*3$^e$ Conférence Francophone de MOdélisation et SIMulation «Conception, Analyse et Gestion des Systèmes Industriels»*
*MOSIM'01 – du 25 au 27 avril 2001 - Troyes (France)*

# UEML: TOWARDS A UNIFIED ENTERPRISE MODELLING LANGUAGE

**F. VERNADAT**

Vice-chairman of the IFAC-IFIP Task Force on Architectures for Enterprise Integration
LGIPM & MACSI-INRIA
ENIM/Université de Metz
Ile du Saulcy
F-57045 Metz cedex 1, France
Mél : vernadat@loria.fr or vernadat@enim.fr

**ABSTRACT :** *The paper presents the rationale and principles of a unified language devoted to the area of Enterprise Modelling. The language, named UEML, for Unified Enterprise Modelling Language, is not intended to replace existing languages but is intended to provide a uniform interface to enterprise modelling tools and a neutral format for exchange of enterprise models. It therefore builds on previous languages and provides constructs to cover function, information, resource and organization aspects of business entities. It is also aligned with results of CEN TC 310, ISO TC 184 and IFAC-IFIP GERAM efforts in the area of enterprise modelling and engineering.*

**KEY-WORDS :** *Enterprise Modelling, Modelling languages, UEML*

## 1. INTRODUCTION

In today's highly competitive global economy, many business companies need to better understand and harness the way they operate. They must also frequently realign their organisation structure to face the need for change as imposed by their environment for increased customer satisfaction in terms of quality, cost and delay.

The efficient design, analysis and optimisation of enterprise operations require notations, formalisms, methods and tools to depict the various facets of a business organisation. All the aspects of this modelling process are collectively called Enterprise Modelling.

Enterprise Modelling (EM) can be defined as the art of externalising enterprise knowledge which adds value to the enterprise or needs to be shared (Vernadat, 2000). The prime objective is to describe the things of the enterprise. This usually concerns function, behaviour, information, resource, organisation or economic aspects of a business entity, be it part of a single enterprise, one enterprise or a network of enterprises. The goal is to represent or formalize the structure and behaviour of enterprise components and operations in order to understand, engineer or re-engineer, evaluate, optimise, and even control the business organisation and operations (Rolstadas, 1995; Scheer, 1999; Vernadat, 1996). These models should be usable for simulation.

Enterprise Modelling has its roots in functional modelling (e.g. SADT), information modelling (e.g. relational and entity-relationship models) and data flow diagrams (DFD) which emerged in the mid-70's for the purpose of software engineering and information system analysis. Models produced were mainly static. In the 80's, specific methods have been proposed to model large manufacturing systems to support their design, possibly with a link to simulation (e.g. IDEF suite of methods, GRAI method). EM has then significantly improved shifting from an activity (or function) centred view to a more business process centred view as advocated by the ESPRIT Consortium AMICE and its CIMOSA architecture in the late 80's (AMICE, 1993). The models became computer executable to make possible enterprise engineering (using model animation) and business process enactment (Bussler, 1996). Furthermore, EM has adopted the object orientation with the emergence of object-oriented approaches in the 90's (Mertins *et al.*, 1995). Finally, ontological approaches are being proposed to capture more of the semantics of aspects of the business modelled (Fox and Gruninger, 1998; Uschold, 1997) and to support systems interoperability, for instance with PIF (Process Interchange Format) (Lee *et al.*, 1998) or PSL (Process Specification Language) (NIST, 1999).

After CIMOSA, which was developed as a European pre-normative project to pave the way for Enterprise Modelling and Integration (EMI) technology, many EM languages (e.g. IDEF3, IEM, event process chains or DEM) and many EM tools appeared on the market place all over the 90's (e.g. ARIS Toolset, FirstSTEP, NCR Metis, Bonapart, Enterprise Modeller, PROPLAN, PrimeObject, MOOGO, CimTool, IMAGIM, to name a few among the 50 which could be listed). The situation became even worse with the emergence of workflow management systems and their myriad of commercial tools which also include business process modelling and execution facilities (WfMC, 1996).

