



Enterprise modelling: from early languages to models transformation

Bruno Vallespir, Yves Ducq

► To cite this version:

Bruno Vallespir, Yves Ducq. Enterprise modelling: from early languages to models transformation. International Journal of Production Research, 2018, 56 (8), pp.2878 - 2896. 10.1080/00207543.2017.1418985 . hal-01836662

HAL Id: hal-01836662

<https://hal.science/hal-01836662>

Submitted on 12 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ENTERPRISE MODELLING: FROM EARLY LANGUAGES TO MODELS TRANSFORMATION

Bruno Vallespir, Yves Ducq

Univ. Bordeaux, CNRS, IMS, UMR 5218, 33405 Talence, France

bruno.vallespir@ims-bordeaux.fr, yves.ducq@ims-bordeaux.fr

Abstract

During the last thirty years, enterprise modelling has been recognised as an efficient tool to externalise the knowledge of companies in order to understand their operations, to analyse their running and to design new systems from several points of view: functions, processes, decisions, resources, information technology.

This paper aims at describing the long evolution of enterprise modelling techniques as well as one of the future challenges of these techniques: the transformation of enterprise models.

So, in a first part, the paper describes the evolution of enterprise modelling techniques from the divergence era to the convergence period. In a second time, the paper focuses on the recent advances in the use of enterprise models through model driven approaches, interoperability problem solving and simulation, all these advances having the same characteristic to use the transformation of enterprise models.

Keywords: Enterprise modelling, modelling framework, meta-modelling, models transformation, model-driven.

1. INTRODUCTION

Formalizing the organization of companies is a necessity to obtain a better efficiency and allow them to evolve in relation with the market and the socio-technical environment. During the last thirty years, enterprise modelling thus stood out as a powerful mean in the service of the industrial organizations with a focus on performances improvement. A large number of tools and methods emerged at first, in particular in the United States and in Europe, highlighting scientific concepts today recognized at the international level with an asserted industrial application purpose.

This development led, according to the diverse needs for every study, to a significant number of tools and specific methods sharing however numerous concepts or aiming at identical application domains. In the face of this diversity, a second phase appeared who leads to a link, even to an integration of the existing approaches, rather than to an emergence of new ones. From then on, the problem was to define how to make modelling languages to communicate. Although based on common concepts stemming from more general theories as the systems theory, the graph theory or the information theory, the languages are nevertheless interpretable of different manners when they are concretely implemented. In summary, the development of enterprise modelling went at first through an era of brainstorming, followed by an era of convergence.

In this article, first of all we shall remind quickly the foundations of enterprise modelling and the main currents having led to the development of modelling languages. We shall then address the approaches the role of which is to move closer, to put in coherence and to integrate these concepts.

In parallel with the process of convergence which has been presented, other developments appeared around enterprise modelling and are presented for some of them. The purpose of these

developments is to ensure the alignment between models at the business level and at the technical level through model driven approaches, the modelling in situation of interoperability and the combine use of enterprise modelling and simulation. All these parts have an essential common point that is the use of model transformation. The purpose of this paper is to present the frames where models transformation take place, and some technical aspects of models transformation including an application case.

2. MODELS AND LANGUAGES: THE INITIAL ERA OF BRAINSTORMING AND DIVERGENCE

An enterprise is a complex reality that it is necessary to model for, on one hand, making it understandable and, on the other hand, allowing actors having a well defined project in mind to reason, decide and act. In other words, a model of a company has for objective to formalize all or a part of the company with the aim of understanding or of explaining an existing situation or to realize and then validate a designed project (Braesch *et al.* 1995).

The actions based on enterprise modelling concern the design, the evaluation and the reengineering of enterprises. Vernadat (1999a) considers that enterprise modelling is a discipline that consists of describing the organization and the operational processes of a company either with the aim of simulating these processes to compare diverse scenarios, or with the aim of analysing and restructuring them to improve the performance of the company. So, it emerges that there is no really a life cycle of enterprise modelling because it intervenes during all the phases of the life of the company: since its creation until its destruction and during its evolution due to the possible re-use of some of its components.

A model is always built on the basis of a language that could be informal (natural language, for example), semi-formal (language essentially graphic for example) or formal (mathematical language). Most of the time, the models based on an informal language are used to describe an existing situation whereas the models based on a formal language allow the check of the properties required in a given project (Chapurlat *et al.* 1999, Kamsu-Foguem and Chapurlat 2006).

