ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2007, Vol.36, No.1A

WORKFLOW MODELS BASED ACQUISITION OF ENTERPRISE KNOWLEDGE

Saulius Gudas, Audrius Lopata

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

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

Abstract. The approach for Enterprise modelling extended by the management point of view is presented. The enterprise processes, management functions, and their interactions are considered as the critical components of the domain knowledge accumulated for the IS engineering purposes. The workflow models analysis based acquisition of knowledge aimed to accumulation of definite Enterprise model is developed and presented in this paper. The architecture of the advanced CASE systems is described as well.

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

1. Introduction

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

Other typical characteristics (disadvantage) of present–day CASE methods should also be mentioned: design stage models are made in an interactive mode (both the designer and CASE tool participate), and only several IS design stage models are partly generated because of an insufficient enterprise model composition. Currently, in the first stage of IS designing cycle, CASE systems generate a diagram of functional hierarchy according to problem domain model (Data Flow Diagram or Work Flow Model), while in the last stage of IS designing cycle, program code (prototype of user interface) is generated according to class model and data base specification. Other project models are formed interactively, i.e. designer, analyst and programmer create IS project models through analyzing models designed in earlier stages.

Therefore, gaps of IS engineering process occur due to the human factor. These gaps mean, that the project model is formed in an interactive way (when the human participates), but not in an algorithmic one. This determines the incompatibility of IS project models and the incoherence of IS designing process, because in IS engineering process human is overloaded. Many mistakes can be avoided when applying formalized (algorithmic) methods of knowledge analysis, control and generating. 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]

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 this 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 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 (the destination) is verification of IS project (i.e. a set of IS models) against the CASE tool Knowledge Base (Figure 1).

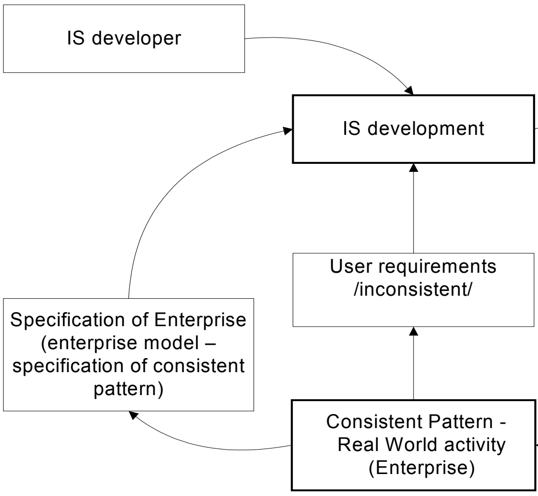

Figure 1. Knowledge–based IS engineering

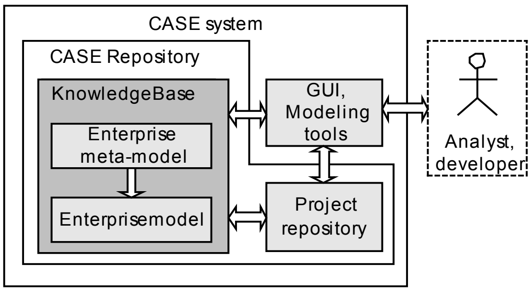

Figure 2. The architecture of CASE system with Knowledge Base

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 of Enterprise (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, the concepts of control theory [8] and management control [7].

Figure 2 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].

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 IS analysis and design models, gives new possibilities for verification and validation of IS development life cycle steps.

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

Function is a workflow element, which controls processes. A Function is a complex construct. The structure of the Function is defined on the basis of the formal definition of management function [7]. The composition of the Enterprise Metamodel (EMM) is presented in Figure 3. At least one Function controls each Process, transforming material input flow into material output flow. Function accomplishes at least one organizational Goal or its subgoal. Process and Function are performed by an enterprise Actor. Not only a human or organizational unit, but also software or device can perform Function or Process. Material processing is stimulated by an environmentally initiated Event. Environment initiates Event and influences enterprise Goals.

