

Method of Educational Domain Knowledge Content Updating Based on Management Information Interactions

Jurij TEKUTOV^{1, 2, 4}, Saulius GUDAS³, Vitalijus DENISOVAS^{2, 4},
Arūnas ANDZIULIS², Julija SMIRNOVA¹

¹ Klaipeda State University of Applied Sciences, Faculty of Technologies, Information Technologies Department, Bijunu St. 10, LT-91223 Klaipeda, Lithuania

² Klaipeda University, Faculty of Marine Technologies and Natural Sciences, Informatics and Statistics Department, Bijunu St. 17, LT-91225 Klaipeda, Lithuania

³ Vilnius University, Faculty of Mathematics and Informatics, Institute of Data Science and Digital Technologies, LT-04812 Vilnius, Lithuania

⁴ Lithuania Business University of Applied Sciences, Informatics Department, Turgaus St. 21, LT-91249 Klaipeda, Lithuania

jurij.tekutov@yahoo.com, saulius.gudas@mii.vu.lt,
vitalijus.denisovas@ku.lt, arunas.iik.ku@gmail.com,
j.smirnova@kvk.lt

Abstract. The hierarchical Detailed Value Chain Model and Elementary Management Cycle model of educational domain knowledge content updating is formally described and computerized process measures are proposed. The paper provides a method for updating the knowledge of the analyzed domain, referred to as “enterprise domain”, based on enterprise modelling in terms of management information interactions. A method, whose formal DVCM and EMC descriptions are provided in BPMN notation, was designed, allowing to develop a two-level (granular) model for describing knowledge of educational domain management information interactions. In implementing this model and its algorithms in technological terms, a subsystem of enterprise knowledge has been created in a knowledge-based CASE system (computerized knowledge-based IS engineering), performing the function of a domain knowledge database. This method is suitable to discover content related to the business enterprise activities and enterprise software development. The main outcome of enterprise modeling approach is the effectiveness of discovering of the deep knowledge, when education domain frameworks are applicable. 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.

Keywords: knowledge-based IS engineering, the Detailed Value Chain Model, the Elementary Management Cycle, development of the problem (educational) domain model

Introduction

Meeting modern, constantly evolving educational enterprise needs requires enterprise integration with information technologies and computerized information systems (IS) that would be able to adapt to organizations' enterprise changes. In the context of this paper enterprise is understood as a educational institution.

Among different IS development 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 (Trask and Roman, 2009). The aim of 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 (Christensen et al., 1995; Gudas, 2012a) 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 (Gudas, 2012b). 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 (Caplinskas, 2002) from similar areas, such as e-learning, Business Intelligence systems (Hagerty, Sallam and Richardson, 2012; Henschen, 2011), business rule modelling (Andreescu and Mircea, 2009).

The analysis of IS engineering stages (1. traditional IS engineering, 2. computerised software engineering, 3. computerised IS engineering (including enterprise modelling), 4. knowledge-based IS engineering) reveals the key features of the variations in the IS engineering process (methods, knowledge sources, software types). Hence, it enables to define the key characteristics of the knowledge-based IS engineering methodology and the composition of the basic components. In traditional computerized IS engineering a system is created empirically, by first collecting, analysing and specifying consumer needs. Works linked to knowledge-based IS development allow to take software development to a higher level of abstraction, partly automate these processes and use the models in other systems. Business activity modelling, the improvement of knowledge formalization and the application of the conceptual modelling in developing the IS are the key things for the further development

of the IS engineering methods. 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 (Figure 1): 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 paper. The development of knowledge-based IS engineering methods requires an internal approach to business process modelling, which should be based on the internal business process logic – the causal links between business processes, i.e. necessary and sufficient components of activity elements and interactions between them (the activity goals of material processes, information processes and structural information units). Therefore, the above-mentioned diagram shows the managerial information exchange (in proposed modified Value Chain Model).

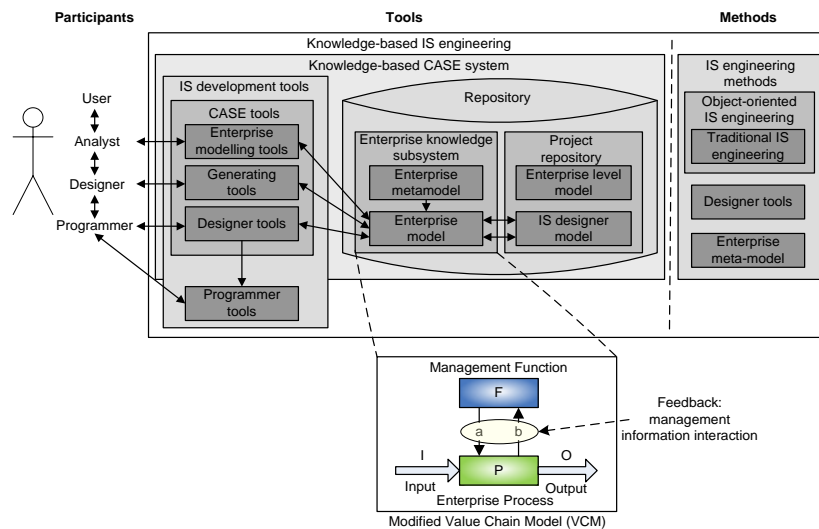


Figure 1. Knowledge-based IS engineering process enterprise structure with proposed model

In our previous work (Tekutov, Gudas and Denisovas, 2012) it was proposed that the development of these systems should be performed by focusing on Value Chain Model (VCM) (Porter, 1998), 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 (Gudas and Lopata, 2011), a Detailed Value Chain Model (DVCM) is created. Hence, knowledge-based IS development is based on knowledge on enterprise

management, which defines fundamental attributes of enterprise management information interactions. In previous work (Gudas, 2009) was shown the enterprise management information interaction is specified as an Elementary Management Cycle (EMC), linking enterprise management function and process by a feedback loop. The novelty of the work is the application of the expanded VCM and the EMC to manage operational knowledge; after defining the assessment criteria, such management model allows to objectively assess and constantly improve the knowledge content of the subject area.

This paper elaborates a proposed domain management knowledge modelling method, which is based on formal DVCM and EMC descriptions presented in a BPMN notation. Educational 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. The 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.

The creation of knowledge content specification includes the collection of domain knowledge, its preparation and analysis, verification and validation. In this paper 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.

The main aim of this paper is to develop the educational domain knowledge content updating method based on formal descriptions of enterprise management information interactions.

The remainder of the paper is structured as follows: Section 2 reviews domain modelling methods in terms of knowledge content modelling. Section 3 introduces the created educational domain management knowledge modelling method, based on enterprise management information interactions. The application of a Detailed Value Chain and Elementary Management Cycle model for knowledge content description and updating is demonstrated. The created educational domain management knowledge content identification and knowledge updating computerized process model are laid out in Section 4.

1. The analysis of current domain modelling methods in terms of knowledge content modelling

Previous works. 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 (Caplinskas and Vasilecas, 2002). 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 developers 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 authors of this paper) have suggested to automate this process by creating a study programme requirement engineering system on the basis of CASE measures (Denisovas, Gudas and Tekutov, 2010). With a direct involvement of the authors, such a system has been created and applied for the development of new study programmes (Tekutov, Gudas, Denisovas and Tekutova, 2012; Tekutov, Gudas and Denisovas, 2012). 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 paper, a DVCN 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 educational domain enterprise model.

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

Therefore, there is a need to systematize the variety of these methods in order to select the most convenient way of modelling management activities. To facilitate this selection a concept map of different enterprise modelling techniques (languages, notations) (Henderson and Venkatraman, 1990; Mertins and Jochem, 2005; Millette, 2005; OMG consortium, 2009; Scheer and Schneider, 2006) was designed (Figure 2).

