ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol.36, No.3

META-MODEL BASED DEVELOPMENT OF USE CASE MODEL FOR BUSINESS FUNCTION

Saulius Gudas, Audrius Lopata

Kaunas University of Technology, Information Systems Department
Studentu St. 50, LT–51368 Kaunas, Lithuania

Vilnius University, Kaunas Faculty of Humanities
Muitines St. 8, Kaunas, Lithuania

Abstract. The principles and major steps of Enterprise Meta-Model (EMM) based development of Use Case model (UCM) in CASE system environment are presented in this paper. The Enterprise Meta-Model represents the key concepts of domain knowledge. The enterprise processes, management functions, and their interactions are considered as the critical components of the domain knowledge accumulated as Enterprise model in the knowledge base of CASE system. The formal background for generation of UCM is mapping rules of EMM constructs to constructs of UCM meta-model. The key rules and steps of Meta-Model based development of UCM for user specified business function are presented and illustrated.

Keywords: Information Systems engineering, CASE system, knowledge-based, Enterprise Meta-Model, domain knowledge, Use Case Meta-Model.

1. Introduction

The critical feature of IS engineering methods is their empirical nature, because the project models in the CASE system are composed on the basis of knowledge which is not verified against formalized criteria. The problem domain knowledge acquisition process relies heavily on the analyst and user. It is not clear whether the knowledge about this problem domain is adequate. The human plays the pivotal role in problem domain knowledge acquisition process, and few formalized methods of knowledge acquisition control are taken into consideration.

There is a great number of Enterprise modelling methods and approaches [9], [12], [13] (such as CIMOSA [2], GERAM [4], IDEF suite, GRAI, DoD [3], MDA [10]), standards (ISO 14258, ISO 15704, PSL, ISO TR 10314, CEN EN 12204 [1], CEN 40003 [2]) and supporting Enterprise modelling tools [11].

Typical disadvantage of CASE methods should also be mentioned: design stage models are constructed in an interactive mode, and only several IS design stage models are partly generated. The reason is an insufficient enterprise model composition. Currently, in the first stage of IS development cycle, CASE systems generate a diagram of functional hierarchy according to problem domain model (Data Flow Diagram or Work Flow Model), while in the last stage of IS development cycle, code (prototype of user interface) is generated according to class model and data base specification. Other system project models are constructed interactively.

Therefore, some gaps of IS development process occur due to the human factor. These gaps mean that the project model is constructed in an interactive way (when the human participates), but not in an algorithmic one. This determines the incompatibility of IS project models and the incoherence of IS development process. Some IS development gaps can be avoided when applying formalized (algorithmic) methods of domain knowledge analysis using meta modeling.

2. The principles of knowledge-based IS engineering

Systems analysis of trends of IS engineering methods towards the knowledge–based engineering shows the cause of feasible changes in architecture of CASE tools [5]. The principles of knowledge-based IS engineering (KB ISE) were stated by analysis of trends of IS engineering [5]. These principles of KB ISE refine the Enterprise model, Enterprise Meta-Model and formal Enterprise model (i.e. some formally defined Enterprise Framework) as the obligatory concepts of any knowledge-based CASE method

and obligatory components of knowledge-based CASE tool.

The knowledge-based CASE process is defined and constructed on the basis of domain knowledge acquired by the Enterprise Meta-Model as an obligatory layer of the Knowledge Base – a part of the Repository of knowledge-based CASE tool [6]

The underlying functionality of knowledge-based CASE tool is verification of IS project (i.e. a set of IS models) against the Knowledge Base of CASE tool.

The user is considered as an intermediate between the Real World (Enterprise) and IS developer. The user knowledge about Enterprise is limited (related with the role of user in enterprise activities), and consequently – user requirements are inconsistent.

The consistent pattern of the Real World Enterprise activities is conceptualized and formally defined as specification including Enterprise Meta-Model and Enterprise model. The development of Enterprise Meta-Model is a fairly complicated problem, related with developments in the areas of enterprise modelling languages, the concepts of control theory [8] and management control [7].