This has created a **Tower of Babel situation** for business users having to use enterprise modelling. Any time end-users utilise a new tool, they have to learn a new language although these tools address the same aspects of the enterprise, i.e. to model processes, activities and objects of the enterprise. The situation is made even worse by the fact that all these tools are not interoperable and they cannot exchange models!

During the last ICEIMT (International Conference on Enterprise Integration and Modelling Technique) in 97 (Kosanke and Nell, 1997), this situation was recognised and the idea of launching a project to define UEML (Unified Enterprise Modelling Language) as a common user language with clear syntax and semantics was proposed. At the IFAC World Congress (Beijing, July 1999), it was decided to have a working group on UEML as one of the two actions of the IFAC-IFIP Task Force on Architectures for Enterprise Integration.

The aim of this paper is to present the principles and rationale of UEML, to outline the set of its core constructs and to report on current achievements.

## 2. EM SCOPE AND APPLICATION DOMAINS

Before presenting the aim and basics of UEML, it may be necessary the remind the purpose and scope and application domains of Enterprise Modelling.

EM mostly consists in modelling the organisation structure, the business processes as well as enterprise objects and their flows for the part of an enterprise (or a network of enterprises) subject to analysis. The main drivers for performing an EM exercise may include, but are by no means limited to:

- diagnosis of a disorder of any type (regarding material, information, control, decision or any other flows)
- restructuring a system to correct an established disorder (i.e. ascertained lack of performance in the system)
- business process reengineering
- new system design or AS-IS system reengineering
- large scale systems integration (possibly resulting from a fusion or an acquisition)
- implementation of MES, ERP or PDM systems
- adapting the organisation structure to face business change (e.g. moving from a functional structure to a distributed structure made of autonomous units, joining a network of enterprises or implementing just-in-time operations)
- alignment or conformance to the norm (e.g. ISO 9000 or ISO 14000)
- development of managerial decision support systems
- better understanding of how the enterprise should work and how it really works (i.e. establish the process map)

Thanks to its capabilities of depicting enterprise flows and components as well as their behaviour either in a graphical form or in more or less formal way, Enterprise Modelling has the prime advantage of being a sound basis for building a *common view* or *consensus* about the things of the enterprise which can be shared by a large population of individuals of the enterprise. Some of these models can also be used by computers in the form of analysis, decision-support, simulation and even workflow systems to support the day-to-day operations of the business.

Furthermore, EM-based simulation can be applied for the following types of problems:

- system dimensioning, i.e. deciding the position, type and number of elements a system must have to achieve its mission. For instance, defining location and capacity level of storage areas or processing units in a supply chain
- parameter optimisation, i.e. iterative tuning of some system parameters. For instance, deciding within a given range the optimal number of processing units to use to obtain a certain performance level
- design of control policies, i.e. defining the best control or co-ordination mechanisms to be put in place to ensure efficient operations of the system. For instance, define an efficient maintenance policy
- system operation checking, i.e. verifying that system behaviour and dynamics comply to the requirements definition. This is usual system simulation which must be performed for several experimental hypotheses to test most of the system operation situations. For large and complex enterprise systems, distributed simulation may be required (Vernadat and Zeigler, 2000)
- performance evaluation, i.e. assessing the performance levels which can be expected from the system. For instance, assessing the throughput times or the production rate deterioration in the case of equipment failures
- comparison of several design alternatives to select the most suitable one
- decision support, e.g. simulating the future behaviour of the system on the basis of the current state of the system to anticipate forthcoming problems

All these activities are typical activities involved in enterprise engineering which heavily relies on enterprise modelling and simulation. Enterprise Engineering (EE) can be defined as the art of designing enterprise systems, be they goods manufacturing systems, process industries, service industries or administrative systems.