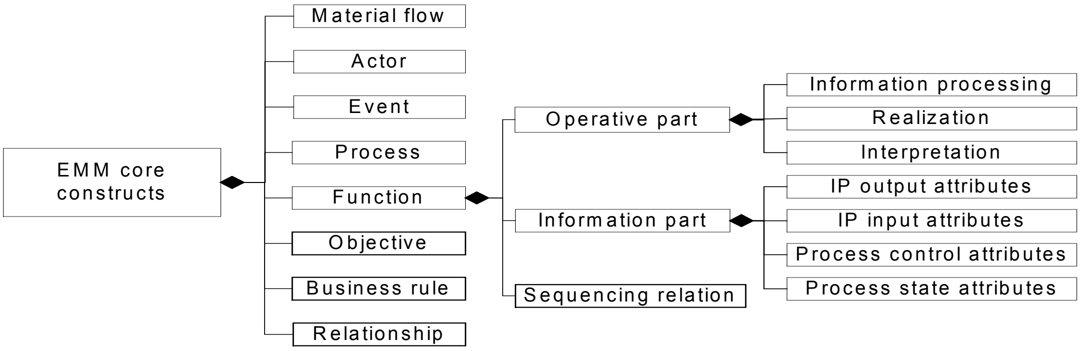

Figure 3. Composition of the Enterprise Metamodel (EMM)

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 composition of enterprise model is regulated by formalized method based specification, which is called enterprise metamodel. IS a development problem occurring when empirically acquired information (requirements) has to be verified and validated.

3. Work Flow models based User requirements acquisition process

This section deals with the major principles of a knowledge–based approach to development of Knowledge Repository of the CASE system. The workflow model based domain knowledge acquisition and analysis is given in Figure 4.

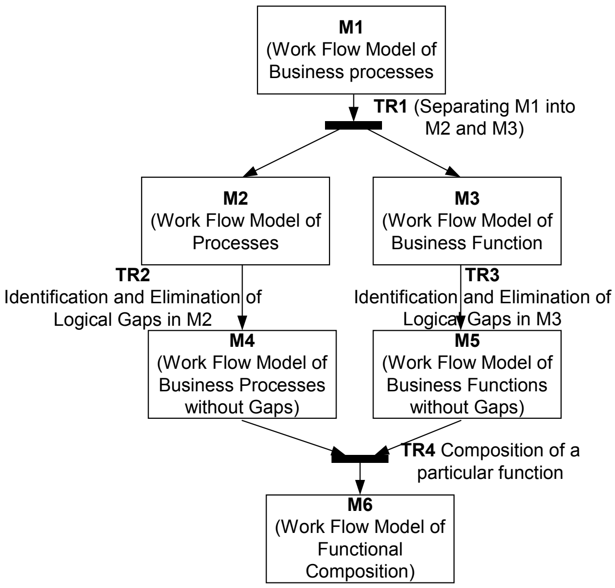

Figure 4. The workflow model based domain knowledge acquisition technique

The process of enterprise knowledge acquisition is based on the modified workflow models. In the stage of problem domain knowledge acquisition six types of modified work flow models are created:

- M1 – Work Flow Model of Business Processes;
- M2 – Work Flow Model of Processes;
- M3 – Work Flow Model of Functions;

- M4 – Work Flow Model of Processes without Gaps,
- M5 – Work Flow Model of Functions without Gaps,

M6 – Work Flow Model of Functional Composition;

In order to create such models and transform knowledge into the enterprise model, algorithms of four types are developed:

- TR1 – separating M1 into M2 and M3;
- TR2 – identifies and eliminates logical gaps in M2;
- TR3 – identifies and eliminates logical gaps in M3;
- TR4 – determines the composition of a particular function according to the internal structure of Enterprise Metamodel.

Transformation TR1. Separation M1 into M2 and M3

Problem domain knowledge, acquired in M1, is transformed into M2 and M3 when separation algorithm is performed. Yet, in the transformation process logical gaps may occur. A logical gap is a semantic discontinuity between the elements of the problem domain model (for instance, workflow model).

Transformations TR2 and TR3. Identification and elimination of logical gaps in M2 and M3

It is likely that throughout separating M1 into M3 and M2 logical gaps may be identified in newly created M3 and M2. A logical gap is a semantic discontinuity among the elements of the workflow model. The logical gaps appear when problem domain knowledge is acquired incompletely. On purpose to eliminate gaps of M2, detecting and eliminating algorithm is applied. Without reference to elimination method, M2 is complemented by non–existing, but wrongly or hardly specified knowledge (process, material flow and actor). Logical gaps of M2 are identified during the analysis of input and output flows of each material process. A logical gap in the M2 and M3 is identified if some Process or Activity is not related to input or output flow. Except the first and the last processes of the workflow model, each Process of the M2 must be related to at least one input material flow and one output material flow, in the same as each Activity of M3 must be related to at least one input information flow and one output information flow. On purpose to eliminate logical gaps of M2, the prototype of informational system, eliminating M2 gaps, is used: it was created by MS "VISIO 2000" CASE tool and MS "ACCESS 2000" data base management system. The algorithm for elimination of M3 logical gaps is analogous to that for elimination of M2 logical gaps. The main difference is that all actions are performed with M3 activities and informational flows, but not with M2 processes and material flows.