Figure 1 depicts the architecture of the CASE system enhanced by the Knowledge Base. The Knowledge Base of the CASE system consists of two parts: an Enterprise Meta-Model (EMM) and Enterprise model (EM). An Enterprise Meta-Model is the generic level model; an Enterprise model includes the partial and particular level models in accordance with GERAM [4].

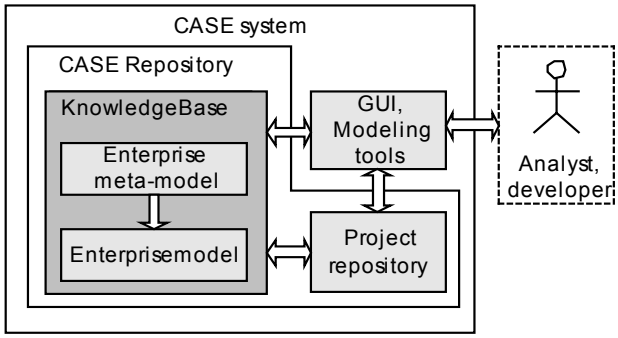

Figure 1. The architecture of CASE system with Knowledge Base

IS development problems occur when empirically acquired information (user requirements) has to be verified and validated.

The Knowledge Base of the CASE system is supposed to be the third active source of Enterprise knowledge (next to Analyst and User) for information systems engineering. Enterprise Meta-Model (EMM) in this enhanced environment of information system development is a source of pre-defined knowledge, and is used to control the process of business domain knowledge acquisition and analysis. It is also used to control the construction of an Enterprise model (EM) for a particular business domain.

Knowledge-based IS development supposes that all stages of IS development life cycle are supported by the CASE system's Knowledge Base. Enterprise model (EM) is used as an alternative source of knowledge (next to IS developer knowledge) during the IS development process.

The Knowledge Base of the CASE system (in conjunction with appropriate algorithms) assures the consistency among the business analysis and IS design models, gives new possibilities for verification and validation of IS development deliverables at the life cycle steps.

Problem domain knowledge (which is examined through formalized criteria) should be stored in the enterprise knowledge repository of CASE tool and should be used to control knowledge of user and analyst also to verify IS project solutions. This repository of CASE tool is used for the generation of IS engineering design stage models too.

The components and composition of Enterprise Model are regulated by formalized specification, which is called Enterprise Meta-Model.

3. Formalized Description of Enterprise Meta-Model

The particularity of this Enterprise Meta-Model is the way of modelling of the interaction of Process and Function. A Process is concerned as a partially ordered set of steps, which can be executed to achieve the desired material end–result. Process consumes material resources and produces some material output, i.e. a product. Processes are triggered by one or more Event occurrences.

Function is concerned as a workflow element, which controls Process. A Function is a complex construct. The structure of the Function is defined on the basis of the formal definition of management function [7]. The composition of key concepts of the Enterprise Meta-Model (EMM) is presented in Figure 2. At least one Function controls each Process, transforming material input flow into material output flow. Function accomplishes at least one organizational Goal or its sub-goal. Process and Function are performed by an enterprise Actor. Not only a human or organizational

unit, but also software or device can perform Function or Process. Material processing is stimulated by an environmentally initiated Event. Environment initiates Event and influences enterprise Goals.

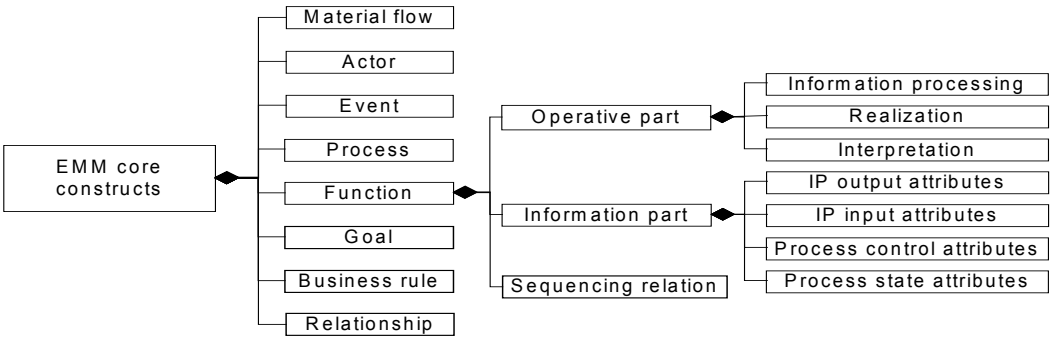

Figure 2. The key concepts of the Enterprise Meta-Model (EMM)

Formalized descriptions of EMM and UCM meta-model and their interplay are the background used to describe UCM generation algorithm.