GERAM, the Generalised Enterprise Reference Architecture and Methodology of the IFAC-IFIP Task Force (1999), provides a reference architecture, GERA, which summarises the main principles for enterprise modelling and engineering (Fig. 1). GERA results from the comparison and consolidation of three major reference architectures for CIM: CIMOSA, GRAI-GIM (Doumeingts *et al*., 1992) and PERA (Williams, 1994). The architecture is organised along three axes:

- Life-cycle activity types: the structure suggests that modelling can be used all over the enterprise life cycle from business entity identification to decommissioning except at the operation phase
- Generic levels: these are made of three layers: (1) a generic layer providing modelling constructs and rules to build models for the various life-cycle activity types, (2) a partial layer containing predefined, standardised or not, partial models which can be used and customised to particular needs, and (3) a particular layer containing the particular models of a given enterprise.
- Views: these are the modelling views to be considered in the particular models to manage complexity and abstraction levels of complex models. They include, but are not limited to, function/process, information/object, resource/agent and organisation/decision views.

GERA has now become part of ISO (ISO TC184/SC5/WG1 IS 15704) and European (CEN TC310/WG1 ENV 40003) standard documents.

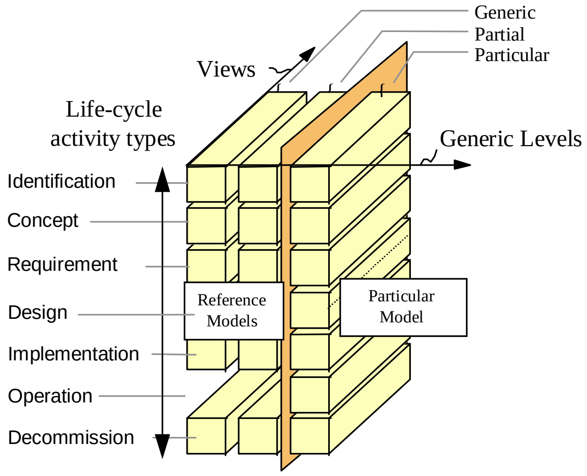

Figure 1. GERA structure

## 3. UEML RATIONALE AND MOTIVATIONS

The many problems and application domains mentioned in the previous section have raised an ever growing interest for enterprise modelling all over the 90's both on the side of IT vendors and academia. However, on the side of business users this uncontrolled development of many similar but diverse languages and related tools has led to the current Tower of Babel situation which can be summarised by the following facts:

- there are too many EM languages to learn and to understand as well as too many EM tools with completely different interfaces
- there is instability of vocabulary and modelling paradigms (from one method to another, the same concept happens to be modelled differently and the same term may refer to different things)
- there are many incompatible EM tools which are not able to inter-operate and which can hardly exchange models
- there are no or poor formal foundations both for EM and EE

The consequence is that there is a refrained interest from business users because they cannot capitalise on previous modelling projects, which usually represent significant efforts in terms of time and money invested.

Recognising this situation, the IFAC-IFIP Task Force on Architectures for Enterprise Integration has set up a working group to investigate the feasibility of a unified enterprise modelling language (UEML) which would provide a consensus in the field and populate the generic layer of GERA. The mandate of the UEML working group of the Task Force reads as follows:

"To define requirements and prepare the ground for a unification process that could lead to the development of a unified EM language by
1) Establishing working collaboration with all relevant groups in the area of enterprise modelling and simulation
2) Proposing a road-map that would lead to a technically feasible and politically acceptable EM solution for relevant user and vendor communities"

To this endeavour, links have been established with the CIMOSA Association, OMG (Object Management Group), WfMC (Workflow Management Coalition), ODP (Open Distributed Processing) and related standardisation bodies (CEN TC310, ISO TC184 and ISO/IEC JTC1). The following languages and tools have been analysed in a first phase of the work as main contributions relevant to the field.

### 2.1. Enterprise modelling languages

CIMOSA: The CIM Open System Architecture (AMICE, 1993; Vernadat, 1996) provides an enterprise modelling framework which contains a rich set of advanced modelling constructs covering function, information, resource and organisation aspects at several modelling levels (requirements definition, design specification and implementation description) of an enterprise and in an integrated way. The modelling language is based on an event-driven process-based approach which views the business entity as a large set of concurrent processes executed by communicating agents (Vernadat, 1998).