Logical gaps in the M2 and M3 models are identified by the algorithms of the M2 and M3 analysis and eliminated by the analyst. The application of these algorithms requires an additional analysis of the problem domain. Logical gaps can be eliminated in two ways:

- New elements of the M2 (Material Flow, Process, Actor) and M3 (Information flow, Activity, Actor) can be added by the Analyst as a result of additional analysis of the problem domain, performed by the User and Analyst;
- Some elements of the M2 (MaterialFlow, Process, Actor) and M3 (Information flow, Activity, Actor) can be excluded by the Analyst during the semantic analysis of the workflow models, performed by the User and Analyst.

The result of logical gaps elimination algorithms are M2 and M3 without logical gaps. In such eliminating process M1 is also updated with knowledge about lacking processes, activities, information or material flows of a particular problem domain. This process is called the first quality assuring cycle of computerized problem domain knowledge.

Transformation TR4. Determination of the composition of a particular function according to the internal structure of Enterprise Metamodel

The algorithm defining functional composition is performed at the next step of the stage of work flow model based computerized problem domain knowledge acquisition and analysis. During this process, the completeness of functional composition, which controls each process, is verified. (i.e. it is verified whether M3 functional elements – activities, controlling each process, are specified). The lacking activities are identified on the basis of enterprise meta-model composition. The process of functional composition algorithm performance indicates activities, which exist in the enterprise problem domain, but are not specified in M3. Information flows, which relate these activities, are also indicated in this process. Material processes, information activities, material and information flows (which are indicated during performance of functional composition algorithm) complement M1 by new elements. This process is called the second quality assuring cycle of problem domain knowledge acquisition process. The result of functional composition defining algorithm is M6. This model specifies the internal composition of particular material process controlling function, i.e. M3 model activities (which are attributed to Interpretation, Information Processing and Decision Making and Realization) and their relating information flows.

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

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

	Traditional WFM	M1	M2	M3	M4	M5	M6
Business Process	+ (not detailed)	+	–	–	–	–	–
Activity		–	–	+	–	+	+
Process		–	+	–	+	–	+
Material Flow	+ (not detailed)	+	+	–	+	–	+
Informational Flow		+	–	+	–	+	+
Actor	+	+	+	+	+	+	+
Activity type	–	–	–	–	–	–	+
Possibility of Logical Gaps	+	+	+	+	–	–	–

4. Work Flow Model of Functional Composition

The result of functional composition verification algorithm is Work Flow Model of Functional Composition (M6). Elements of M3 are specified in M6 as components of the functional composition, defined in enterprise metamodel. M6 specifies only one function, which controls one or more processes, specified in M2. In accordance with the internal structure of function, which is defined by enterprise metamodel, there are three types of M3 activities: Information activity of interpretation, information activity of processing and decision making (IP), Information activity of realization. Each M3 activity can correspond to one of the above mentioned component parts of functions. The algorithm, which defines functional composition, determines what part of function activities belong to and what material process do they control in M3. Each activity of M3, specified in M6, can be analogous component (Interpretation, IP or Realization) of several M6. The M6 metamodel is presented in Figure 5.