Formalized Enterprise Meta-Model **M** is described on the basis of abstract algebra:

$$.\mathbf{M=<K, R>,}$$

K–set of classes of EM, R–set of relations. K={K1, K2,…, K21}, R={r1, r2, r3} [7].

Thus **Enterprise Meta-Model** consists of classes as follows: M=<{K1, K2,...,K21}, {r1, r2, r3}>, where K1– class Process, K2– class Function, K3– class Actor, K4– class Event, K5– class Goal, K6– class Material Flow, K7– class Input Material Flow, K8– class Output Material Flow, K9– class Information Flow, K10– class Interpretation, K11– class Information Processing and Decision Making (IP), K12– class Realization, K13– class Information Activity, K14– class Business Rule (BR), K15– class Interpretation Business Rules, K16– class Information Processing and Decision Making Business Rule (IP BR), K17– class Realization Business Rules, K18– class Process Output, K19– class IP Input , K20– class IP Output, K21– class Process Input.

The set of relations in the EMM is as follows: r1– Aggregation relation, r2– Generalization relation, r3– Association. The conceptual scheme of Enterprise Meta-Model is shown in Figure 3.

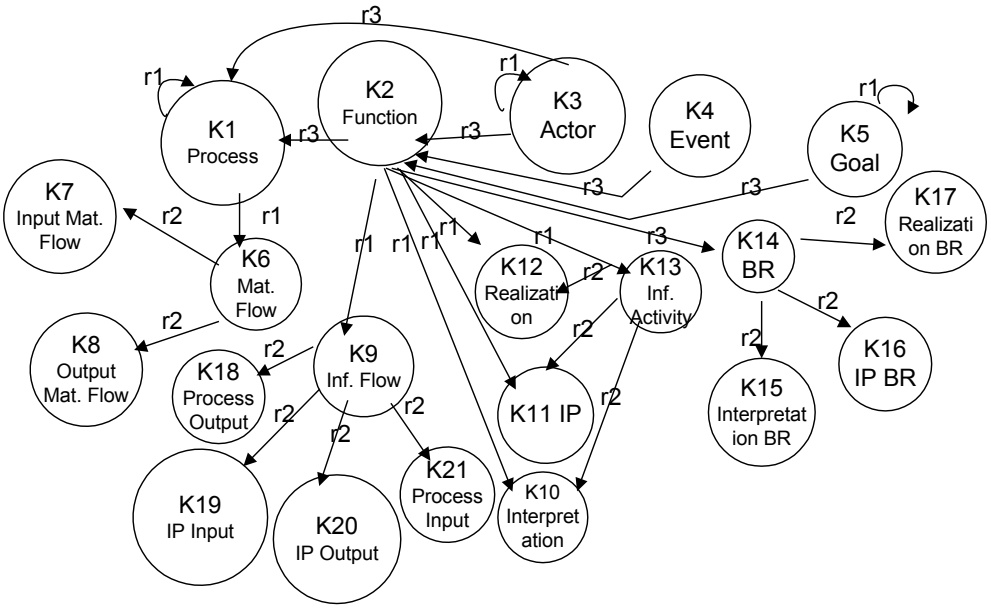

Figure 3. Conceptual scheme of Enterprise Meta-Model **M**

4. Formalized Description of Use Case Model

There are different techniques of user requirements acquisition and specification: Use Case model of UML, user requirements specification templates (Volere) and other. The Use Case model (UCM) is a popular UML model aimed at user requirements specification. Use Case models are used to specify the relationships (transactions) between the Actors that use the system and the Use Cases they use.

