. DENISOVAS, Vitalijus; GUDAS, Saulius; **TEKUTOV, Jurij**; and TEKUTOVA, Julija. Requirements management system in the field of informatics non-university study programmes. *Higher Education Studies: Challenges and Opportunities.* Šiauliai: Šiauliai College, 2010, p. 158–160. ISSN 978-609-415-014-2. (In Lithuanian).

Articles in local conference proceedings

11. GUDAS, Saulius; DENISOVAS, Vitalijus; **TEKUTOV, Jurij**; and BRAUKLYTĖ, Ilona. Implementation of the requirements engineering methodology in modernisation of graduate study programmes of informatics. *Speeches of the 14th interuniversity scientific conference of graduates and PhD students "Information Technologies 2009".* Vilnius University, Vytautas Magnus University, Kaunas University of Technology, 2009, p. 229–234. ISSN 2029-249X. (In Lithuanian).
12. **TEKUTOV, Jurij**; TEKUTOVA, Julija; and BRAUKLYTĖ, Ilona. Analysis of the knowledge-based activity modelling methods and measures. *Fundamental Research and Innovations in the Juncture of Sciences.* Klaipeda: Klaipeda University Press, 2009, p. 117–126. ISBN 978-9955-18-329-7. (In Lithuanian).

Acknowledgements. I sincerely thank study supervisor Prof Dr (HP) Saulius Gudas for an opportunity to improve, and for a consistent guidance and support in preparing this dissertation. Special thanks to consultant Prof Dr Vitalijus Denisovas for valuable scientific consultations, constant support, expressed interest in the success of the work and comprehensive assistance.

Short description about the author. Jurij Tekutov was born in 1983 in Klaipeda (Lithuania). In 2006 he has finished Computer science (informatics) studies and a received Bachelor in Informatics at Klaipeda University. In 2008 he finished the informatics study program at Klaipeda University and obtained a Master's degree in Informatics. 2008 – 2012 – PhD studies at the Vilnius University of Kaunas Faculty of Humanities, Informatics Department. Current position: a lecturer at the Department of Informatics Engineering at Klaipeda University and at the Department of Information Technologies at Klaipeda State College. In 2008 he took part in project "Modernisation, development and assurance of mobility of Master study programmes in information technology area" implementation. In 2009 he attended 10th extended workshop / summer school of NordForsk network "Methodologies for Interactive Networked Enterprises – MINE" in Nida. E-mail: jurij@ik.ku.lt.

REZIUMĖ

Tyrimų sritis ir problemos aktualumas. Pastaruoju metu informacijos sistemų (IS) kūrimo proceso gyvavimo ciklai nagrinėjami skirtingais požiūriais. Vienas perspektyviausių yra modeliais grindžiamas požiūris (angl. *model-driven approach*), kuriuo siekiama panaudoti formalias žinias apie dalykinę sritį inžinerijos procesui vykdyti, koreguoti ir valdyti. Vienas pirmųjų IS kūrimo gyvavimo ciklo etapų yra veiklos modeliavimas (angl. *enterprise modelling*), kuris tampa ypač svarbus taikant modeliais grindžiamą IS kūrimo metodologiją – MDE (angl. *Model Driven Engineering*) ir jos metodus. Veiklos modeliavimo tikslas – sukurti organizacijos veiklos modelį (angl. *enterprise model*), kuris padėtų spręsti analizuojamos dalykinės srities problemas. Modeliais grindžiamos architektūros MDA (angl. *Model Driven Architecture*) koncepcija apibrėžia IS kūrimo procesą kaip trijų lygmenų modelių (CIM (angl. *the Computation Independent Model*), PIM (angl. *the Platform Independent Model*) ir PSM (angl. *the Platform Specific Model*)) sąveiką, kurios paskirtis – tiesiogiai susieti veiklos modeliavimo bei tolimesnius IS analizės, projektavimo ir įgyvendinimo etapus. Tad veiklos modeliavimo specialistų požiūriu, MDA koncepciją reikėtų vertinti kaip pastangas veiklos modeliavimo metodus susieti su viso gyvavimo ciklo (angl. *full life-cycle*) CASE (angl. *Computer-Aided Systems Engineering*) technologijomis. Šioje koncepcijoje atsispindi principas „iš apačios į viršų“ (angl. *bottom-up approach*), t. y. nuo programų inžinerijos metodų prie veiklos modeliavimo metodų IS inžinerijos poreikiams tenkinti. Tokiu būdu moderniose IS kūrimo metodikose vis didesnis dėmesys teikiamas ne tik (ir ne tiek) programų ir IS inžinerijos metodams, bet ir organizacijų veiklos modeliavimo problematikai, t. y. dalykinės srities žinių atvaizdavimui.