ARIS: The Architecture of Integrated Information Systems proposed by Prof. Scheer (Scheer, 1992, 1999) follows CIMOSA ideas in terms of modelling levels (requirements definition, design specification and implementation description) and integrated modelling (i.e. providing a set of integrated and non redundant modelling constructs). It differs in the structure of modelling views (function, control, data and organisation views) and uses a different modelling language based on process-event chains to model business processes.

IDEF suite: The IDEF family of languages (Menzel and Mayer, 1998), comprising IDEF0 (or SADT), IDEF1x (or EXPRESS-G), IDEF3 and IDEF4, is the most widely used set of modelling techniques in North America. IDEF languages have been used for the purpose of DoD and DoC large projects (e.g. ICAM, IPAD, CALS, EIF). IDEF0, 1 and 2 are the ancestors of enterprise modelling. However, these are non integrated modelling methods in the sense that each method models specific aspects of the enterprise and the same aspect can be modelled differently by two or more methods.

IDEF*: This is a method proposed by the Gintic Institute of Manufacturing Technology, Singapore, aiming at integrating IDEF0 and IDEF1x in a single-environment multi-view CASE tool to ensure consistency of IDEF0 and IDEF1x models produced for one system (Ang *et al.*, 1999).

IEM: The Integrated Enterprise Modelling approach proposed by IPK-Berlin (Mertins *et al.*, 1995; Spur *et al.*, 1996) is fundamentally based on the IDEF0 activity construct but in addition advocates a strong object-orientation for business process modelling. It primarily considers only two modelling views: function view and information view. IEM defines three fundamental types of object classes in any enterprise: Orders (i.e. objects stimulating execution of activities), Products (i.e. objects that are processed) and Resources (i.e. objects executing the activities).

WfMC meta-model: The Workflow Management Coalition, a consortium of workflow system vendors, users and researchers aiming at promoting the use of workflow technology (WfMC, 1996), has produced a meta-model for its Workflow Process Description Language (WPDL) and Process Definition Interchange (WfMC, 1997). WPDL is the neutral language used for exchanging process definitions among workflow systems.

ODP Enterprise Language: The Open Distributed Processing group has published a Reference Model of Open Distributed Processing (RM-ODP) which defines an Enterprise Language to enable full enterprise viewpoint specification of an ODP system (ODP, 2000). The language development is part of the work of the joint technical committee ISO/IEC JTC 1/SC33.

PSL/NIST: NIST is supporting the development of PSL (Process Specification Language). The aim of PSL is to provide a standard representation of process to allow companies to clearly and precisely document and share their manufacturing processes among different functions both within and among organisations (NIST, 1999). PSL defines a process as one or more activities that occur over a period of time in which objects participate. The work is supported by ISO activities in its TC 184.

### 2.2. Enterprise modelling tools

Many enterprise modelling tools could be mentioned. We only refer to the following ones because they relate to the previous modelling languages.

ARIS Toolset: This tool from IDS Scheer is the world leader in terms of market sales. ARIS Toolset is based on the ARIS architecture and is certainly the most complete tool on the market, although incomplete according to the GERA coverage. It is targeted at business process and organisation modelling as well as information system implementation.

FirstSTEP: This is a Canadian tool by Interfacing Technologies which has some CIMOSA flavour. It is organised around five major constructs: organisation units, processes, activities, resources and materials (to describe activity inputs and outputs). It includes a simulation engine and is targeted at business process improvement and what-if scenario analysis at the strategic management level.

KBSI offer: Knowledge Based Systems Inc. (KBSI) is a company based at College Station, TX. Its offer includes AI0 Win, an IDEF0 tool to model activities, ProCap, an IDEF3 tool to model business processes, ProSim, a simulation module based on Witness to simulate IDEF3 models, SmartER, an IDEF1x tool to model information schemata, and SmartCost, an activity-based costing tool.