Kosanke *et al.* (1999) state that a model is a representation of an abstraction of a part of real world, expressed in a language of representation. So, a model of company is always associated with a purpose and it must, according to needs, be capable of taking the structural, functional and behavioural aspects into account. Besides these aspects, it must also be capable of understanding the particular point of view of an actor. Finally, a specific approach is often associated with every modelling language. This approach clarifies the various stages necessary for the building and for the exploitation of the model (life cycle).

Doumeingts *et al.* (1995) consider that a model is a representation of knowledge to be communicated without ambiguity. Neaga and Harding (2005) go towards the same direction by linking enterprise modelling and knowledge management.

The supplied models have also to perform a role concerning the human actors of the project. Indeed, they have to play a role of support in:

- the communication inside the project, in space (exchange between actors of different skills and belonging to structures or departments with different purposes) and in time (archiving and documentation of the project); and

- the decision of design or reengineering of the system.

These two requirements result in one of the major difficulties of enterprise modelling. On one hand, it has to propose languages leading to models allowing the valuation of the system and from which a calculation approach can be implemented (evaluation). On the other hand, the obtained models must be easily understandable by the actors of the company and allow to gather the various advices and expertise, while taking the behavioural variability of the system into account.

These various observations explain why the works led within the framework of enterprise modelling gave rise to numerous languages and modelling tools. So, it is very difficult to supply an exhaustive list, especially as most of the disciplines having to deal with artificial complex systems made their contribution. We shall quote in particular the engineering of information systems, the automatic control and the industrial engineering. In the case of information systems, it must be noticed that this domain was always source of innovation in enterprise modelling whether it is through already old methods such as MERISE, more recent such as PROCESS UNITED or works such as those led by the OMG (Object Management Group) or the WfMC (Workflow Management Coalition), which have an incidence on enterprise modelling, in particular within the framework of the integration of industrial applications.

Within the framework of industrial engineering, the first developments were led in the United States in the 70s and produced in particular SADT, SSAD, IDEF0, Data Flow Diagram (Doumeingts *et al.* 1992, 1995). Concerning Europe, the programs of the European Commission widely allowed developing and spreading these tools, this since the first Framework Program of Research and Development (Esprit program, the 80s). Since then, a large number of languages of enterprise modelling appeared in North America and in Europe. We can quote for example MERISE (Tardieu *et al.* 1983), GRAI (Chen *et al.* 1997), (Vallespir and Doumeingts 2007a, 2007b), NIAM (Abrial 1974), M* (Di Leva *et al.* 1987), CIMOSA (AMICE 1993), (Kosanke *et al.* 1999), (Vernadat 1998), (Zelm 1995, 2001), OMT, IEM/MO2GO (Mertins and Jaekel 1998), IDEFx (Menzel and Mayer 1998), METIS and AKM (Lillehagen and Krogstie 2002) or ARIS Toolset (Scheer and Nüttgens 2000, Scheer 1994, 2000, Scheer and Schneider 1998).

At the end of the 80s, fewer new languages appear as if, due to being each other inspired, the variety of the used concepts reached a relative stability. The problem changed then. Based on the observation of their complementarity, it appears a need to move closer, to assemble even to integrate these languages to obtain more powerful and coherent methodological sets. It is the beginning of the era of convergence.

3. THE ERA OF CONVERGENCE

Several stages punctuate this convergence. It begins with the development of multi-languages methods, continues by the analysis of the methodological components and the standardization and next by the semantic fusion of the languages (Vallespir *et al.* 2003, Vallespir 2003).

3.1. First stage of convergence: the interoperability of languages supported by modelling

frameworks

The modelling languages allow to instantiate generic concepts within the framework of the modelling of the studied system. These formalisms are enough important in number to allow to cover all the requirements of the domain by taking into account:

- The diverse objectives of the modelling (analysis of the performances of the system, upgrade of an existing system, design of a new system),
- Considered points of view (physical system, control system),
- Levels of analysis or questioning (functional or semantic level, level of components supporting these functions), and
- Levels of detail (analysis of the system as a whole or focus on a precise point).