Vidiniu veiklos modeliavimu grindžiami IS inžinerijos metodai (IS gyvavimo ciklo etapų modeliai), kurių pagrindas yra veiklos srities (domeno) dėsningumo modelis, vadinami *žiniomis grindžiamais* (angl. *Knowledge-based*) metodais. MDE grindžiamoje kompiuterizuotoje IS inžinerijoje veiklos modelio paskirtis – susisteminti ir formalizuoti žinias apie tam tikrą analizuojamą domeną (dalykinę sritį). Šiuo metu formuojasi IS inžinerijos metodai, kurie leistų panaudoti CASE sistemose kaupiamas žinias apie veiklos funkcijas, procesus ir taisykles intelektualizuojant IS kūrimo procesą. Taip pat bandoma taikyti dalykinės srities žinių turinio sudarymo bei atnaujinimo metodus iš kitų gretimų IS inžinerijai sričių, tokių kaip el. mokymas, intelektinės verslo sistemos (angl. *Business Intelligence – BI*), veiklos taisyklių modeliavimas.

Žinių turinio specifikacijos sudarymas apima žinių apie dalykinės srities rinkimą, specifikacijos rengimą ir analizę, verifikavimą ir validavimą. Šiame darbe dalykinės srities žinios (angl. *domain knowledge*) suprantamos kaip dinamiškas žinių savybių rinkinys, aprašytas pagal nustatytus reikalavimus kaip faktų, principų ir teorijų, kurios yra susijusios su tam tikra profesine veikla, sistema. Kitaip tariant, žinių savybių rinkinys – tai dalykinės srities išraiška esamame kontekste. Veiklos modeliavimą galima priskirti reikalavimų rinkimo ir analizės etapui. Toliau veiklos modeliai transformuojami į kuriamos sistemos reikalavimų specifikaciją. IS inžinerijos proceso metu kuriamos sistemos (pavyzdžiui, domeno žinių turinio) ir atskirų jos komponentų specifikacijos. Kiekviename efektyviame dalykinės srities žinių (reikalavimų) valdymo procese svarbiausias žingsnis yra reikalavimų, kurie identifikuoti reikalavimų išrinkimo metu, formalus užrašymas. Tai padeda užtikrinti tikslų reikalavimų perdavimą ir valdymą juos plėtojant. Dokumentai yra tradicinė žmogui suprantama forma reikalavimams užrašyti.

Jie suteikia pagrindą kontekstui nustatyti ir reikalavimams fiksuoti, bet nėra labai tinkami papildant ir ypač modifikuojant reikalavimų informaciją. Dabartiniams, nuolat besikeičiantiems, veiklos poreikiams tenkinti reikalingas veiklos integravimas su informacijos technologijomis bei tokios kompiuterizuotos IS, kurios adaptuotųsi prie organizacijų veiklos pokyčių. Tokios sistemos turėtų būti kuriamos orientuojantis į vertės grandinės modelį (VGM), nes kompiuterizavimas apima veiklos valdymo funkcijas ir veiklos procesus bei organizacijos funkcinių sričių sąveikas, kurias ir atskleidžia VGM. Kompiuterizuojant veiklą atliekamas veiklos valdymo funkcijų ir veiklos procesų informacinių sąveikų tyrimas, kuris reikalauja tolimesnio vertės grandinės elementų dekomponavimo, t. y. vyksta VGM modifikavimas, sudaromas detalizuotas vertės grandinės modelis (DVGM). Tokiu būdu žiniomis grindžiamas IS kūrimas vykdomas remiantis tomis žiniomis apie veiklos valdymą, kurios apibrėžia esmines veiklos valdymo informacinių sąveikų savybes. Veiklos valdymo informacinė sąveika specifikuojama kaip elementarus valdymo ciklas (EVC), grįžtamojo ryšio kontūru susiejantis veiklos valdymo funkciją ir veiklos procesą.