The Use Case meta–model comprises the following constructs: Actor, Use Case and Relation (there

are three types of relations: Association, Include and Extends). An Actor defines a coherent set of roles that users of an entity can play while interacting with the entity. The Use Case construct is used to define the behaviour of a system or other semantic entity without revealing the entity's internal structure. Each Use Case specifies a sequence of actions, including variants that the entity can perform, while interacting with actors of the entity. The construct Association can be refined as an Information Flow between the constructs Actor and Use Case. Association states that an instance of the Use Case and a user playing one of the roles of the Actor communicate. An Include relationship defines that a Use Case contains the behaviour defined in another Use case. An Extend relationship defines that instances of a Use Case may be augmented with some additional behaviour defined in an extending Use Case. Detailed composition of the UCM is specified in the UML specification version 1.5. The UCM meta-model, presented in this specification version, is depicted in Figure 4.

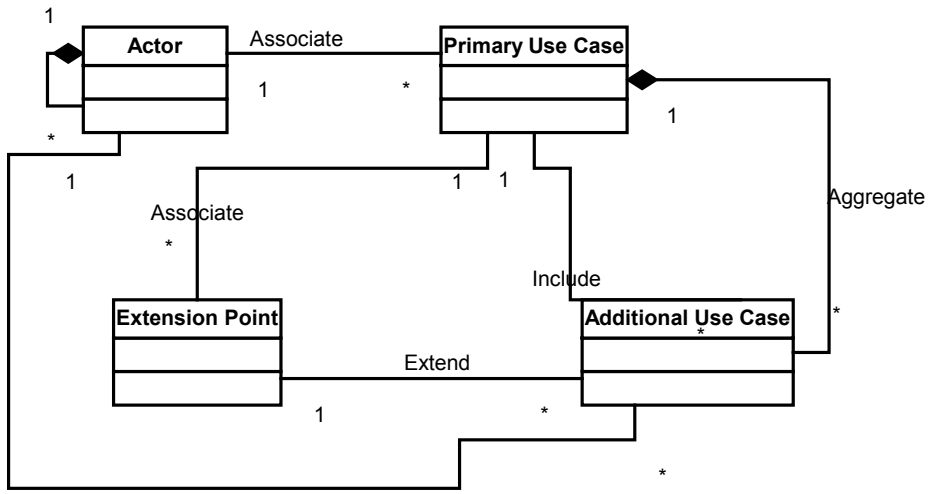

Figure 4. UCM meta-model

Traditionally, UCM is aimed to specify user requirements for particular task (i.e. some information processing activity).

In this approach the user identified elements of the particular Enterprise model (for instance, user required identifiers of business function or business process, etc.) are used to single out related subset of knowledge (i.e. subset of coherent EM elements) in the particular EM and depict them according to UCM notation rules.

5. Principles of Enterprise Meta-Model Based UCM Generation

Any element of particular EM can be identified by user as requirement and, consequently, used as a starting point for user requirements analysis resulting as Use Case diagram. Notionally, the particular UCM can be generated throughout analysis of the Enterprise Meta-Model in several different ways – starting with any required type of Enterprise Meta-Model components (Figure 3, classes Process, Function, Actor, Goal etc.).

For instance, in case of UCM generation for required element Process* of particular EM, analysis of the Enterprise Meta-Model is conducted starting with class Process of the *EMM* (Figure 3). Consequently, a list of related EM elements of definite type (Material flow, Function, Actor) are selected and specified as UCM. This type of UCM is called *UCM of Processes*.

The *UCM of function* specifies the composition of function (its components) and actors. The components of Use Case model of Function are Function, Information Activities, Information Flows and Actors.

The types of UCM generated for the class Actor, specify actor's material processes, functions and informational activities. UCM generated for class Goal specifies Functions, related to organizational goals, and their components (informational activities).

The basic variants of Enterprise meta-model based UCM generation are shown in Table 1. The other possible ways of UCM generation for such EMM classes as Material flow, Information flow, Information Activity and Business Rule are not included, but could be discussed.