So, considering this, when a methodological approach aims at a wide cover of modelling, it uses several languages. The problem of the consistence of the various models is then put. A solution is to propose a modelling framework which characterizes the relative positioning of models, the overlapping areas and the mechanisms to go from one to another.

The first approach having put clearly this concept is certainly MERISE which, by proposing three levels of abstraction (*conceptual*, *organizational / logical* and *realizational / physical*) and two views (*data* and *processes*), leads to a modelling framework with six domains of modelling. Each of these domains uses a language (for the most abstract levels) or several in the case of the lowest abstract levels.

Other methods followed the same way. They propose different views in kind and in number, different levels but the underlying concept of modelling framework remains the same. Among these approaches, let us quote in particular GIM (Doumeingts *et al.* 1992, Chen *et al.* 1997), CIMOSA (AMICE 1993) and the Zachman framework (Zachman 2008).

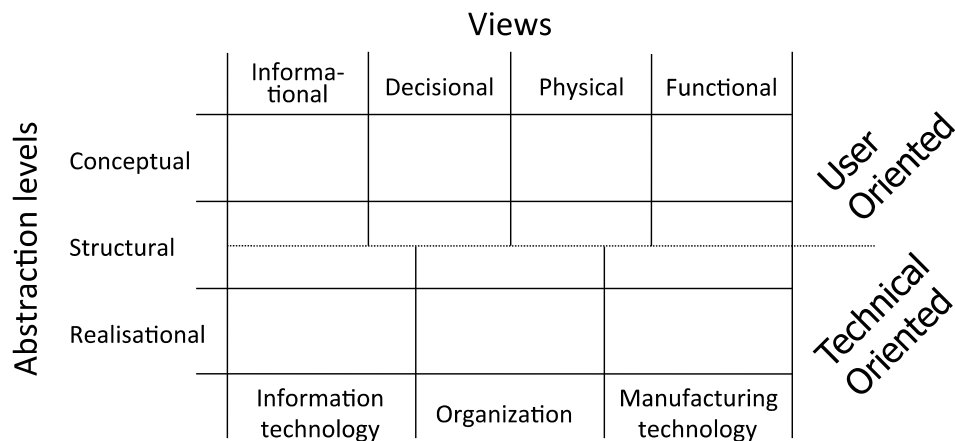


Figure 1. The GIM modelling framework

The GIM modelling framework is composed of two axes (Figure 1):

- An *abstraction levels* axis based on a multi-strata approach and allowing the modelling of the system according to several semantic levels considering the required depth of analysis (*conceptual* level: no consideration about organization or techniques, *structural* level: consideration of the options of organization, *realizational* level: consideration of the technical options); and
- A *views* axis which allows to take into account several points of view stemming from a

systematic decomposition (for the upper part – user oriented): *decisional* view, *informational* view or *physical* view to which a *functional* view is added, allowing to present a simple model of the studied system and the relations with its environment.

The well-known CIMOSA cube follows the same type of reasoning with a three-dimension framework (Figure 2):

- The *Generation* dimension corresponds to the same concept as the GIM views and is composed of *Function*, *Information*, *Resource* and *Organization*;
- The *Derivation* dimension is linked to the life cycle of the system with *Requirement definition*, *Design specification* and *Implementation description* and is closed to the GIM abstraction levels; and
- The *Instantiation* dimension goes from the language to the specific model with *Generic*, *Partial* and *Particular*.

The CIMOSA cube has evolved within the ISO 19439 standard (ISO, 2006).