Šiame darbe pasiūlytas dalykinės srities valdymo žinių modeliavimo metodas, kurio formalūs DVGM ir EVC aprašai pateikti BPMN notacija. Tuo pat metu neapsiribuojama formaliu pateikimu, todėl metodas detalizuojamas ir taikomas konkrečiame domene – aukštojo mokslo studijų programų konstravimo ir atnaujinimo srityje. Ši dalykinė sritis pasirinkta neatsitiktinai, bet remiantis metodologinėmis nuostatomis. Akademinėje srityje IS kūrimas tradiciškai vyksta pradedant veiklos valdymo funkcijų analize, kurioje svarbus veiklos valdymo informacinių sąveikų aspektas. Analizuojama aukštojo mokslo studijų dalykinė sritis yra daugiamatė, dinamiška ir reikalauja nuolatinio žinių turinio atnaujinimo. Tik laiku aktualizuojant studijų programos struktūrą ir žinių turinį, galima tinkamai atsižvelgti į nuolat besikeičiančius poreikius ir visų suinteresuotų pusių (angl. *stakeholders*) reikalavimus.

Vieni pirmųjų studijų programos kūrimą kaip IS inžinerijos procesą pradėjo nagrinėti tarptautinio MOCURIS (angl. *Modern Curriculum Development in Information Systems at Master Level*) projekto vykdytojai, kurių darbuose buvo pasiūlytas modelinės studijų programos konstravimo metodas. Jie įvedė naują sąvoką „studijų programų inžinerija“, apibrėžė studijų programos architektūrą ir, nagrinėdami studijų programą kaip sistemą, susidedančią iš kursų ir modulių, pritaikė standartinį IS inžinerijos procesą modelinei magistrantūros studijų programai kurti. Šiuo metodu buvo sukurtas didelės apimties IS srities magistrantūros programos specifikacijos dokumentas, kuriuo turėjo vadovautis konkrečių programų, kursų ir modulių kūrėjai. Tačiau jau projekto metu paaiškėjo esminių šio metodo trūkumų. Efektyviai taikyti tokį programos kūrimo procesą gebėjo tik jo autoriai, o sukurtas artefaktas (tekstinis specifikacijų dokumentas) dėl savo formalaus pobūdžio, sudėtingumo ir apimties buvo netinkamas vykdyti pagrindinę reikalavimų specifikacijos funkciją – užtikrinti visų proceso dalyvių komunikaciją ir koordinuoti jų darbą. Analizuojant atsiradusias reikalavimų atskyrimo, lokalizacijos bei detalumo (granuliavimo) problemas paaiškėjo, kad tai būdinga tradiciniam, neautomatizuotam, reikalavimų inžinerijos procesui. Todėl kitų tyrėjų (tarp jų ir šio darbo autoriaus) pasiūlyta šį procesą automatizuoti kuriant studijų programos reikalavimų inžinerijos sistemą CASE priemonių pagrindu. Tiesiogiai dalyvaujant autoriui tokia sistema buvo sukurta ir pritaikyta naujoms studijų programoms kurti. Tačiau greit besikeičiančioje aplinkoje ypač svarbu tampa laiku modernizuoti studijų programas, užtikrinti jų kokybę, atnaujinti žinių turinį. Visa tai reikalauja efektyvaus grįžtamojo ryšio su visomis suinteresuotomis pusėmis. Šioje disertacijoje minėtoms problemoms spręsti panaudotas detalizuotas vertės grandinės modelis (DVGM), užtikrinantis nuolatinį informacinį grįžtamąjį ryšį, kuris perduoda veiklą apibūdinančius duomenis (atributus) veiklos valdymo funkcijai ir grąžina valdymo sprendimus veiklos procesui nukreipti. Metodas realizuojamas sukuriant IS inžinerijos CASE sistemoje veiklos žinių saugyklą, kuri formuojama kompiuterizuojamos dalykinės srities veiklos modelio pagrindu.