NCR METIS: METIS is an enterprise and knowledge modelling platform that is built from software tools and a set of generic methodologies and models. Its open, web-based architecture makes METIS extensible and customisable to specific user-defined constructs.

MOOGO: This tool, marketed by PSI, Berlin, is the software tool that supports the IEM methodology. It can be used to document business processes, to produce quality management manuals, to produce organisational manuals and to plan process control procedures.

PrimeObjects: This tool has been developed by PrimeSoft, Turin. It models business processes using a Petri net-based formalism. This way, models become operational and can be executed therefore enabling validation and performance evaluation of the system being analysed through simulation and animation.

IMAGIM: This tool, marketed by GRAISoft, Bordeaux, implements the GRAI grid of the GRAI method and provides support to analyse the decision system of a manufacturing organisation (Doumeingts *et al.*, 1998).

CimTool: This tool, available from René Gaches Consultants (http://www.rgcp.com), is a CIMOSA tool organised around the following constructs: domain process, business process and enterprise activity for the function view, object view and enterprise object for the information view, and capability set for the resource view. It is limited to business process and information model description and applies to requirements definition.

A comparison and consolidation effort was made by CEN TC 310 on the basis of these languages and tools to identify the set of constructs necessary to cover all modelling views of ENV 40 003. This set of constructs, subject to revision in 2001, is documented in CEN ENV 12204 (CEN, 1995) and is summarised by Figure 2. It provides a starting point for the UEML definition.

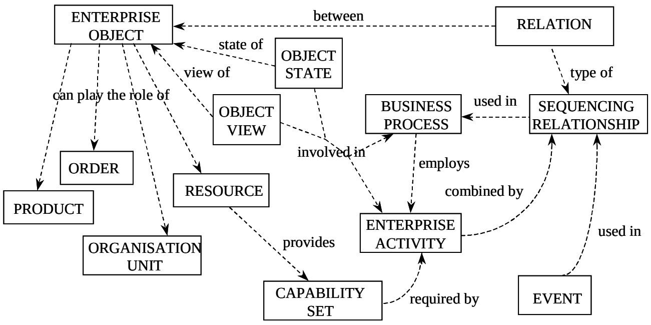

Figure 2. CEN ENV 12204 set of constructs

### 2.3. The need for UEML

The idea of proposing a unified language for enterprise modelling is not to propose the ultimate language supposed to replace all existing EM languages.

The idea is first to provide a language which is easy to learn and easy to use and which can be provided as a standard user interface on top of existing systems. Therefore, business users need to know only one language to access many different systems. The second idea is to provide a standard syntax and semantic definition so that particular models stored in one system can be translated to the UEML format to be exported towards other systems, making model exchange and reuse possible among special-purpose systems.

UEML is intended to provide a consensus in the enterprise modelling community, both in terms of terminology to be used and structure of concepts to be represented. It must be based on a meta-model (and underlying ontologies) widely accepted by business users and tool developers from which meta-models of specific systems can be mapped. Obviously, the perfect match will never exist, but a reasonable match can be envisioned, at least for the set of constructs of ENV 12 204, which is at the core of EM. A preliminary step to UEML development is therefore a comparison of the meta-models of existing languages and tools.

## 4. UEML PRINCIPLES

Based on the current understanding and usage of Enterprise Modelling, the development of UEML must take into account the following principles:

a) The language will be defined as a finite set of constructs, i.e. structured templates defining construct properties for which syntax and semantic rules need to be defined.

b) Principle of separation of processes and resources: The language is compliant with a vision which views a business entity (whatever its size may be) as (1) a large collection of concurrent business processes and (2) a large collection of communicating resources with a clear separation between the two (no fixed coupling between processes and resources).

c) Principle of separation of enterprise behaviour and enterprise functionality: Enterprise functionality refers to the things to be done while enterprise behaviour refers to the order in which things are done. Separate constructs need to be provided for enterprise functionality and behaviour (usually activity versus process) so that one can be changed in the model without necessarily having to change the other. This provides greater system flexibility for the management of change.

d) Principle of separation of resources and organisation units. Organisation units, loci of decision, must not be confused with resources, the doers. For instance, a person must not be confused with his/her position. An organisation unit is defined for the purpose of an organisation structure while a resource can be allocated to different organisation units according to needs.