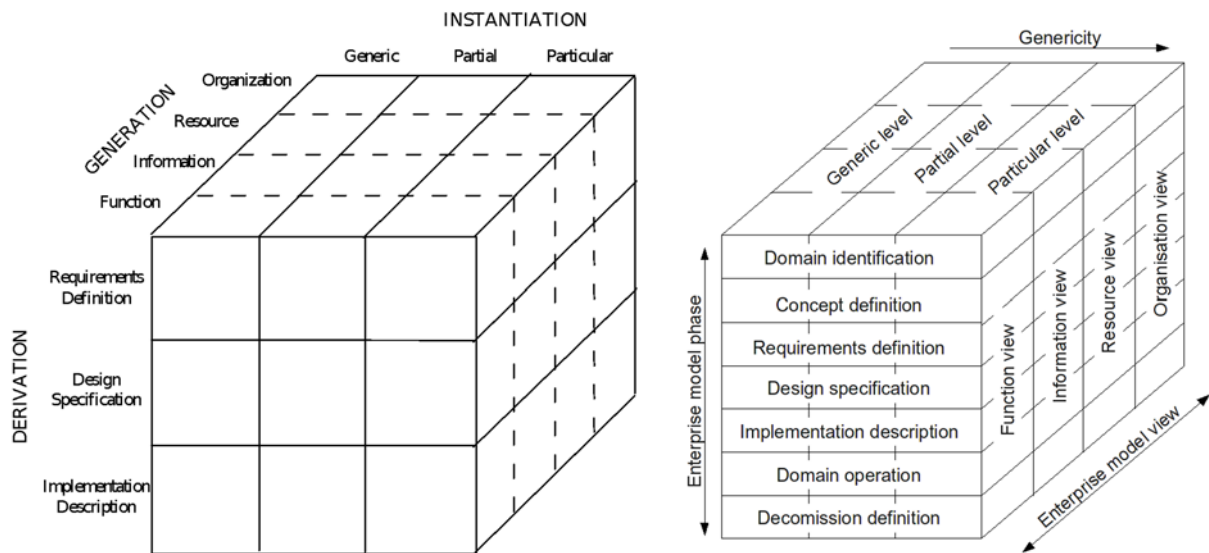


Figure 2. The CIMOSA cube (left) and its evolution within the ISO 19439 standard (right)

The Zachman framework (Hay 1997) has two dimensions (Figure 3):

- The first one is based on the same concept than the GIM *View* dimension or the CIMOSA *Generation* dimension and is composed of *Data* (answering the “What?” question), *Function* (How), *Network* (Where), *People* (Who), *Time* (When) and *Motivation* (Why); and
- The second dimension is closed to the GIM *Abstraction levels* dimension or the CIMOSA *Derivation* dimension with *Objectives / Scope*, *Model of the Business*, *Model of the Information System*, *Technology Model*, *Detailed Representation* and *Function System*.

Unlike GIM and CIMOSA, The Zachman framework does not propose any modelling languages. So, it does not really participate to the convergence and the interoperability of modelling languages.

	Data (What)	Function (How)	Network (Where)	People (Who)	Time (When)	Motivation (Why)
Objectives / Scope	List of things important to the enterprise	List of processes the enterprise performs	List of locations where the enterprise operates	List of organizational units	List of business events / cycles	List of business goals / strategies
Model of the Business	Entity relationship diagram (including m:m, n-ary, attributed relationships)	Business process model (physical data flow diagram)	Logistics network (nodes and links)	Organization chart, with roles; skill sets; security issues.	Business master schedule	Business plan
Model of the Information System	Data model (converged entities, fully normalized)	Essential Data flow diagram; application architecture	Distributed system architecture	Human interface architecture (roles, data, access)	Dependency diagram, entity life history (process structure)	Business rule model
Technology Model	Data architecture (tables and columns); map to legacy data	System design: structure chart, pseudo-code	System architecture (hardware, software types)	User interface (how the system will behave); security design	"Control flow" diagram (control structure)	Business rule design
Detailed Representation	Data design (denormalized), physical storage design	Detailed Program Design	Network architecture	Screens, security architecture (who can see what?)	Timing definitions	Rule specification in program logic
	(Working systems)					
Function System	Converted data	Executable programs	Communications facilities	Trained people	Business events	Enforced rules

Figure 3. The Zachman framework

3.2. Second stage of convergence: the macroscopic integration (GERAM)

GERAM (Generalised Enterprise Reference Architecture Methodology) was developed by a task force on the architectures for the enterprise integration (IFAC/IFIP Task Force) (GERAM 1999) based mainly on CIMOSA, GRAI/GIM and PERA (Williams 1996).

GERAM proposes an organization of the main concepts constituting the existing approaches. The final result is an articulation of a set of components (languages, structured approach, reference models, etc. - Figure 3) allowing to cover domains related to enterprise engineering and integration. Most of these components are based on works already widely recognized by enterprise modelling.