Darbo mokslinė reikšmė. Tradicinėje kompiuterizuotoje IS inžinerijoje sistema kuriama empiriškai, pradedant vartotojo poreikių rinkimu, analize ir specifikavimu. Žiniomis grindžiamoje kompiuterizuotoje IS inžinerijoje sistema kuriama naudojant veiklos žinių bazę ir remiantis joje saugomomis dalykinės srities žiniomis. Manoma, kad pagrindinis žiniomis grindžiamos IS inžinerijos skiriamasis bruožas yra dalykinės srities žinių posistemis, kuris sudaromas pagal organizacijos veiklos modelį ir skirtas kaupti žinias apie kompiuterizuojamą dalykinę sritį. Toks žinių kaupimo posistemis turi atlikti dalykinės srities žinių, būtinų IS projektiniams modeliams kurti, saugyklos funkcijas. Pasiūlyta tokia šio posistemio struktūra: veiklos modelis ir veiklos metamodelis (formalizuota žinių apie veiklos valdymą struktūra). Dalykinės srities žinių kaupimo posistemis turi tapti esmine žiniomis grindžiamos CASE priemonės komponente, intelektualizuojančia visą IS kūrimo procesą. Tad tokio posistemio sudarymas yra viena iš aktualių mokslinių problemų, sprendžiamų šiame darbe.

Kita aktuali problema – valdymo informacinių sąveikų vertinimas modeliuojant organizacijos veiklą. Pasiūlyta nemažai veiklos modeliavimo metodų ir kalbų (DFD, IDEFx, UML plėtiniai, BPMN, DVGM ir kiti), tačiau tik nedaugelis jų įvertina dalykinės srities valdymo informacines sąveikas, kurios sistemų teorijos atžvilgiu būtinos veiklai valdyti. Dalykinės srities modelio, kuris apimtų duomenų, žinių, tikslų komponentus, būtinus valdyti mūsų nagrinėjamos dalykinės srities (t. y. organizacinės sistemos) veiklą, sudarymo ir tokių žinių atnaujinimo metodai yra aktuali mokslinė problema. Kompiuterizuojamo domeno (veiklos) informacinių sąveikų turinio identifikavimas bei atnaujinimas IS kūrimo, BI priemonių, žinių valdymo srityse šiandien yra dar neformalizuotas. Šiuo metu, taikant kompiuterizuotas priemones IS inžinerijos srityje, tik formuojasi formalizuoti metodai domeno valdymo informacinių sąveikų turiniui sudaryti bei atnaujinti (apimant duomenis, žinias, tikslus).

Darbe pateiktas nagrinėjamos dalykinės srities, toliau vadinamos „veiklos domenu“, žinių modelio atnaujinimo (žinių turinio identifikavimo ir atnaujinimo) metodas, grindžiamas veiklos modeliavimo valdymo informacinių sąveikų aspektu, kai yra identifikuojamos valdymo informacinės sąveikos tarp veiklos domeno komponentų. Sukurtas žinių turinio atnaujinimo modelis aprašytas formaliai, pasiūlytos kompiuterizuoto proceso įgyvendinimo priemonės. Metodas pritaikytas kuriant veiklos domeno žinių turinio identifikavimo bei atnaujinimo CASE sistemą, kuri aprobuota sudarant ir atnaujinant aukštojo mokslo informatikos pagrindinių (bakalauro) bei magistrantūros studijų programas.