Finally, one point must be clarified: UEML versus UML. The name UEML may remind UML (Unified Modelling Language) (Rumbaugh *et al.*, 1999) and was chosen intentionally. However, UEML and UML must not be confused. UML is a general purpose modelling language primarily dedicated to information systems but widely applicable. UEML is a specialised language dedicated to enterprise modelling and as such, it is made of constructs based on concepts which are natural to business users (e.g. process, activity, resource, event, …). Therefore, UML, which has been found too general for enterprise modelling, is a suitable tool to help in the specification of UEML constructs (as used in Figure 3 of the next section).

## 5. UEML CORE CONSTRUCTS

Like PSL, it is planned that UEML will be made of a set of core constructs and sets of additional constructs. Core constructs are constructs which form the basis of any UEML description, while additional constructs will only be used as specialised constructs for dedicated sectors or applications (e.g. product data management, activity-based costing, medical services, electronics industries, etc.). To be a UEML compliant system, the modelling system must support all core constructs and eventually some of the sets of specialised constructs.

This section presents the current set of UEML core constructs which includes (Figure 3):

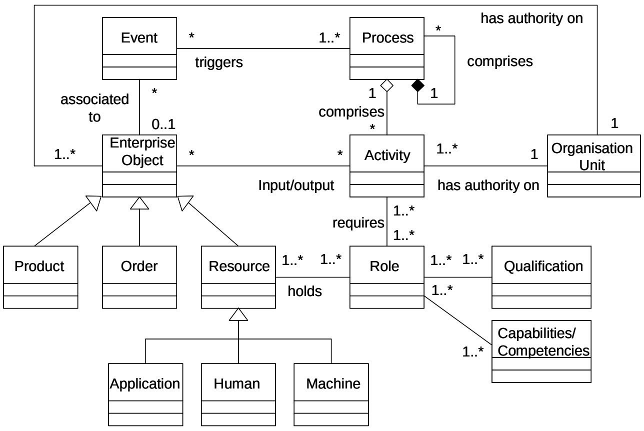

Figure 3. UEML core constructs

**Event** (or process triggering condition): An event depicts a change in the system state. It represents a solicited or unsolicited fact which will trigger a process. For instance, the arrival of a customer order, a machine failure, clock time = 5:00 pm, start or end of an activity are all examples of events. An event Ev can be formally defined as:

Ev = ⟨Evid, predicate, time, related object⟩

where *Evid* is the event identification, *predicate* is a Boolean expression which evaluates to true when the event becomes active, *time* is clock time when the event occurs and *related object* is an enterprise object associated to the event (for instance, the customer order in the case of a customer order arrival).

**Process**: A process is a partially ordered set of steps. Process steps can be sub-processes or activities (i.e. elementary steps). Processes are triggered by one or more event occurrences. A process P can be formally defined as:

P= ⟨Pid, alphabet, triggering-condition, behaviour⟩

where *Pid* is the process identification, *alphabet* is the set of process steps, *triggering-condition* is the Boolean expression made of triggering events which needs to be true to start the process and *behaviour* defines the control flow (or workflow) of the process. The control flow requires control structures such as sequential control, conditional branching, parallel spawning (synchronous or asynchronous) and rendez-vous (or synchronisation). Loop structures are usually constructed by means of an exclusive branching.

**Activity**: An activity is the locus of action which transforms inputs into outputs over time by means of resources. An activity A can be formally defined as:

A = ⟨Aid, pre-cond, input, output, req-roles, post-cond⟩

where *Aid* is the activity identification, *pre-cond* is a set of pre-conditions to be satisfied to enable activity execution, *input* defines the input object flow, *output* defines the output object flow, *req-roles* indicates the role(s) to be played by the executing resource(s) and *post-cond* defines the set of post-conditions (for instance, ending statuses as found in CIMOSA). To capture more of the semantics of activities, input and output flows can be categorised as control, function and resource flows.