A list of five different techniques (variants) of UCM generation which are based on the different constructs of Enterprise Meta-Model, used as a starting point for user requirements analysis and Use Case diagram generation (Fig. 5) is as follows:

a) *The UCM of Process* is generated starting with the definite ID of the particular Process specified in the user requirements. If this definite Process is the object of the particular Enterprise model, the coherent UCM (comprising Enterprise model (EM) elements related with this Process) can be generated;
b) *The UCM of Actor* is generated starting with the definite ID of the particular Actor specified in the

user requirements. If this definite Actor is the object of the particular Enterprise model, the coherent UCM (comprising Enterprise model (EM) elements related with this Actor) can be generated;

c) *The UCM of Function* is generated starting with the definite ID of the particular Function specified in the user requirements. If this definite Function is the object of the particular Enterprise model, the coherent UCM (comprising Enterprise model (EM) elements related with this Function) can be generated;

d) *The UCM of Goal* is generated starting with the definite ID of the particular Goal specified in the user requirements. If this definite Goal is the object of the particular Enterprise model, the coherent UCM (comprising Enterprise model (EM) elements related with this Goal) can be generated;

e) *The UCM of Information Activity* is generated starting with the definite ID of the particular Information Activity specified in the user requirements. If this definite Information Activity is the object of the particular Enterprise model, the coherent UCM (comprising Enterprise model (EM) elements related with this Information Activity) can be generated.

Table 1. Variants of Enterprise meta-model based UCM generation

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

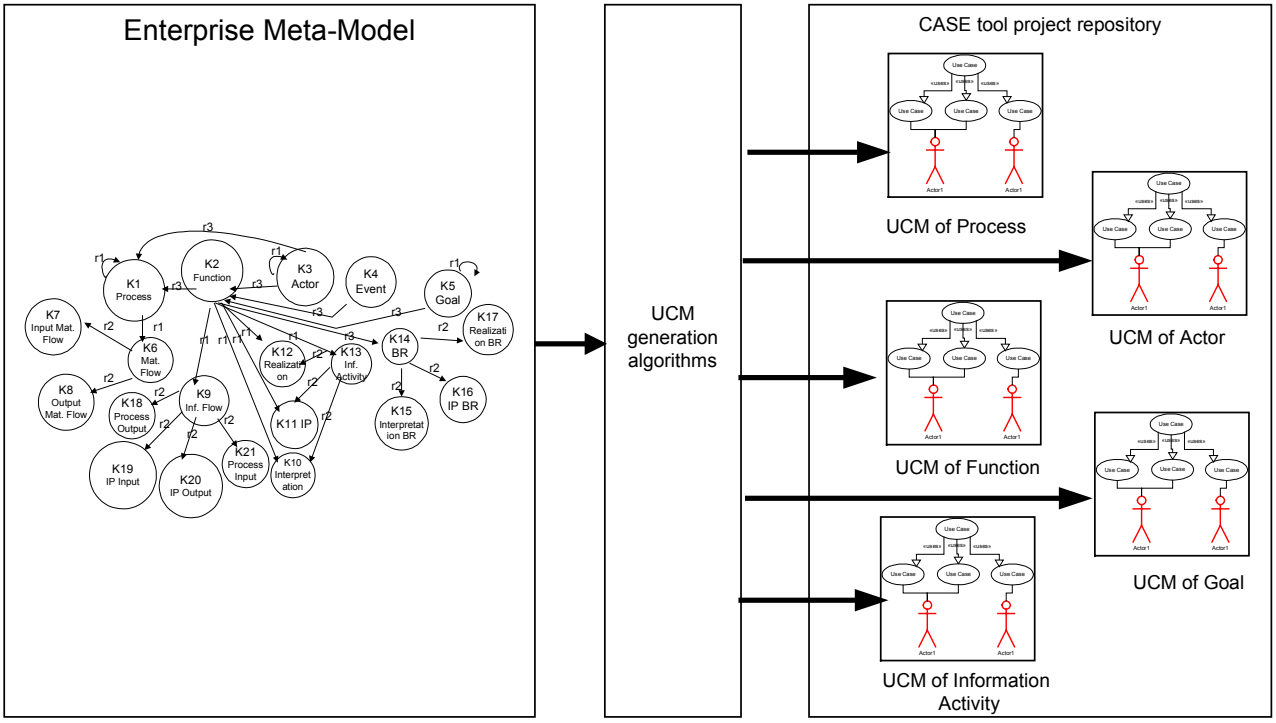

Figure 5. Different techniques of Enterprise Meta-Model based UCM generation

6. Generation of UCM for Particular Business Function

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

Formally UCM of function is described as system F2: F2=<{K22, K23,K24,K25}, {r1, r3} F2– stands for UCM of *Function*, K22– class *Actor* , K23– class *PA(Function)*, K24– class *Extension point*, K25– class *PA (Activity)*, r1– *Aggregation* relation, r3– *Association*. Relation *Generalization* is not used. The

relations among components of F2 are described as follows:

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

Graphical scheme of meta-model of UCM F2 is presented in Figure 6.