Tyrimo objektas. Dalykinės srities žinių turinio, kuris apima veiklos domeno valdymo informacines sąveikas, identifikavimas, modeliavimas ir atnaujinimas.

Tyrimo tikslas – sukurti dalykinės srities žinių modeliavimo metodą, grindžiamą veiklos valdymo informacinėmis sąveikomis, ir kompiuterizuoti dalykinių žinių modelio atnaujinimo procesą.

Tikslui įgyvendinti darbe sprendžiami tokie **uždaviniai**:

1. Atlikti esamų dalykinės srities modeliavimo metodų analizę žinių turinio modeliavimo požiūriu.
2. Sukurti dalykinės srities valdymo žinių modeliavimo metodą, grindžiamą veiklos valdymo informacinėmis sąveikomis:
   2.1. apibrėžti veiklos valdymo funkcinių priklausomybių, sudarančių informacinį grįžtamąjį ryšį, seką;
   2.2. pritaikyti modifikuotą vertės grandinės ir elementaraus valdymo ciklo modelį domeno žinių turiniui aprašyti bei atnaujinti.
3. Sukurti domeno valdymo žinių turinio identifikavimo ir žinių atnaujinimo kompiuterizuoto proceso modelį bei informacijos sistemos prototipą.
4. Pritaikyti žinių turinio identifikavimo ir atnaujinimo metodą studijų programos reikalavimų analizei bei programos turinio atnaujinimui, pasiūlyti sistemos taikymo metodiką.
5. Atlikti sukurto metodo ir CASE priemonių taikymo eksperimentinį tyrimą, įvertinant atnaujintų domeno žinių kokybę.

Darbo mokslinis naujumas. Buvo gauti šie nauji rezultatai:

- sukurtas dalykinės srities (domeno) valdymo žinių modeliavimo metodas, grindžiamas veiklos valdymo informacinėmis sąveikomis;
- modifikuotos vertės grandinės ir elementaraus valdymo ciklo pagrindu sukurtas domeno valdymo žinių turinio identifikavimo bei žinių atnaujinimo kompiuterizuoto proceso modelis;
- realizuojant šį modelį ir jo algoritmus, žiniomis grindžiamoje CASE sistemoje sukurtas veiklos žinių posistemis, atliekantis dalykinės srities žinių saugyklos funkciją;
- sukurtas metodas ir sistema aprobuoti pasirinktoje dalykinėje srityje – aukštojo mokslo studijų programų kūrimas bei atnaujinimas.

Asmeninis indėlis. Esminį autoriaus indėlį sudaro:

a) valdymo funkcinių priklausomybių, sudarančių informacinį grįžtamąjį ryšį, sekos apibrėžimas žinių modeliavimo ir valdymo srityje;
b) sukurtas modifikuotas vertės grandinės ir elementaraus valdymo ciklo modelis valdymo žinių atnaujinimo procesui aprašyti;
c) sukurtas kompiuterizuoto proceso modelis ir žiniomis grindžiama CASE sistema.

Darbo rezultatų praktinė reikšmė. Pasiekti šie praktinę vertę turintys rezultatai:

- sukurta valdymo žinių kompiuterizuota priemonė, kuri gali būti taikoma ne tik pasirinktai dalykinei sričiai atnaujinti, bet ir kitose srityse, pavyzdžiui: įmonių žinių bazėms kurti ir atnaujinti, BI priemonėms tobulinti, įdiegiant žinių bazes, vertinti sukauptų repozitorijų turinio pilnumą;
- metodas ir kompiuterizuota priemonė panaudota studijų programos žinių turiniui patikrinti bei atnaujinti;
- sukurti „studijų programos sudarymo / vykdymo valdymo“, „turimo žinių modelio struktūrinimo valdymo“, „dalykinės srities žinių modelio sudarymo valdymo“ ir „žinių modelio atnaujinimo valdymo“ informaciniai DVGM modeliai. Gautieji rezultatai gali būti naudojami tolesniuose nagrinėjamo domeno proceso valdymo ir žinių turinio tobulinimo modelių tyrimuose;
- darbo rezultatai panaudoti sukuriant ir įgyvendinant studijų programos reikalavimų analizės sistemą vykdant projektą „Informacinių technologijų

srities magistrantūros studijų programų modernizavimas, plėtra ir mobilumo užtikrinimas“ (SFMIS kodas: BPD2004-ESF-2.5.0-03-05/0063, paramos sutarties Nr. ESF/2004/2.5.0-03-407/BPD-178);