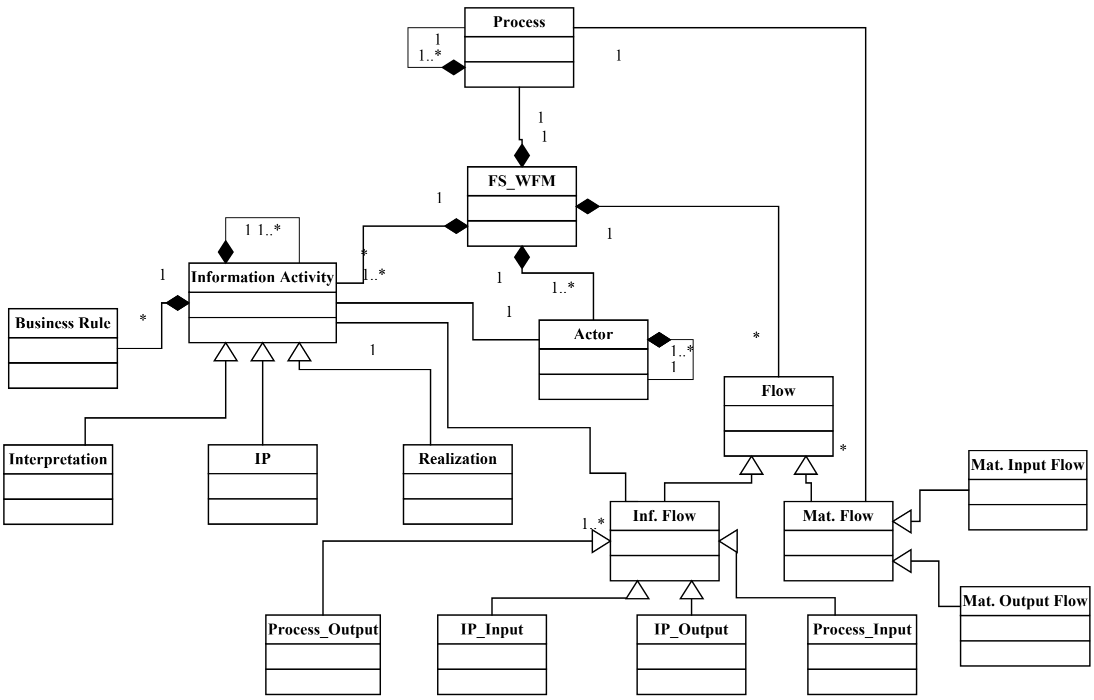

Figure 5. Metamodel of Work Flow Model of Functional Composition (M6)

Table 2 presents the M6 components of function, which are defined according to activities input and output flows existing in M3. According to the types of informational input and output flows, three types of the following activities can be distinguished: Interpretation, IP and Realization.

Table 2. Possible combinations of M6 activities input and output

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

If both input and output of M6 activity are information flow "Process Output", impossible type of activity is identified. Activities of M6, according to composition of enterprise metamodel, cannot have informational input and output flows of the same type. Activities, which have information input and output flows ("Process Output", "IP Input", "IP Output", "Process Input") of analogical type, can exist neither. If activity input is "Process Output" and output is "IP Input", the activity will be a component of function called Interpretation. Interpretation is a set of rules, intended to transform information flow "Process Output" into "IP Input", which is prepared for IP processing. Interpretation is a necessary component of function, because "Process Output" information flow can mismatch data format, determined for functional IP element input "IP Input". If activity input is "IP Input" and output is "IP Output", the activity is IP component of function. IP is a functional component, which is mainly intended to control process of information processing and decision making. If activity input is "IP Output" and output is "Process Input", the activity is part of function called Realization. Realization is a functional part, performing process, which is contrary to interpretation. Realization transforms "IP Output" data (processed in IP stage) into "Process Input" format (suitable to direct process control).

There are some cases when M3 activities define several component parts of function, according to activity input and output flows. If activity input is "Process Output" and output is "IP Output", the activity will have such functional components as IP and Interpretation as well as information flow "IP Input" (which links IP and Interpretation). If activity input is "Process Output" and output is "Process Input", activity will consist not only of Interpretation, IP and Realization but also "IP Input" (which links Interpretation and IP) and "IP Output" (which links IP and Realization). Such a composition indicates that this activity is function.

Activity input "IP Input" indicates two possible types of outputs: "IP Output" and "Process Input", while enterprise output "Process Input" indicates activities IP and Realization as well as information flow "IP Output" (which links IP and Realization). Activity input "Process Input" and output "Process Output" signal an error in M3, thus such a type of activity is impossible.

5. Conclusions

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

- Work Flow Model of Business Processes (M1), designed to specify problem domain processes, material and informational flows and actors;
- Work Flow Model of Material Processes (M2), designed to specify problem domain material processes, material flows and actors;
- Work Flow Model of Business Functions (M3), designed to specify problem domain activities, informational flows and actors;
- Work Flow Model of Functional Composition (M6) designed to specify the composition of business function.

The traditional composition of work flow model had to be modified in order to develop this method. Processes and enterprise functions were distinguished as things of qualitatively different nature in terms of control theory. Enterprise process models the material processes of enterprise, while the function models the informational one. The peculiarity of enterprise model based computerized specification method of user functional requirements is that computerized problem domain knowledge is verified according to the enterprise metamodel composition. The enterprise metamodel is used as source and criterion of enterprise knowledge necessary to IS Engineering. This determines the possibility of the user requirements specification quality control performed by CASE system.

The problem domain analysis is performed through the distinction of qualitatively different enterprise functions and enterprise processes. While analyzing the interaction of enterprise functions and processes from the informational aspect, the opportunity to verify the functional composition of the enterprise model according to formalized criteria occurs. Here two things are distinguished: material processes in computerized enterprise domain and component parts– activities of enterprise functions, which control these processes. The article presents the business process analysis method, which identifies and eliminates logical gaps. Such gaps occur when the user gives incomplete information about the material processes and material input and output flows in the computerized problem domain.

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

References

[1] ENV 12 204. *Advanced Manufacturing Technology Systems Architecture – Constructs for Enterprise Modelling*. CEN TC 310/WG1, 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, 2004, 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, 2*nd 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 March 2007.