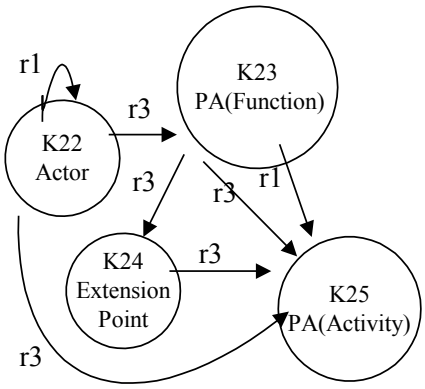

Figure 6. Meta-model of UCM of function

7. Relation Between Enterprise Meta-Model and UCM Meta-Model of Function

Logical relation between enterprise model and UCM of function is determined on the basis of their class models analysis. Such enterprise model classes as *Function*, *Actor*, *Informational activity* and *Business rules* are necessary to generate UCM model of function. During the generation of UCM model of function, it is reflected into classes *PA (Function)*, *Actor*, *PA (Activity)* and *Extension point*. This can be formally defined by reflections of sets φ_1: K2→K23; φ_2:K3→K22; φ_3:K13→K25; φ_4:K14→K24 (Table 2). The names of sets correspond to class names.

Table 2. Mapping of the Enterprise Meta-Model **M** elements to elements of UCM meta-model **F2**

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

The relationships of Enterprise Meta-Model **M** and UCM Meta-Model **F2** are shown in Figure 7. Scheme of generation of UCM for business function is presented in Figure 8.

Some UCM for particular business function F2* were developed (generated) according to these rules. Examples of UCM generated for business function are presented in Figure 9.