- pasiūlyta sistemos taikymo metodika, leidžianti susisteminti, analizuoti, lokalizuoti ir vertinti reikalavimus kuriamoms aukštojo mokslo studijų programoms, modernizuoti šių studijų programų struktūrą ir atnaujinti žinių turinį;
- atnaujinta informatikos studijų programos struktūrą ir žinių turinys sukurto mokslinio metodo pagrindu.

Ginamieji teiginiai:

1. Sukurtas metodas, leidžiantis sudaryti analizuojamo domeno valdymo informacinių sąveikų dviejų lygmenų (granuliuotą) žinių aprašymo modelį.
2. Pasiūlytas hierarchinis modifikuotas vertės grandinės ir elementaraus valdymo ciklo modelis leidžia struktūrizuoti domeno žinių identifikavimo bei atnaujinimo procesus.
3. Žinių atnaujinimo kompiuterizuoto proceso modelis leidžia sukurti žiniomis grindžiamą CASE sistemą, kuri gali būti taikoma judrių (greitai besikeičiančių) dalykinių sričių (ne tik studijų programų) žinių turiniui atnaujinti.
4. Darbe sukurta studijų programos reikalavimų analizės ir žinių turinio atnaujinimo sistemos taikymo metodika įgalina susisteminti, analizuoti, lokalizuoti ir vertinti reikalavimus kuriamoms aukštojo mokslo studijų programoms bei lanksčiai atnaujinti šių studijų programų struktūrą ir žinių turinį.

Tyrimo metodika. Darbe taikomi teoriniai ir eksperimentiniai tyrimo metodai. Dalykinės srities žinių turinio analizės tyrimas apima sisteminę mokslinės literatūros analizę, veiklos modeliavimo, žinių valdymo modeliavimo srityse apžvelgtų metodų apibendrinimą, tradicinius veiklos modeliavimo metodus (DFD, IDEF0, IDEF3, BPMN ir kitų metodų savybių analizę), kokybinės analizės (koncepcinių žemėlapių) taikymą, žiniomis grindžiamo veiklos modeliavimo metodus, lyginamosios analizės ir apibendrinimo metodų taikymą. Teoriniams ir eksperimentiniams tyrimams atlikti buvo taikomos CASE priemonės (*IBM Rational RequisitePro™*, *MagicDraw™*, *MS Visio™*) ir kita kompiuterinė programinė įranga: koncepcinių žemėlapių įrankis (*Mindjet MindManager Pro™*), duomenų bazių valdymo sistemos (*MS Access™*, *MySQL Server™*).

Darbo rezultatų aprobavimas. Pagrindiniai darbo rezultatai yra publikuoti 12 moksliniuose leidiniuose: 1 straipsnis spausdintas ISI Web of Science duomenų bazėje referuojamuose ir turinčiuose citavimo indeksą leidiniuose, 1 straipsnis – kituose ISI duomenų bazėje referuojamuose leidiniuose (Proceedings ir kt.), 5 straipsniai – tarptautinėse duomenų bazėse referuojamuose periodiniuose leidiniuose, 1 straipsnis – Lietuvos recenzuojamuose moksliniuose periodinėse leidiniuose, 1 straipsnis – tarptautinių mokslinių konferencijų leidiniuose, 1 tezės – tarptautinių mokslinių konferencijų leidiniuose, 2 straipsniai – respublikinių mokslinių konferencijų leidiniuose. Tyrimų rezultatai buvo pristatyti ir aptarti 7 mokslinėse konferencijose Lietuvoje ir užsienyje.