**Enterprise Object**: An enterprise object is any entity which is used, processed, transformed or created by activities in the day-to-day operations of the enterprise. Enterprise objects and their states are the elements involved in the input/output flows of activities. Enterprise objects are defined by their properties, i.e. attributes for static properties and methods for behavioural properties. They can be related to one another by three kinds of relationships: *Is-a* relationship or generalisation abstraction mechanism, *Part-of* relationship or aggregation abstraction mechanism, and *association* relationship or user-defined link (this latter case is defined by means of an attribute the values of which are entities from another enterprise object class). Enterprise objects can be categorised into three main classes as suggested by IEM: Product, Order and Resource. An enterprise object EO can be formally defined as:

EO = ⟨EOid, isa, partof, properties⟩

where *Eoid* is the enterprise object identification, *isa* indicates if the object is a sub-type of a more generic object, *partof* indicates if the object is a component of a compound object and properties gives the list of properties describing this enterprise object class.

**Resource**: This is a special class of enterprise objects used in support to the execution of activities. As such, it inherits all properties of enterprise object just defined previously but in addition it gives the list of roles that the resource can hold, the resource availability (usually in the form of a calendar) and the resource capacity. A usage cost per unit of time can also be given. Three generic classes of resources, from which all other types can be derived, can be defined as suggested by CIMOSA: IT applications, humans and machines (instrumented with a control device).

**Organisation Unit**: An organisation unit defines an element of an organisation structure provided with authority and responsibility on identified activities and enterprise objects of the enterprise. It defines a decision centre at a certain decision level (position, department, division, direction, …) characterised by a given planning horizon with a given review period as suggested by GRAI. An organisation unit OU can be formally defined as:

OU = ⟨OUid, responsible, responsibilities, authorities, horizon, period⟩

where *Ouid* is the organisation unit identification, *responsible* is the person in charge of this organisation unit, *responsibilities* is the set of responsibilities assigned to the organisation unit, *authorities* is the set of authorities exercised by the organisation unit, *horizon* is the timeframe on which decisions are made and *period* is the time-scale after which decisions can be revised.

It can be noticed that this minimal set of constructs complies with the basic principles stated in section 4, especially the behaviour and functionality separation principle (process construct and activity construct), the process and resource separation principle (activity construct separated from resource construct by role) as well as the resource and organisation unit separation principle (resource construct and organisation unit construct).

## 6. CONCLUSION

The basics for the development of a unified language for enterprise modelling have been presented.

It is believed that providing the user community with one universal language, available on most EM platforms, would significantly leverage awareness and use of enterprise modelling technology and make possible the wide exchange of models as required by enterprise engineering.

At the present time, detailed requirements for UEML are being established by the UEML working group of the IFAC-IFIP Task Force while a European consortium is being set up to propose and demonstrate a first version of UEML in connection with software developers and the community of business users. A first version of UEML publicly released is expected by 2002.

## ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks to all members of the IFAC-IFIP Task Force involved in the UEML working group for fruitful discussions and material provided.

## REFERENCES

AMICE, 1993. *CIMOSA: CIM Open System Architecture, 2nd revised and extended edition*, Springer-Verlag, Berlin.

Ang, C.L., L.P. Khoo and Gay Robert, 1999, IDEF* - A Comprehensive Modelling Methodology for the Development of Manufacturing Enterprise Systems, *International Journal of Production Research*.

Bussler, C., 1996. Specifying enterprise processes with workflow modelling languages, *Concurrent Engineering: Research and Applications*, Vol. 4, No. 3, pp. 261-278.

CEN, 1995. ENV 12 204: Advanced Manufacturing Technology. Systems Architecture. Constructs for Enterprise Modelling. CEN, Brussels.