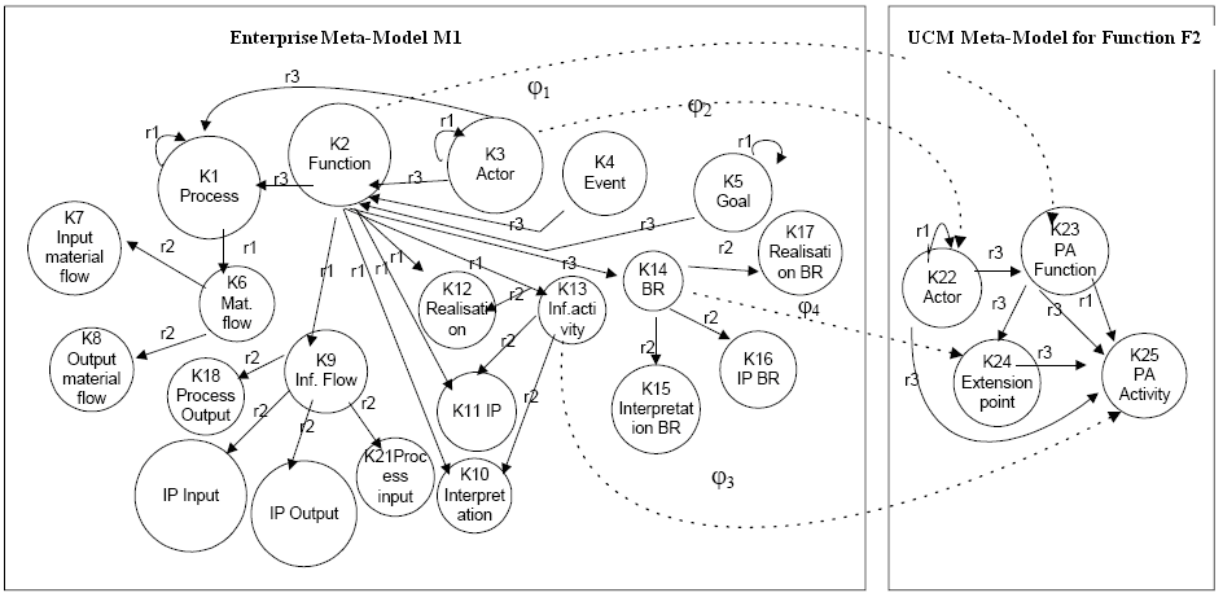

Figure 7. The mapping of Enterprise Meta-Model M1 elements to UCM Meta-Model F2

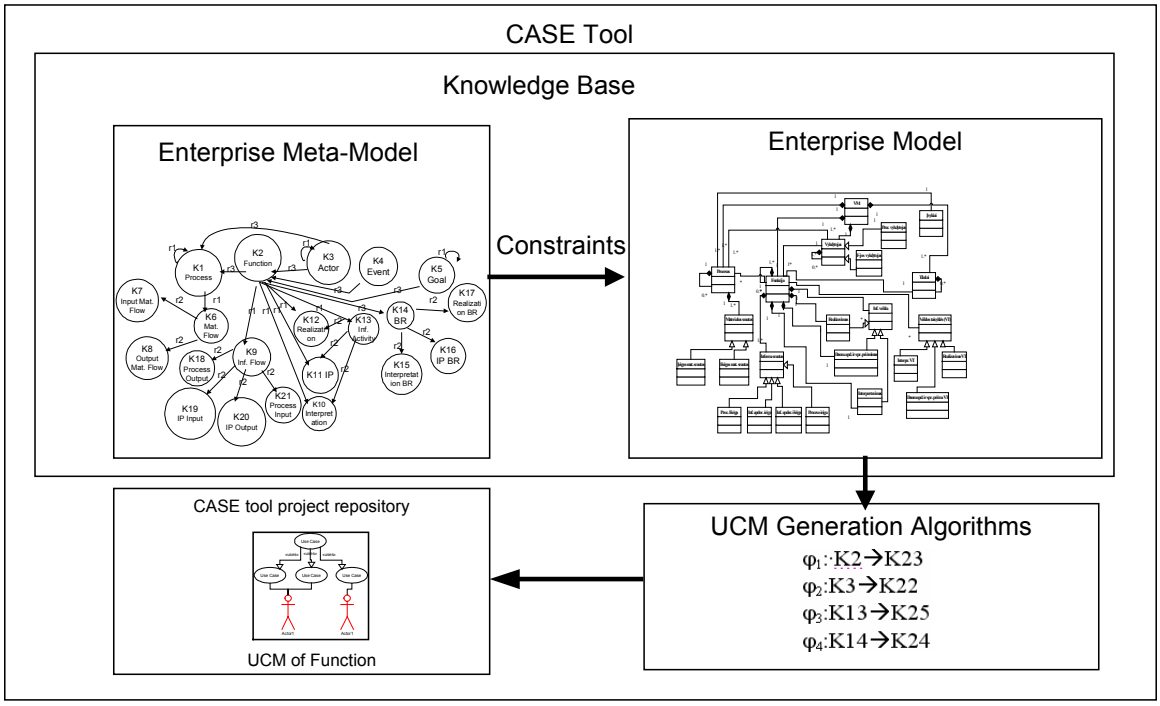

Figure 8. Scheme of generation of UCM for business function

8. Conclusions

This method ensures the user functional requirements specification, which is verified against domain knowledge, acquired on the basis of formalized Enterprise Meta-Model and Enterprise Model.

The user requirements specifications generation and control is based on the Enterprise Meta-Model and Enterprise Model stored in the enhanced repository of CASE system.

The enterprise processes, management functions, and their interactions are considered as the critical components of the domain knowledge accumulated as Enterprise model in the knowledge base of CASE system.

The Use Case model generation principles are based on the mapping of Enterprise Meta-Model and UCM meta-model. The Enterprise Meta-Model is used as the source of domain knowledge for requirements specification.