For example, GERA (Generic Enterprise Reference Architecture) supplies an analysis and modelling frame the structure of which was widely inspired by CIMOSA and PERA. Besides, EEM (Enterprise Engineering Methodology) makes a reference to modelling methodologies such as GIM or PERA. The latter use themselves languages sometimes supported by software tools gathered in EET / EML (Enterprise Engineering Tools / Languages). IDEF, IDEF3, CIMOSA and the GRAI grid on one hand and ARIS Toolset, FirstSTEP, MOGO (IEM) (Mertins and Jochem, 1998) on the other hand appear among the languages and the tools which may be selected.

The concepts are formalized in the form of ontologies, of meta-models or of glossaries in GEMC (Generic Enterprise Modelling Concepts). The models are, if possible, included in PEMs (Partial Enterprise Models). Finally, EMOs (Enterprise Modules) and EMs (Enterprise Models) contribute to the representation of all or part of the company and are used to guide the implementation of the operational systems of the company (EOS - Enterprise Operating Systems) and to estimate the current and future performances as well.

In conclusion, GERAM, being inspired by existing approaches, proposes an organized set of relevant concepts of enterprise modelling and integration as well as their relations (Vernadat,

1999a), (Figure 4). So, it is possible to determine the positioning and the potentialities of any approach with regard to this set. However, it constitutes only a theoretical framework of meta-modelling, non-operational thus and not dealing with the integration of the languages themselves.

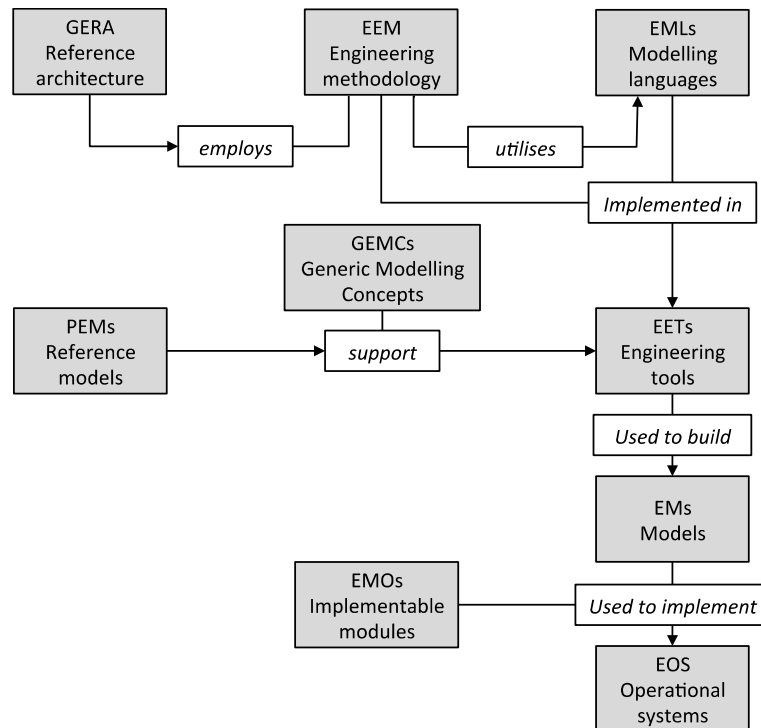


Figure 4. GERAM – Set of enterprise engineering and integration components

The fact remains that reference tables common and accepted by all are essential to succeed in federating the modelling approaches. This is why the standardization works constitute an important stage of the reflection.

3.3. Third stage of convergence: the standardization

To increase the quality of systems and propose work reference tables, numerous organizations as CEN (European Standardization Committee), ISO (International Organization for Standardization), ISA (Instrument Society of America), IEEE (Institute of Electrical and Electronics Engineers), OMG (Object Management Group) and the OAG (Open Application Group) defined a set of standards based widely on the concept of "industrial consensus" (Shorter 2000).

These bodies and working groups for standardization (Chen and Vernadat 2001) developed frameworks and basic constructs (Panetto *et al.* 2001, Chen *et al.* 2002a, Chen 2007) now reused and applied in related domains (ISA 95 for example). So, the concepts necessary for the modelling of the activity, the functions, the organization or the resources of a company, but also the rules of construction and manipulation of these concepts, are now established and approved by a more and more wide industrial, scientific and academic community. At the international level, the main actor in the field of enterprise modelling and integration is the technical committee ISO TC184 SC5/WG1 and CEN TC310/WG1. Figure 5 gathers the main CEN and ISO standards of the domain.