Enterprise models are divided into the following categories: business process models, data flow models, data models, business objective models, control flow models, management process models, knowledge management processes (Giaglis, 2001). Widely known enterprise modelling methods are structural-functional modelling (Data Flow Diagrams or DFD), business process modelling languages, BPMN (Business Process Modeling) notation, IDEF (Integration of computer aided manufacturing DEFinition) set

of standards and enterprise modelling language UEML (Unified Enterprise Modelling Language), and business architecture modelling language UPDM (The Unified Profile for DoDAF/MoDAF). Well known enterprise modelling methodologies and standards

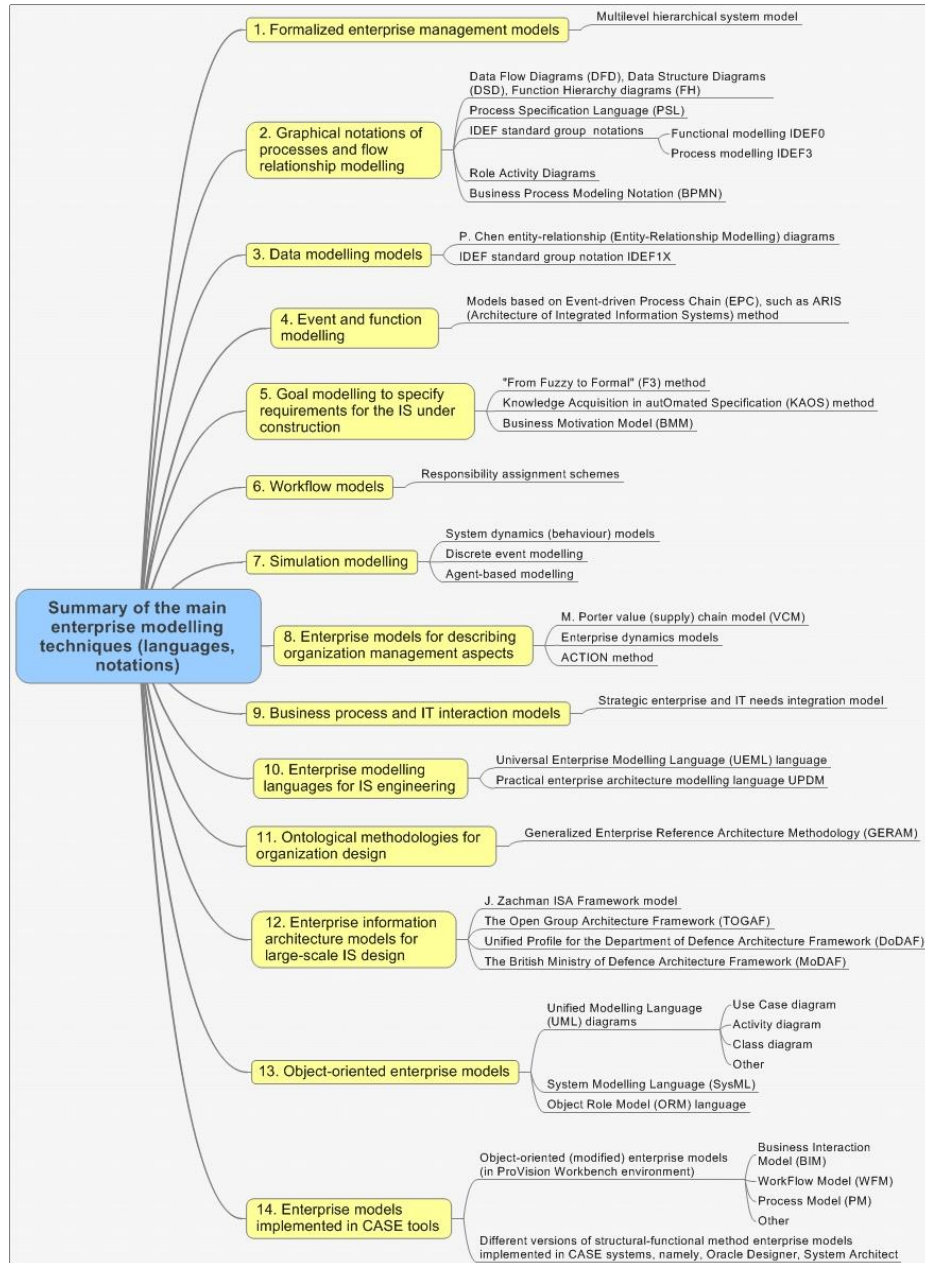


Figure 2. Concept map of enterprise modelling techniques (languages, notations) (using *Mindjet MindManager ProTM* tool)

are CIMOSA, GERA, EPC, DoDAF, MODAF, MDA (Morkevičius and Gudas, 2011). However, amongst them one can rarely find those that would model enterprise in terms of management, i.e. performance management information components and their interactions.

When comparing modern enterprise modelling methods (languages, notations), it is becoming clearer that 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 concept 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 domain management, it is becoming clearer that only a few methods can actually evaluate management information interactions that are necessary to manage domain 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 (Figure 3) (Ahmed and Sharma, 2006; Bandarian, 2008; Beard, Schwieger and Surendran, 2010; Chen, Yang and Lin, 2004; Cummins, 2010; Gudas and Brundzaitė, 2005; Gudas, Lopata and Skersys, 2005; Gui and Ye, 2008; Koc and Bozdog, 2017; Lee and Han, 2009; Landry, Amara, Pablos-Mendes, Shademani and Gold, 2006; Najmaei, and Sadeghinejad, 2009; V. Pathak and K. Pathak, 2010; Powell, 2001; Roko and Opusunju, 2016; Saha, 2012; Stewart, 1997; Sweeney, 2009; Williamson, Harrison and Jordan, 2004; Wong, 2004).

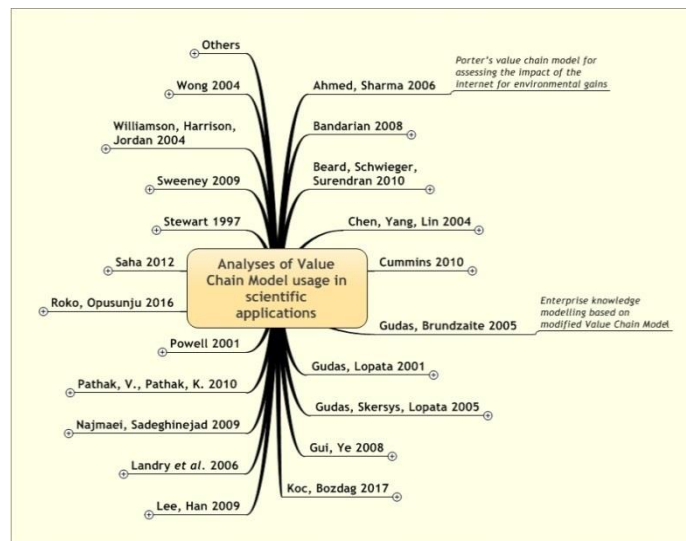


Figure 3. Conceptual map of Value Chain Model in scientific applications

The concept map (Fig. 3) illustrates examples of the application of VCM in scientific research (other examples of the analysis can be accessed by clicking on the plus sign in the concept map in *Mindjet MindManagerTM* environment): Porter's value chain model for assessing the impact of the internet for environmental gains (Ahmed and

Sharma, 2006); Enterprise knowledge modelling based on modified Value Chain Model (Sweeney, 2009); others.