Doumeingts, G., B. Vallespir, M. Zanettin, D. Chen, 1992. GIM: GRAI Integrated Methodology, a methodology for designing CIM systems, Tech. Rep., GRAI/LAP, Bordeaux, France, Version 1.0.

Doumeingts, G., B. Vallespir, D. Chen, 1998. GRAI Grid Decisional Modelling. In *Handbook of Information Systems*, P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer-Verlag, Berlin. pp. 313-337.

Fox, M.S., Gruninger, M., 1998. Enterprise Modelling, *AI Magazine*, AAAI Press, Fall. pp. 109-121.

IFAC-IFIP Task Force, 1999. GERAM: Generalized Enterprise Reference Architecture and Methodology, Version 1.6.3, IFAC-IFIP Task Force on Architecture for Enterprise Integration.

Kosanke, K. and J.G. Nell (Eds.), 1997. *Enterprise Engineering and Integration: Building International Consensus*, Springer-Verlag, Berlin.

Lee, J., M. Gruninger, Y. Jin, T. Malone, A. Tate, G. Yost, 1998. PIF: The Process Interchange Format. In *Handbook of Information Systems*, P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer-Verlag, Berlin. pp. 167-189.

Menzel, C. and R. Mayer, 1998. The IDEF family of languages. In *Handbook of Information Systems*, P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer-Verlag, Berlin. pp. 209-241.

Mertins, K., H. Edeler, R. Jochem, and J. Hofmann, 1995. Object-oriented modelling and analysis of business processes. In *Integrated Manufacturing Systems Engineering*, P. Ladet and F. Vernadat (Eds.), Chapman & Hall, London, pp. 115-128.

NIST, 1999. PSL: Process Specification Language. http://www.mel.nist.gov/psl.

ODP, 2000. Information Technology – Open Distributed Processing – Reference Model – Enterprise Language, ISO/IEC 15414, ISO/IEC JTC 1/SC 7/WG 17.

Rolstadas, A., 1995. Enterprise Modelling for competitive manufacturing, *Control Engineering Practice*, Vol. 3, No. 1, pp. 43-50.

Rumbaugh, J., I. Jacobson, G. Booch, 1999. *The Unified Modeling Language Reference Manual*, Addison-Wesley, Reading, MA.

Scheer, A.-W., 1992. *Architecture of Integrated Information Systems: Principles of Enterprise Modeling*, Springer-Verlag, Berlin.

Scheer, W.-A., 1999. *ARIS – Business Process Modeling*, Springer Verlag, Berlin.

Spur, G., K. Mertins, R. Jochem, 1996. *Integrated Enterprise Modelling*, Beuth Verlag, Berlin.

Uschold, M. *et al.*, 1997. The Enterprise Ontology, Res. Rep., AIAI, The University of Edinburgh, Scotland.

Vernadat, F.B., 1996. *Enterprise Modeling and Integration: Principles and Applications*, Chapman & Hall, London.

Vernadat, F.B., 1998. The CIMOSA Languages. In *Handbook of Information Systems*, P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer-Verlag, Berlin. pp. 243-263.

Vernadat, F.B., 2000. Enterprise Modeling and Integration: Myth or reality? Proc. 2nd Int. Conf. on Management and Control of Production and Logistics (MCPL'2000), Grenoble, France, July 5-8, 2000 (plenary talk).

Vernadat, F.B. and B.P. Zeigler, 2000. New simulation requirements for model-based Enterprise Engineering. Proc. IFAC/IEEE/INRIA Int. Conf. on Manufacturing Control and Production Logistics (MCPL'2000), Grenoble, 4-7 July 2000. CD-ROM.

WfMC, 1996. *Workflow Management Coalition. Workflow Handbook*, John Wiley & Sons, New-York.

WfMC, 1997. Workflow Management Coalition Interface 1: Process Definition Interchange, WfMC TC-1016.

Williams, T.J., 1994. The Purdue Enterprise Reference Architecture. *Computers in Industry*, 24 (2-3), 141-158.