Disertacijos apimtis ir struktūra. Disertaciją sudaro santrumpų ir terminų žodynėliai, įvadas, penki skyriai, išvados, naudotos literatūros sąrašas, publikacijų sąrašas ir 16 priedų. Skyriai yra suskaidyti į poskyrius, o kai kurie poskyriai – į skyrelius.

Įvadiniame skyriuje aptariama tiriamoji sritis, problemos aktualumas, darbo mokslinė reikšmė, aprašomas tyrimo objektas, formuluojami darbo tikslas bei uždaviniai, pateikiami darbo mokslinis naujumas, asmeninis indėlis, darbo rezultatų praktinė reikšmė, ginamieji teiginiai, aprašoma tyrimų metodika. Įvado pabaigoje pristatomos disertacijos tema autoriaus paskelbtos publikacijos ir pranešimai konferencijose bei disertacijos struktūra.

Pirmasis skyrius skirtas dalykinės srities modeliavimo metodų apžvalgai žinių turinio modeliavimo požiūriu. *Antrajame skyriuje* pateikiamas sukurtas dalykinės srities valdymo žinių modeliavimo metodas, grindžiamas veiklos valdymo informacinėmis sąveikomis. Žinių turiniui aprašyti bei atnaujinti taikomas modifikuotas vertės grandinės ir elementaraus valdymo ciklo modelis. Sukurtas domeno valdymo žinių turinio identifikavimo ir žinių atnaujinimo kompiuterizuoto proceso modelis bei informacijos sistemos prototipas aprašomi *trečiame skyriuje*. *Ketvirtame skyriuje* pristatoma studijų programos reikalavimų analizės ir žinių turinio atnaujinimo sistema bei jos taikymo metodika. *Penktas skyrius* skirtas sukurto dalykinės srities valdymo žinių modeliavimo metodo aprobacijai, pateikiami ir aptariami eksperimentinio tyrimo rezultatai. Kiekvieno skyriaus pabaigoje pateikiamas apibendrinimas ir formuluojamos išvados. Disertacijos pabaigoje pateikiamos bendrosios išvados, taip pat priedai.

Bendra disertacijos apimtis 177 puslapių (A4 formato), kuriuose pateikiama 13 lentelių ir 105 paveikslai. Rašant disertaciją naudotasi 214 literatūros šaltiniais.

Padėka. Nuoširdžiai dėkoju mokslinio darbo vadovui prof. dr. (HP) Sauliui Gudui už suteiktą galimybę tobulėti, nuoseklų vadovavimą ir pagalbą rengiant šią disertaciją. Ypatingą padėką reiškiu konsultantui prof. dr. Vitalijui Denisovui už vertingas mokslines konsultacijas, nuolatinį palaikymą ir domėjimąsi darbų sėkme bei visokeriopą pagalbą.

Trumpos žinios apie autorių. Jurij Tekutov gimė 1983 m. Klaipėdoje. 2006 m. Klaipėdos universiteto Gamtos ir matematikos mokslų fakultete įgijo informatikos bakalauro laipsnį. 2008 m. Klaipėdos universitete įgijo informatikos magistro laipsnį. 2008–2012 – Vilniaus universiteto Kauno humanitarinio fakulteto doktorantas. Pagrindinė veikla: Klaipėdos universiteto Jūrų technikos fakulteto Informatikos inžinerijos katedros ir Klaipėdos valstybinės kolegijos Technologijų fakulteto Informacinių technologijų katedros lektorius. 2008 m. dalyvavo projekto „Informacinių technologijų srities magistrantūros studijų programų modernizavimas, plėtra ir mobilumo užtikrinimas“ veikloje ir vasaros stovykloje. 2009 m. dalyvavo vasaros stovykloje “Methodologies for Interactive Networked Enterprises – MINE” Nidoje. El. paštas: jurij@ik.ku.lt.