The structured DVCM is used to identify the information transactions between *Management Functions* and *Enterprise Processes*. Detailed Value Chain Model (DVCM) embodies a procedural approach to enterprise consisting of: primary activities (operational processes) and support activities (operational functions). The formal description of DVCM is as follows:

$$DVCM = \{(F_i) \times (P_j)\} \quad (1)$$

where: $(F_1, \dots, F_i, \dots, F_n)$ – a set of *Enterprise Management Functions*, $(P_1, \dots, P_j, \dots, P_m)$ – a set of *Enterprise Processes*, \times – relationship between a set of *Management Functions* $\{F\}$ and a set of *Enterprise Processes* $\{P\}$ (management information transactions $(F_i \times P_j)$).

The interrelationship between primary and secondary business processes explored in Gudas, Lopata (2001) 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 paper presents more detailed content of *Function* F_i since it defines a sequence of definite types of interacting information activities directed to control *Process* P_j .

In brief, the concept of 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 decomposition of the information interactions $(F_i \times P_j)$ in the DVCM between the *Management Function* F_i and *Enterprise Process* P_j is defined as Elementary Management Cycle (EMC) on the lower level in Figure 4. 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* F_i and the *Process* P_j . The two levels of granularity: 1) the Detailed VCM (DVCM) and 2) the Elementary Management Cycle (EMC).

A detailed model of the informational interactions between the *Function* F_i and *Process* P_j can be defined by:

$$EMC(F_i, P_j) = \left(\begin{array}{l} P_j(A, G) \rightarrow IN(A, B, G) \rightarrow P(B, C, G) \rightarrow \\ M(C, D, G) \rightarrow RE(D, V, G) \rightarrow P_j(V, G) \end{array} \right) \quad (2)$$

where A – state attributes of the process P_j , needed in terms of G ; B – systemized (interpreted) primary data, needed for the enterprise management function F_i 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 P_j , formed by the realization process RE .

The expanded DVCM and EMC was printed in the general format (Tekutov et al., 2012; Tekutov, Gudas and Denisovas, 2012). As proposed in work (Tekutov, 2013), 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 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_i . The *Management Function* F_i consists of a sequence of definite types of goal-driven information transformation activities (steps of EMC) aimed to control a state of an *Enterprise Process* P_j . For example, in the structure of an EMC: $P_j (A, G)$, $P_j (V, G)$ – *Technological Process* (managed object), its input (I) and output (O) are *Material Flows*. *Material Flows* are defined by state attributes of a specific *Process* P_j , which are necessary to perform a specific enterprise management *Function* F_i in combination with enterprise *Goals* (G).

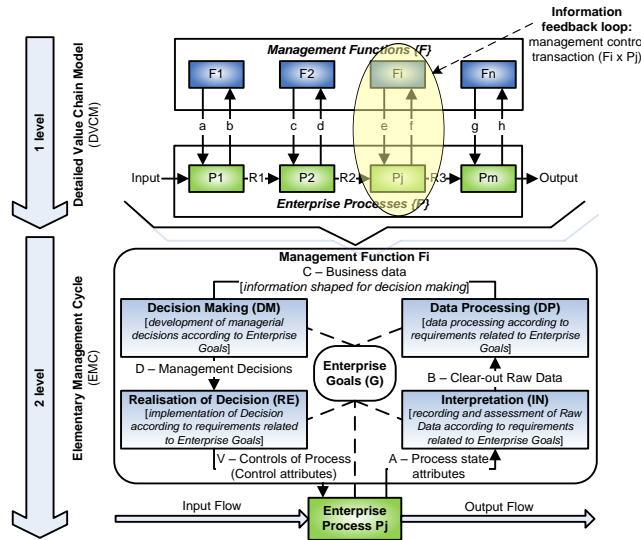


Figure 4. The formalized model of the interaction between the Function F_i and Process P_j (Elementary Management Cycle – EMC) in modified (structured) DVCm

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

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

where: $FD1 = (S_i \rightarrow IN(G) \rightarrow S_j)$ – *Functional Dependence* of the *Interpretation* step, S_i – *Business Process* (technological process) state attributes ($S_i = A$), S_j – output attributes of the *Interpretation* step ($S_i = B$); $FD2 = (S_j \rightarrow DP(G) \rightarrow S_n)$ – *Functional Dependence* of the *Data Processing* step, S_i – output attributes of the *Interpretation* step are an input ($S_j = B$), S_n – *Data Processing* output ($S_n = C$); $FD3 = (S_n \rightarrow DM(G) \rightarrow S_m)$ – *Functional Dependence* of the *Decision Making* step, S_n – output attributes of the *Data Processing* step are an input of *Decision Making* ($S_n = C$), S_m – *Decision Making* output ($S_m = D$); $FD4 = (S_m \rightarrow RE(G) \rightarrow S_k)$ – *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:

$$MFD = \left\{ \begin{array}{l} FD1 = (Si \rightarrow IN(G) \rightarrow Sj); \\ FD2 = (Sj \rightarrow DP(G) \rightarrow Sn); \\ FD3 = (Sn \rightarrow DM(G) \rightarrow Sm); \\ FD4 = (Sm \rightarrow RE(G) \rightarrow Sk); \\ FD5 = (Sk \rightarrow P(G) \rightarrow Si) \end{array} \right\} \quad (4)$$

Based on this general model, the authors have proposed a specialised model for particular problem domain.

2. Educational domain management knowledge modelling method, based on enterprise management information interactions

Educational 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 educational domain management information interactions, to form current knowledge and domain knowledge models and to perform the analysis of knowledge models. The computerisation of the organisational activities involves an examination of the information interactions between the management functions and the business processes, which requires the decomposition of the value chain elements, i.e. the VCM is modified and a Detailed Value Chain Model (DVCM) is created. Thus, the knowledge-based IS development is performed based on that business management knowledge which defines the key features of business management information interactions. A business management information interaction is specified as an Elementary Management Cycle (EMC), which links the business management function and the business process through a feedback circuit. A two-level subject area management knowledge model was selected. The two levels of knowledge description are:

1. First level of detail – DVCM that identifies information interactions.
2. Second level of detail – a detailed model of each management information interaction by applying EMC.

Such model is believed to be sufficient in all cases of activity knowledge modelling. The decomposition of the organisational system business management functions and the business processes can be continued up to the required level of detail of the business management function.

Paper comprise known approaches to create educational domain content knowledge models updated existing. Model is created on to levels that combines DVCM and EMC model presenting the transformation of the existing and the domain knowledge models on the basis of management information interactions”, BPMN notation (Figure 5).

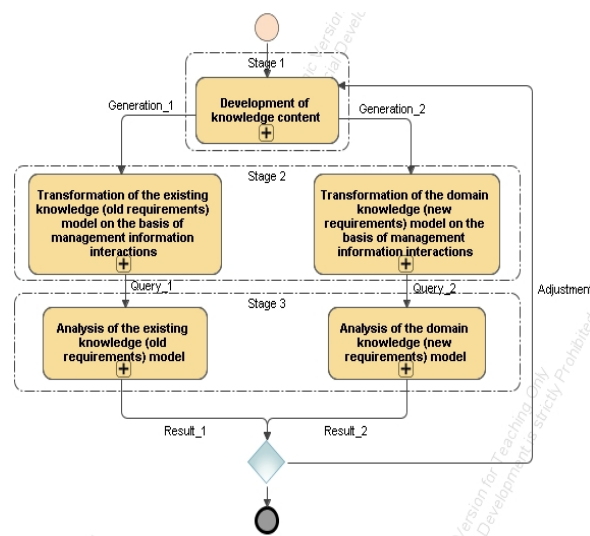


Figure 5. A principal scheme of educational domain management knowledge modelling method (BPMN notation)

Development of knowledge content (Figure 6) involves structural, functional, quality and other requirements.