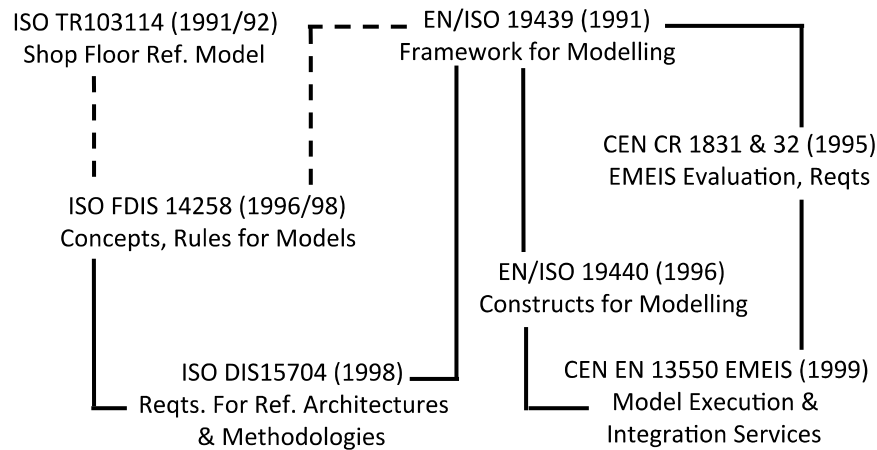


Figure 5. Main CEN and ISO standards of the enterprise modelling domain

3.4. Fourth stage of convergence: the microscopic integration of languages (UEML)

It is generally recognized that there are too many heterogeneous modelling languages available on the market. So, it is difficult for the users to understand them really and to choose the most adequate. Furthermore, every language has its own syntax, textual or graphic. Finally, even if they possess a defined syntax, most of the languages do not possess a clear semantics. However, as we have already mentioned, the concepts on which these various languages are based are similar or differ only in detail.

The initiatives around UEML (Unified Enterprise Modelling Language) (Petit *et al.* 1997, GRPGT5 1999) have turned to a consensus between the various actors and the various standardization bodies working in the field of enterprise engineering and integration. The result would be the development of a high-level pivot language allowing several modellers to exchange information, data and knowledge contained and already taken into account within their own models established by means of languages, of basic constructs and different methods (CIMOSA, GRAI, etc.). However, it is not obvious to get a really unified language that answers to all the problems of modelling that could occur. So, it is not possible to ignore the approach which would consist in making only communicate the computing tools rather than to integrate the languages.

This approach is based on the principles of integration of software tools by the data that they exchange. The current enterprise modelling languages would not then be questioned. This integration can basically correspond to the exchange of neutral files (said low integration as proposed in the past to describe the architectures of the control systems (PTA 1992)), with all possible losses or bad understandings of the information. For example, the PRIMA II ESPRIT project (EP 20775) allowed, by focusing only on CIMOSA, to define neutral formats of exchange by using the XML standard (Salvato *et al.* 1999). The integration of tools can also be based on a semantic identity of the data and information processed by tools (said strong integration as proposed to model and exchange data products (Bouaza, 1995)). The main problem is then to define these semantic links. This objective can be reached only with meta-modelling and the definition of constructs.

The development of a unified enterprise modelling language seems thus inescapable. The expected result has not for objective to be a new language that must replace those existing and

at present used but rather to be capable of interpreting the latter. UEML could be an "*Esperanto*" (Vernadat 1999b, 2001, 2002) in the field of enterprise modelling and engineering. It does not establish the ultimate enterprise modelling language replacing all the precedents but a standard meta-model (with its associated ontology) widely accepted by the users as by the developers of software tools. In other words, UEML does not aim at being integrated into *the analyst's toolbox*. Then, it is not constrained by operational usability criteria (Vallespir *et al.* 2001). Thus, UEML would be a compatible language with the most used operational languages.

In this sense, UEML can become a general conceptual base. The existing modelling languages become then operational interfaces (a projection of UEML in a specific semantic plane, Figure 6, Chen *et al.* 2002b). It means implicitly that UEML covers more than the individual modelling languages do. UEML so appears as the union of the existing languages.