User requirements specification algorithms were created to generate Use Case models starting with particular construct of enterprise model: function, process, actor, goal and activity. Consequently, different techniques (variants) of the UCM generation which are based on the different constructs of Enterprise

Meta-Model (used as a starting point for user requirements analysis and Use Case diagram generation) are identified.

This meta-model based approach to functional requirements specification could be used to enhance the functionality of CASE tools.

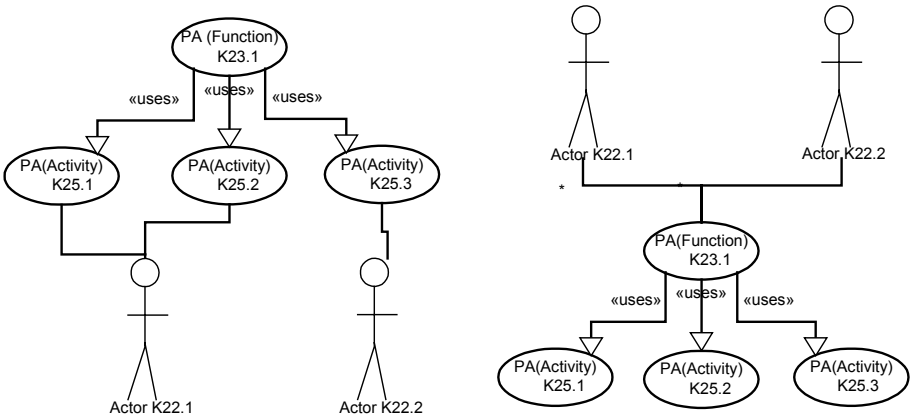

Figure 9. Examples of UCM generated for business function

References

[1] ENV 12 204. Advanced Manufacturing Technology Systems Architecture – Constructs for Enterprise Modelling. *CEN TC* 310/*WG*1, 1996.

[2] ENV 40 003. Computer Integrated Manufacturing Systems Architecture – Framework for Enterprise Modelling. *CEN/CENELEC*, 1990.

[3] Department of Defence Technical Architecture Framework for Information Management. Version 3.0. *Defence Information Systems Agency, Center of Standards*, 1996.

[4] GERAM: Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3. *IFIP–IFAC Task Force on Architectures for Enterprise Integration*, 1999.

[5] **S. Gudas.** Organisational System as a Hierarchy of Information Processes. *Applications of Artificial Intelligence in Engineering* VI (*AIENG* 91). *Computational Mechanics Publications, Southampton Boston*, 1991, 1037 –1050.

[6] **S. Gudas, T. Skersys, A. Lopata.** Framework for Knowledge-based IS Engineering. *Advances in Information Systems ADVIS*'2004, *T. Yakhno, ed., Springer-Verlag, Berlin, LNCS* 3261, 2004, ISSN 0302-9743, 512-522.

[7] **S. Gudas, A. Lopata, T. Skersys.** Approach to Enterprise Modelling for Information Systems Engineering. *INFORMATICA, Vol*.16, *No*.2, *Institute of Mathematics and Informatics, Vilnius*, 2005, 175-192.

[8] **M.M. Gupta, N.K. Sinha.** Intelligent Control Systems: Theory and Applications. *The Institute of Electrical and Electronic Engineers Inc., New York*, 1996.

[9] **J. Schekkerman.** How to Survive in the Jungle of Enterprise Architecture Frameworks. *Trafford, Canada*, 2003.

[10] **J. Stephen, S. Kendall, A. Uhl, D. Weise.** MDA Distilled: Principles of Model-driven Architecture. *Addison-Wesley Pub. Co*., 2004.

[11] **T. Totland.** Enterprise Modelling as a Means to Support Human Sense-Making and Communication in Organizations. *Norwegian University of Science and Technology (NTNU), Trondheim IDI-raport* 1997:8, 1997.

[12] **E. Turban, E. McLean, J. Wetherbe.** Information Technology for Strategic Advantage, 2nd Edition. *John Wiley & Sons, London*, 1999.

[13] **F. Vernadat.** UEML: Towards a Unified Enterprise modelling language. *International Conference on Industrial Systems Design, Analysis and Management* (*MOSIM*'01), *Troyes, France*, 2001.

Received June 2007.