The method is elaborated in a specific domain – higher education study programme design and updating. 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 (component_1 in Fig. 6). 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 (component_2 in Fig. 6).

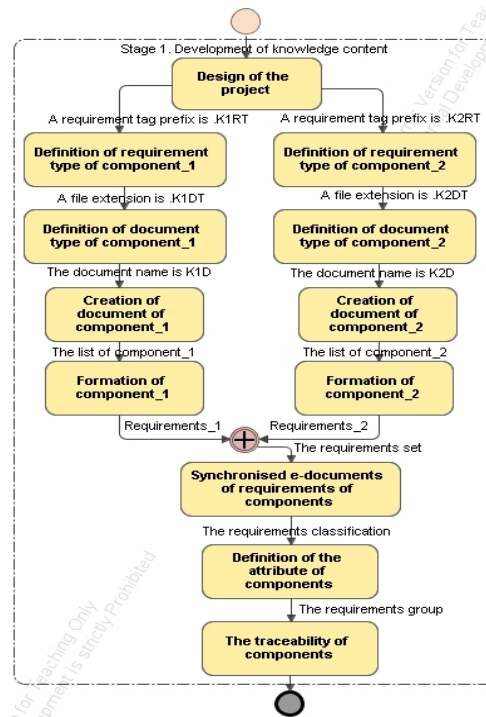


Figure 6. Part of the method (Stage 1 “Development of knowledge content”, BPMN notation, using *MagicDrawTM* tool)

Stage 1 “Development of knowledge content” specifications:

1. *Design of the project* – in CASE tool IBM Rational RequisiteProTM.
2. Each requirement belongs to a particular requirement type: 2a. *Definition of requirement type of component_1* (a requirement tag prefix is .K1RT – knowledge_1 requirement type); 2b. *Definition of requirement type of component_2* (a requirement tag prefix is .K2RT – knowledge_2 requirement type).
3. Each document belongs to a particular document type. The document type defines a template that is applied to a document: 3a. *Definition of document type of component_1* (a file extension is .K1DT – knowledge_1 document type); 3b. *Definition of document type of component_2* (a file extension is .K2DT – knowledge_2 document type).
4. Creating a document from a document type that is already in the project: 4a. *Creation of document of component_1* (the document name is K1D – knowledge_1 document); 4b. *Creation of document of component_2* (the document name is K2D – knowledge_2 document).
5. According a study programme is composed of 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: 5a. *Formation of component_1* (the list from component_1); 5b. *Formation of component_2* (the list from component_2).

6. *Synchronised e-document of requirements of components* (the requirements set).

7. *Definition of the attribute or components* (the requirements classification) – every requirement has associated attributes. Attributes provide information for managing the requirement.

8. *The traceability of components* (the requirements group) – a traceability item is project element that needs to be explicitly traced from another textual or model item to keep track of the dependencies between them.

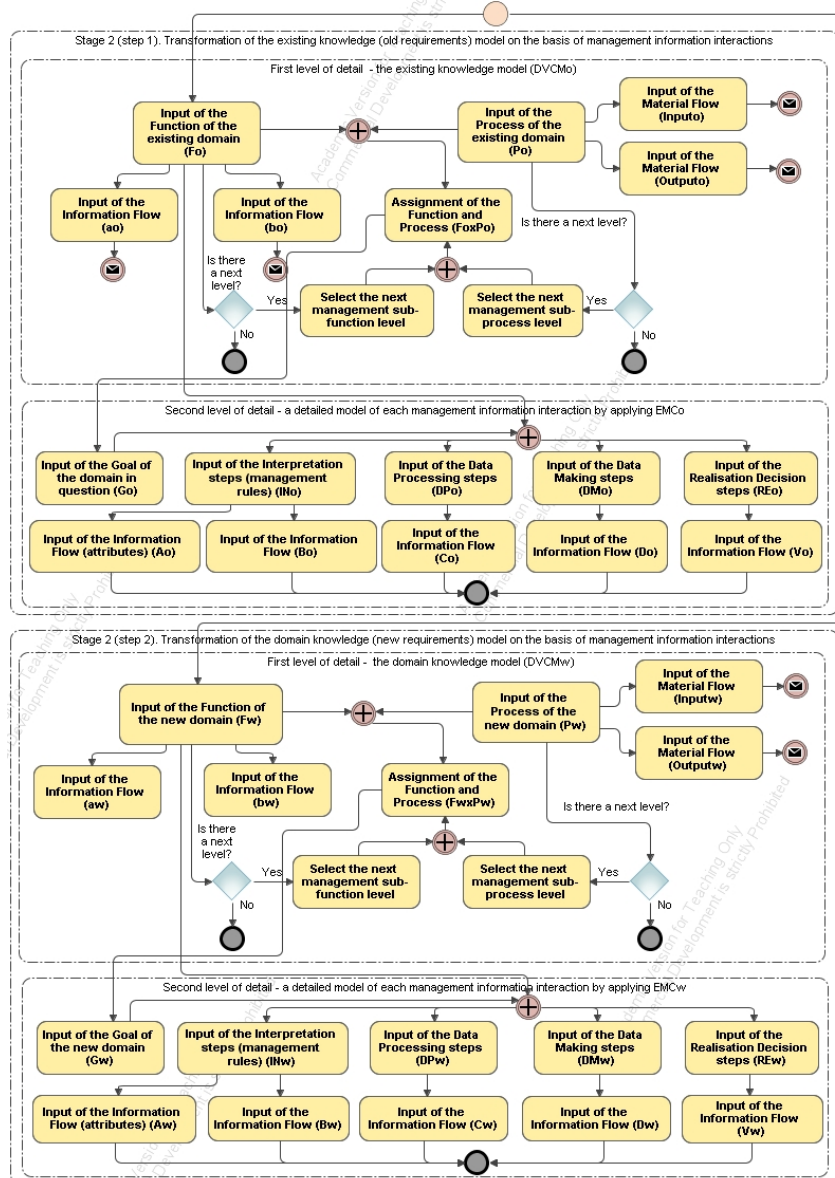


Figure 7. Part of the method (Stage 2 “Transformation of the existing and the domain knowledge models on the basis of management information interactions”, BPMN notation)

Designing a knowledge base on the basis of DVCM and EMC. Management knowledge model based on DVCM and EMC is created in two stages (Figure 7):

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*:

1. Analyze the general case of the *Detailed Value Chain Model* (DVCM) which is the formalized DVCM structure.
2. Verify that *management Function* hierarchy exists.
3. Verify that *Process* hierarchy exists.
4. Verify that data (information attributes) related to each intersection between a *management Function* and a *Process* exists.

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:

1. Select a specific *management Function* and *Process* pair (intersection) identified with DVCM, that will further be modelled as a managed process (formally described EMC).
2. 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.
3. Illustrate DVCM and EMC through domain entity class model.
4. Develop a prototype of a knowledge base.

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 (Figure 8). Within the selected CASE tool (*IBM Rational RequisiteProTM*) 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.

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 current 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 its structural components (study blocks and modules).

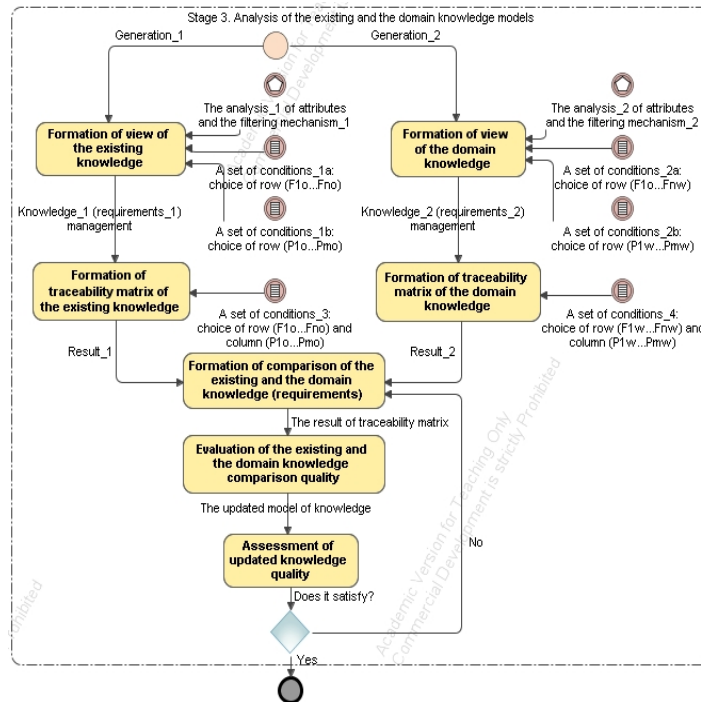


Figure 8. Part of the method (Stage 3 “Analysis of the existing and the domain knowledge models”, BPMN notation)

Modification of a structured Value Chain Model to identify educational domain knowledge content. To reflect these needs the authors have proposed a new problem domain life cycle model, which is based on modified VCM. This specialised model is presented in Figure 9.

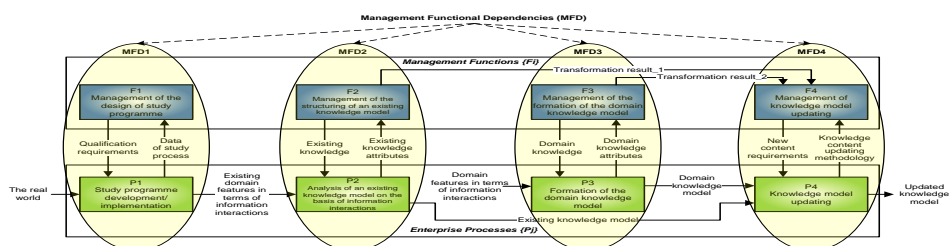


Figure 9. Informational model of educational domain knowledge content identification process

Each interaction between management *Function* F_i and *Process* P_j ($F_i \times P_j$) 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* $F_2 =$ “*Management of the structuring of an existing knowledge model*” and the *Process* $P_2 =$ “*Analysis of an existing knowledge model on the basis of information interactions*” (see Fig. 9) is presented in Figure 10.

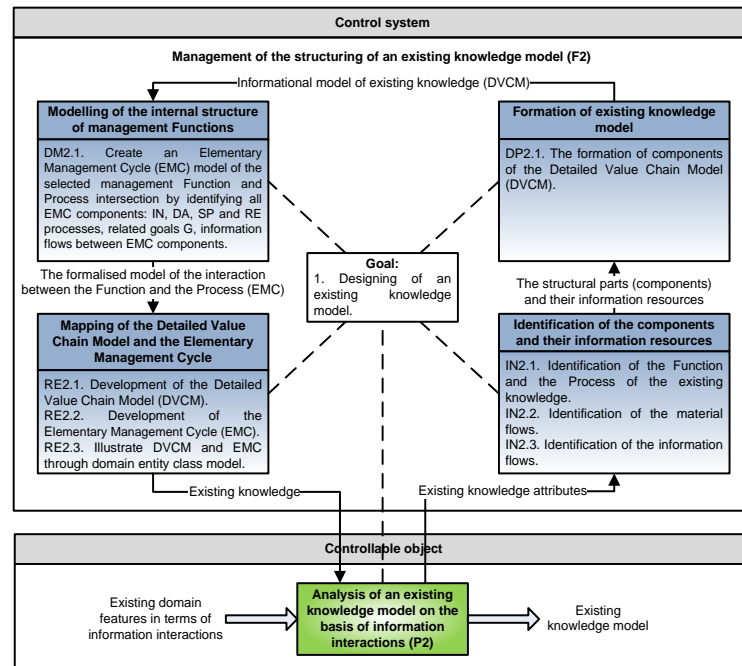


Figure 10. The Formalised Model of the Interaction between the *Function* $F_2 =$ “*Management of the structuring of an existing knowledge model*” and the *Process* $P_2 =$ “*Analysis of an existing knowledge model on the basis of information interactions*”

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. Educational domain management knowledge content identification and knowledge updating computerized process model

This section provides a principal scheme of knowledge content identification and educational 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 11.

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*.

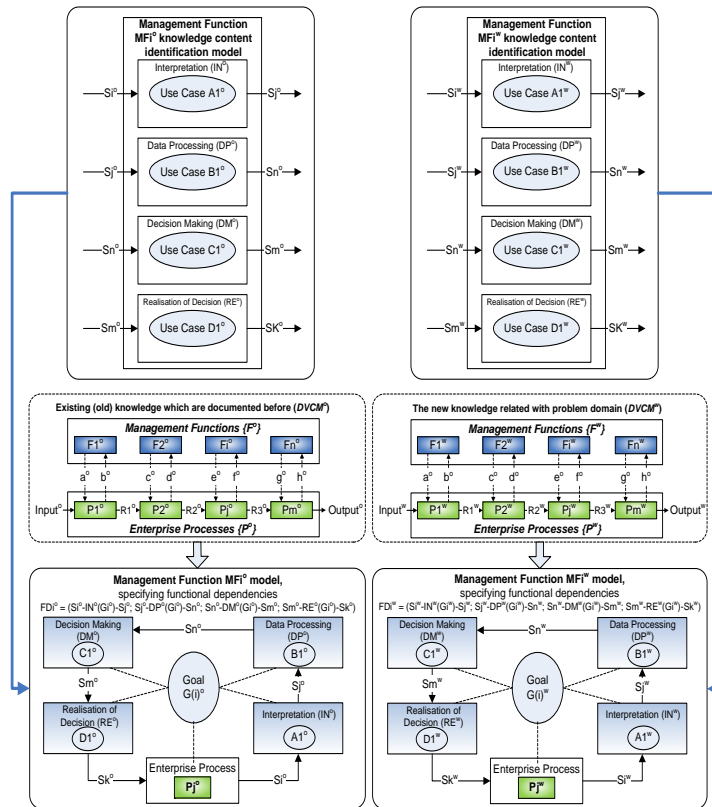


Figure 11. A principal scheme of knowledge content identification and domain knowledge updating model (based on enterprise management function)

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 DVCM in Figure 12.

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

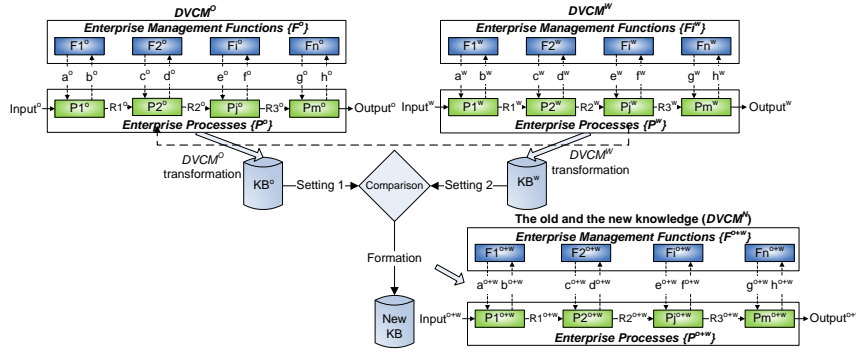


Figure 12. The identification of the existing and the new knowledge (here: $DVCM^o$ – the old knowledge; $DVCM^w$ – the new knowledge; $DVCM^N$ – the renewed domain content model)

A technique of knowledge analysis is used for comparison between $DVCM^o$ and $DVCM^w$. Comparison of these two ($DVCM^o$ and $DVCM^w$) domain models allow to identify the changes of problem domain content (i.e. identify the actual changes of knowledge about domain) and to construct a derivative new Detailed VCM ($DVCM^N$) encompassing all actual domain knowledge components. Afterwards the $DVCM^N$ is used for identification of new knowledge requirements. Traceability Matrix shows the relation of two knowledge sets (Figure 13).

There are several situations that can be identified: 1) *the sameness* of knowledge (equal requirements, knowledge remains *unchanged*, denoted by S); 2) *the supplementation* of existing knowledge with lacking new subject area knowledge (current knowledge only partially covers new requirements and should be *supplemented*, denoted by P); 3) *the addition* (inclusion) of new domain knowledge (denoted by N); 4) *the exclusion* of present knowledge from the current domain (no longer meets subject area requirements, competency becomes less important, denoted by D).

Identifying the discrepancy between the existing and new knowledge.

Traceability matrix is developed to identify the discrepancy between existing knowledge (old requirements) and new knowledge (requirements) and is presented in Figure 14.

Formal description of traceability matrix elements is as follows:

$$R^T = \{S, P, N, D\} \quad (5)$$

Evaluation of current and domain knowledge comparison quality. Starting from 1965 when L. Zadeh introduced the basic principles of fuzzy set theory (Zadeh, 1965) and its application, this method has been evolving ever since. Fuzzy set methods allow to incorporate human experience in management and modelling algorithms (Kim, Wang and Yang, 2005). The application of fuzzy set methods has gained wide recognition and has been used successfully worldwide to optimize sets of human behaviour rules for managing certain processes.

Current and domain knowledge are evaluated based on the created quality model by applying triangular and trapezoidal fuzzy number methods (Ounaies, Jamoussi and Ghezala, 2009). 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 (Zadeh, 1965).

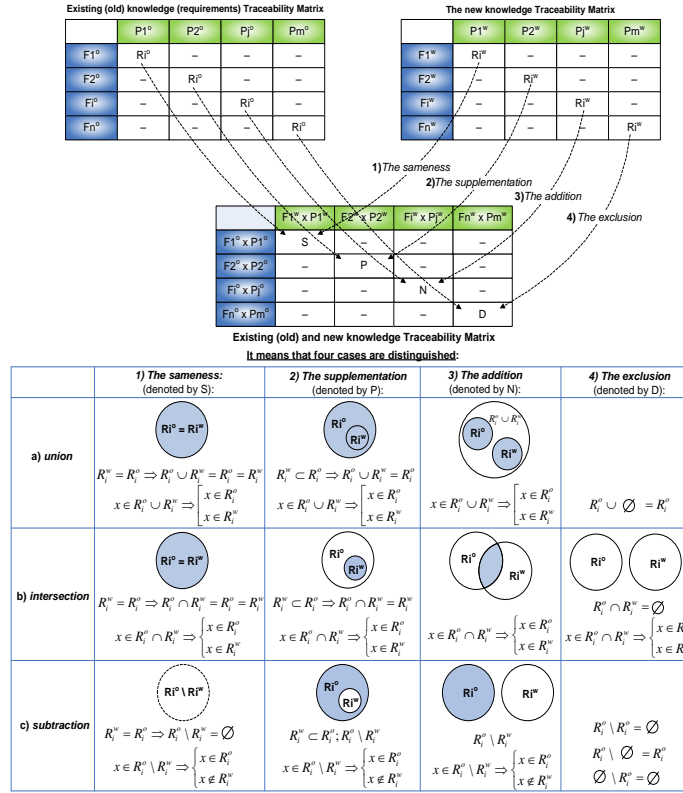


Figure 13. The relations between existing (old) and the new knowledge (requirements) (where $R_1^o, R_2^o, \dots, R_i^o, R_1^w, R_2^w, \dots, R_i^w$ – the existing and new managed process results)

Existing (old) and new knowledge Traceability Matrix

	Pj(G) ^w	A ^w	IN(G) ^w	B ^w	DP(G) ^w	C ^w	DM(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	-	-	-	-	-	-
DP(G) ^o	-	-	-	-	R ^T	-	-	-	-	-
C ^o	-	-	-	-	-	R ^T	-	-	-	-
DM(G) ^o	-	-	-	-	-	-	R ^T	-	-	-
D ^o	-	-	-	-	-	-	-	R ^T	-	-
RE(G) ^o	-	-	-	-	-	-	-	-	R ^T	-
V ^o	-	-	-	-	-	-	-	-	-	R ^T

Figure 14. Traceability matrix for identifying the discrepancy between the existing and new knowledge

As explained by the authors above, fuzzy numbers are an approximation of random numbers. They are characterized by a triad $M = (l, m, u)$, $l \leq m \leq u$, where m is the mode, l is the lower limit, and u is the upper limit.

There are four graphical and analytical expressions for the most commonly used membership functions: triangular, trapezoidal, Gaussian, and \cos^2 functions.

Triangular and trapezoidal fuzzy number methods. These three values also define an analogue of distribution – the so called triangular membership function:

$$\mu_M(x) = \begin{cases} \frac{x-l}{m-l}, & \text{when } x \in [l, m], \\ \frac{x-u}{m-u}, & \text{when } x \in [m, u], \\ 0, & \text{when } x \notin [l, u] \end{cases} \quad (6)$$

Trapezoidal fuzzy numbers are fuzzy numbers defined by four points as follows:

$$\mu_M(x) = \begin{cases} 0, & \text{when } x < a, \\ \frac{x-a}{b-a}, & \text{when } a \leq x \leq b, \\ 1, & \text{when } b \leq x \leq c, \\ \frac{d-x}{d-c}, & \text{when } c \leq x \leq d, \\ 0, & \text{when } x > d. \end{cases} \quad (7)$$

Triangular fuzzy number is illustrated in Figure 15 (a), trapezoidal fuzzy number is illustrated in (b).

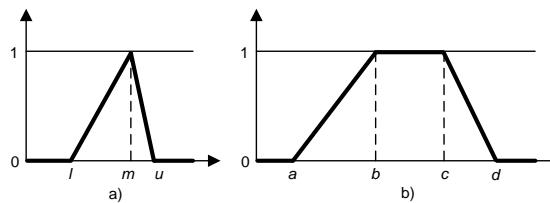


Figure 15. a) Triangular fuzzy number; b) Trapezoidal fuzzy number

Conversion of linguistic variables into triangular and trapezoidal fuzzy numbers is provided in Table 1.

Table 1. Conversion of linguistic variables into triangular and trapezoidal fuzzy numbers

Linguistic variables	Triangular fuzzy numbers	Trapezoidal fuzzy numbers
Excellent	(0,700 0,850 1,000)	(0,800 1,000 1,000 1,000)
Good	(0,525 0,675 0,825)	(0,600 0,800 0,800 1,000)
Fair	(0,350 0,500 0,650)	(0,300 0,500 0,500 0,700)
Poor	(0,175 0,325 0,475)	(0,000 0,200 0,200 0,400)
Bad	(0,000 0,150 0,300)	(0,000 0,000 0,000 0,200)

In this paper, some criteria were formulated to evaluate current and domain knowledge quality, i.e. (1) enterprise process demand, (2) knowledge change speed, (3) Compliance with professional standards and other. The experts used linguistic variables “bad”, “poor”, “fair”, “good”, and “excellent” to establish ratings (values) of the quality criteria. According to (Zhang and Cheng, 1992; Kurilovas, Serikoviene and Vuorikari, 2014), linguistic variables were converted into average triangle fuzzy numbers as follows: “excellent” = 0,850; “good” = 0,675; “fair” = 0,500; “poor” = 0,325; “bad” = 0,150; and trapezoidal fuzzy numbers as follows: “excellent” = 0,100; “good” = 0,800; “fair” = 0,500; “poor” = 0,200; and “bad” = 0,000.

The scalarisation method can be seen as the experts’ additive utility function (a possible decision transforms a multi-criteria task into one criterion task obtained by adding all the criteria ratings (values) together with their weights), represented by the formula:

$$f(X) = \sum_{i=1}^m a_i f_i(X), \quad \sum_{i=1}^m a_i = 1, \quad a_i > 0. \quad (8)$$

Here $f_i(x_i)$ is the rating (i.e. non-fuzzy value) of the criterion i , and a_i are the weights of the quality criteria.

With the use of these two methods, two knowledge sets (KS) have been evaluated – KS1: Current knowledge and KS2: Domain knowledge. In this case, evaluation is understood as a summarized opinion of a specialist group.

For example, comparison of current and domain knowledge by applying triangular fuzzy number method is illustrated in Table 2.

Table 2. Comparison of current and domain knowledge by applying triangular fuzzy number method

No.	Criteria	KS1 ratings	KS2 ratings
1.	Enterprise process demand.	0,675	0,850
2.	Knowledge change speed.	0,675	0,675
3.	Compliance with professional standards.	0,675	0,850
4.	Compliance with market demands.	0,675	0,675
5.	Compliance with student demands.	0,675	0,850
6.	Ability to apply knowledge in practice.	0,500	0,500
7.	Scenario flexibility in terms of content and use of technologies.	0,850	0,850
8.	Sufficiency of equipment and technical measures.	0,500	0,500
9.	Compliance with methodological aspects.	0,850	0,850
10.	Teaching opportunity.	0,675	0,850
In total:		0,6750	0,7450

When using triangular fuzzy numbers, overall quality evaluation results show that KS1 matches 67,50 % of the quality level in comparison with that of “ideal” (100 %), whereas KS2 matches 74,50 %. After applying different weights to knowledge, quality evaluation results show that KS1 matches 69,25 % of quality in comparison with “ideal”, whereas KS2 matches 76,60 %. The results indicate that when applying the triangular fuzzy number method, KS2 receives the best rating when using both equal and different knowledge weights.

When using trapezoidal fuzzy numbers, overall quality evaluation results show that KS1 matches 84,00 % of the quality level in comparison with that of “ideal”,

whereas KS2 matches 92,00 %. After applying different weights to knowledge, quality evaluation results show that KS1 matches 85,20 % of quality in comparison with “ideal”, whereas KS2 matches 93,60 %. The results indicate that when applying the trapezoidal fuzzy number method, KS2 receives the best rating when using both equal and different knowledge weights.

We can see that quality evaluation results are similar between triangular and trapezoidal fuzzy number methods. Trapezoidal fuzzy number results show a slightly larger difference between current and domain knowledge quality evaluation results when applying both equal (8,00 % against 7,00 %) and different knowledge weights (8,4 % against 7,35 %).

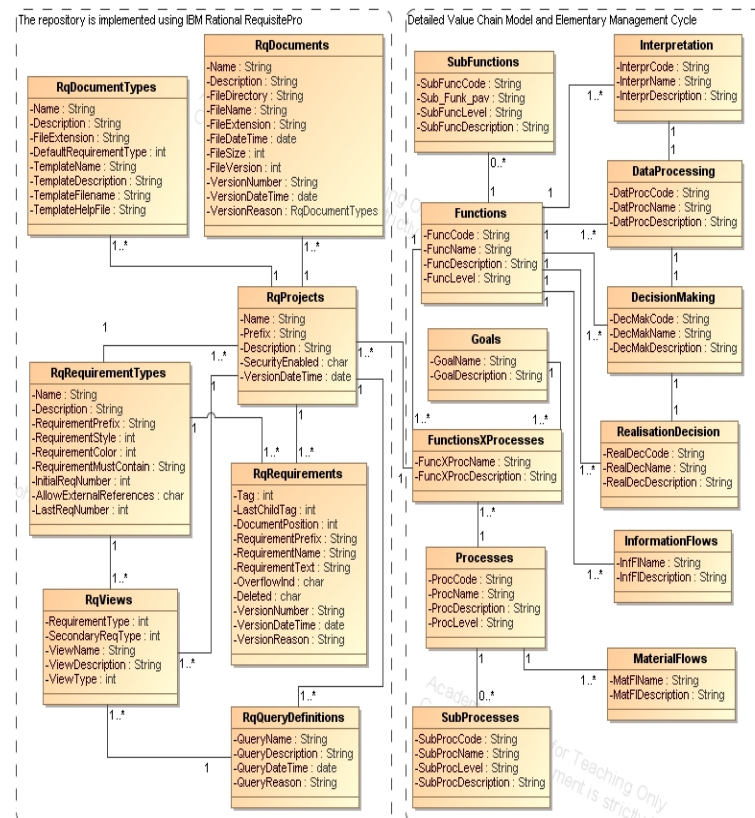


Figure 16. Subject area entity class model (logical data model)

Implementation of the knowledge repository. 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 16 (using *MagicDrawTM* CASE tool). The solution is implemented by extending the requirement management system by knowledge-based components.

Requirement management is performed on the basis of *IBM Rational RequisiteProTM* tool, meanwhile, the structure of a knowledge base is based on Detailed Value Chain and Elementary Management Cycle.

Subject area entity class model of the knowledge repository specifies DVCM 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 DVCM and EMC together with requirements are stored in the knowledge databases and can be used for solving specific problems of expected application areas.

Conclusions and discussions

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

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.

The Detailed Value Chain Model (DVCM) is defined as a problem domain content model. The peculiarity of the DVCM is refinement of the Management Functions {Fi} and Processes {Pj} as an obligatory component types of problem domain knowledge content. Management Functions (F) possess information transformations; while Enterprise Processes (P) are concrete material transformations.

The educational domain management knowledge modelling method was designed, allowing to create a two-level (granular) model for describing knowledge of domain management information interactions: the highest level Detailed Value Chain Model is further elaborated into a set of Elementary Management Cycle models; the chosen hierarchical structure allows to create new structural knowledge models and to update existing ones.

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. 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. 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.

The resulting knowledge-based computerized CASE system may be further elaborated and applied not only in identifying and updating the knowledge content of the educational domain knowledge in question, but also in other domains (creating and updating corporative knowledge bases, improving business intelligence measures, implementing knowledge bases, evaluating the completeness of the content knowledge of accumulated in repositories). It may be objectives of further works.

References

- Ahmed, N.U., Sharma, S.K. (2006). Porter's value chain model for assessing the impact of the internet for environmental gains, *Management and Enterprise Development*, Vol. 3, No. 3, 278–295.
- Andreescu, A., Mircea, M. (2009). Managing Knowledge as Business Rules, *Informatica Economică*, Vol. 13, No. 4, 63–74.
- Bandarian, R. (2008). Exploiting value chain process concepts in research organisations, *International Journal of Value Chain Management*, Vol. 2, No. 3, 400–416.
- Beard, D., Schwieger, D., Surendran, K. (2010). A Value Chain Approach for Attracting, Educating, and Transitioning Students to the IT Profession, *Information Systems Education Journal*, Vol. 8, No. 7, 1–12.
- Caplinskaskas, A. (2002). Curricula engineering: application of systems engineering methods to the development of university curricula, *Information technology and control*, Vol. 22, No. 1, 53–58.
- Caplinskaskas, A., Vasilecas, O. (2002). Modern curricula in information systems: a case study, *Information technology and control*, Vol. 21, No. 1, 59–63.
- Chen, Y-L., Yang, T.-C., Lin, Z-S. (2004). A study on the modelling of Knowledge value chain, *Knowledge Management*, 1–12.
- Christensen, L.C., Johansen, B.W., Midjo, N., Onarheim, J., Syvertsen, T.G., Totland, T. (1995). *Enterprise modelling – practices and perspectives*, available at <http://www.idi.ntnu.no/grupper/su/publ/pdf/asme95.pdf>.
- Cummins, F. (2010). Value Chain Modelling: Linking Customer Value to Business Process Design and 2010 Automation, *Cutter IT Journal*, Vol. 23, No. 2, 23–30.
- Denisovas, V., Gudas, S., Tekutov, J. (2010). Study Program Requirements Engineering Method and Information System, *Information Sciences*, Vol. 53, 106–126.
- Giaglis, G.M. (2001). A Taxonomy of Business Process Modelling and Information Systems Modelling Techniques, *International Journal of Flexible Manufacturing Systems*, Vol. 13, No. 2, 209–228.
- Gudas, S., Lopata, A. (2001). The enterprise model based on identification of information resources, *Information Sciences*, Vol. 19, 43–50.
- Gudas, S., Brundzaitė, R. (2005). Enterprise knowledge modelling based on modified value chain model, *Information Sciences*, Vol. 35, 179–192.
- Gudas, S., Lopata, A., Skersys, T. (2008). Approach to enterprise modelling for information systems engineering, *Informatics*, Vol. 16, No. 2, 175–192.
- Gudas, S. (2009). Enterprise knowledge modelling: domains and aspects, *Technological and Economic Development of Economy*, Vol. 15, No. 2, 281–293.
- Gudas, S. (2012). *Knowledge-Based Enterprise Framework: A Management Control View*, *New Research on Knowledge Management Models and Method*, available at <http://www.intechopen.com/books/new-research-on-knowledge-management-models-and-methods/knowledge-based-enterprise-framework-a-management-control-view>.
- Gudas, S. (2012). *Information systems engineering theory*, Vilnius: Vilnius University Press, 25–65.
- Hagerty, J., Sallam, R.L., Richardson, J. (2012). *Magic Quadrant for Business Intelligence Platforms*, available at <http://www.gartner.com/technology/reprints.do?id=1-1982NPD&ct=120208&st=sb>.
- Gui, S., Ye, Z. (2008). The Research of Value Chain Operation Model in E-Commerce, *International Seminar on Business and Information Management*, 238–241.
- Henderson, J., Venkatraman, N. (1990). *Strategic alignment: A model for organization transformation via information technology*, 3223–3290.
- Henschen, D. (2011). *BI and Information Management Trends*, available at http://www.umsl.edu/~sauterv/DSS/research-2012-bi-and-information-management_9951311.pdf.

- Koc, T., Bozdog, E. (2017). Measuring the degree of novelty of innovation based on Porter's value chain approach, *European Journal of Operational Research*, Vol. 257, No. 2, 559-567.
- Kim, K.S., Wang, T.C., Yang, X.B. (2005). Simulation of apparent infection rate to predict severity of soybean rust using a fuzzy logic system, *Phytopathology*, Vol. 95, No. 10, 1122-1131.
- Kurilovas, E., Serikoviene, S., Vuorikari, R. (2014). Expert Centred vs Learner Centred Approach for Evaluating Quality and Reusability of Learning Objects, *Computers in Human Behavior*, Vol. 30, 526-534.
- Lee, M.C., Han, M.W. (2009). Knowledge Value Chain Model Implemented for Supply Chain Management Performance, *Fifth International Joint Conference on INC, IMS and IDC*, 606-611.
- Landry, R., Amara, N., Pablos-Mendes, A., Shademani, R., Gold, I. (2006). The knowledge-value chain: a conceptual framework for knowledge translation in health, *Bulletin of the World Health Organization*, Vol. 84, No. 8, 597-602.
- Mertins, K., Jochem, R. (2005). Architectures, methods and tools for enterprise engineering, *International Journal of Production Economics*, Vol. 98, No. 2, 179-188.
- Millette, C.A. (2005). Status of department of defense architecture framework (DoDAF) implementation within the aeronautical systems center (ASC), *Air force institute of technology*, available at <http://www.stormingmedia.us/11/1134/A113434.html>.
- Morkevičius, A., Gudas, S. (2011). Enterprise Knowledge Based Software Requirements Elicitation, *Information technology and Control*, Vol. 40, No. 3, 181-190.
- Najmaei, A., Sadeghinejad, Z. (2009). Competitive Strategic Alliances Through Knowledge Value Chain, *International Review of Business Research Papers*, Vol. 5, No. 3, 297-310.
- OMG consortium (2009). Business modelling specifications, available at <http://www.omg.org/spec/>.
- Ounaies, H.Z., Jamoussi, Y., Ben Ghezala, H.H. (2009). Evaluation framework based on fuzzy measured method in adaptive learning system, *Themes in Science and Technology Education*, Vol. 1, No. 1, 49-58.
- Pathak, V., Pathak, K. (2010). Reconfiguring the higher education value chain, *Management in Education*, Vol. 24, No. 4, 166-171.
- Porter, M.E. (1998). *Competitive Advantage: Creating and Sustaining Superior Performance*.
- Roko, L. P., Opusunju, M. I. (2016). Value Chain and Performance in Agro Allied Small and Medium Scale Enterprise, *International Journal of Business and Social Research*, Vol. 6, No. 9, 8-19.
- Saha, A. (2012). *Mapping of Porter's value chain activities into business functional units*, available at <http://www.managementexchange.com/hack/mapping-porter%E2%80%99s-value-chain-activities-business-functional-units>.
- Scheer, A.-W., Schneider, K. (2006). ARIS – Architecture of Integrated Information Systems, *Handbook on Architectures of Information Systems*, Part Three, 605-623.
- Stewart, G. (1997). Supply-chain operations reference model (SCOR): the first cross-industry framework for integrated supply-chain management, *Logistics Information Management*, Vol. 10, No. 2, 62-67.
- Sweeney, E. (2009). Supply Chain Management and the Value Chain. Supply Chain Perspectives, *Journal of the National Institute for Transport and Logistics*, Vol. 10, No. 2, 13-15.
- Tekutov, J., Gudas, S., Denisovas, V., Tekutova, J. (2012). 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*, 108-120.
- Tekutov, J., Gudas, S., Denisovas, V. (2012). The Refinement of Study Program Content Based on a Problem Domain Model, *Transformations in Business & Economics*, Vol. 11, No. 1, 199-212.
- Tekutov, J. (2013). *Model of Domain Knowledge Content Updating Based on Management Information Interactions*, PhD dissertation. Vilnius: Vilnius University.

- Trask, B., Roman, A. (2009). Introduction to Model Driven Development with Examples using Eclipse Frameworks, *ACM Conference on Object-Oriented Programming, Systems, Languages and Applications*, Orlando.
- Williamson, E. A., Harrison, D. K., Jordan, M. (2004). Information Systems Development within Supply Chain Management, *International Journal of Information Management*, Vol. 24, No. 4, 375–385.
- Wong, H.K. (2004). Knowledge Value Chain: Implementation of New Product Development System in a Winery, *The Electronic Journal of Knowledge Management*, Vol. 2, No. 1, 109–122.
- Zadeh, L. (1965). Fuzzy sets, *Information and Control*, Vol. 8, No. 3, 338–353.
- Zhang, L.L., Cheng, D.Y. (1992). Extent analysis and synthetic decision, *Support Systems for Decision and Negotiation Processes*, Vol. 2, 633–640.

Received October 9, 2018, revised February 10, 2019, accepted March 12, 2019