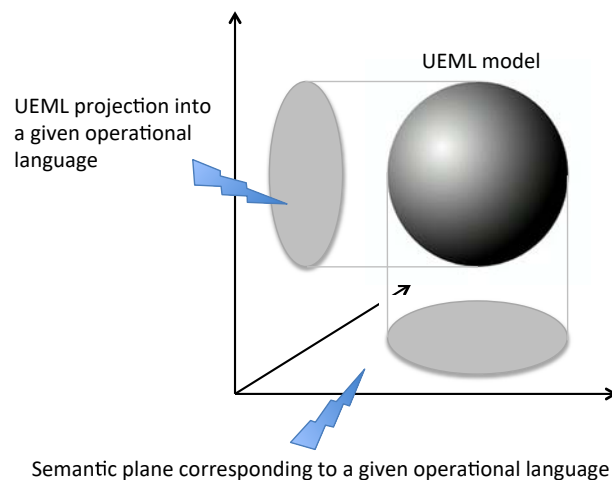


Figure 6. UEML position with regard to the operational languages (projection)

Several approaches are possible for the development of UEML: bottom-up, top-down or hybrid (Chen *et al.* 2002b).

The bottom-up approach begins with the analysis of the existing enterprise modelling languages. The approach can be structured into four stages:

- Choice of the existing enterprise modelling languages (state of the art and then choice of languages according to criteria established as usability, recognition, etc.),
- Decomposition of the chosen formalisms into elementary concepts,
- Union of the elementary concepts, and
- Fusion and establishment of a unique syntax.

The advantage of the bottom-up approach is to be fast and to avoid rediscovering what already exists. The inconvenience is that nothing insures that the chosen languages are representative.

The top-down approach is based on a conceptual analysis. It can be also structured in four stages:

- Precise definition of the needs and domain,
- Choice of paradigm (for example: systems theory),
- Definition of elementary concepts, and
- Definition of a syntax around these concepts.

The advantage of the top-down approach is the theoretical coherence. The problems are the time of development and the fact that existing formalisms risk certainly to be redefined.

An interesting compromise between these two approaches consists in using a hybrid approach that seems better adapted to the development of UEML. It is composed of the following stages:

- Precise definition of functional needs that UEML has to match,
- Choice of the existing languages in coherence with the required features and, at the same time, choice of one paradigm to ensure the theoretical coherence,
- Definition of a set of elementary concepts obtained by decomposition of the chosen languages,
- Union of the elementary concepts and elimination of the redundancy, and
- Definition of the syntax around the elementary concepts.

It is admitted that the development of UEML would contribute to a better interoperability and a capacity of communication between modelling actors in heterogeneous environment. This is the main objective. However, this development presents interests that exceed this operational objective because it allows:

- A clear definition of the common semantics of the languages and a better demarcation of the domain of enterprise modelling and engineering;
- A better definition of the scientific corpus of enterprise modelling and engineering and thus an increase of their visibility within the framework of the scientific community;
- A vocabulary accepted and used by standardization bodies at every level (national, European and international) working in the domain.

This observation is generalizable to all the points developed in this chapter about convergence. The modelling frameworks allowed to establish modelling domains in connection with the underlying modelling paradigms (systems theory for example); GERAM allowed to define the big methodological components and to answer consequently the question "what is an enterprise modelling method?"; the standardization participated in the organization of the concepts and the stabilization of the terminology and UEML is the opportunity to work and to organize semantics of the domain. At the end, this era of convergence is the era of the conceptual and semantic structuring of the domain.

In conclusion of this part, meta-modelling and models transformation explicitly arose within UEML initiatives and stay a hard issue. Even if UEML did not match the intended results because of a methodological lack at the level of meta-modelling, this domain remains a scientific challenge because, the models transformation is at the base of several current developments on enterprise modelling. The next chapter will present some of them.

4. CURRENT DEVELOPMENTS BASED ON MODELS TRANSFORMATION

Three approaches based on model transformation are described hereafter. They are:

- model driven approaches which aim to ensure the progressive modelling of the system from business to execution,
- the works on enterprise interoperability, and
- the combined use of enterprise modelling and simulation aiming at verifying the performance of the future system corresponding to the model.

4.1. Model driven approaches

Nowadays, a major topic frequently encountered in companies is the gap between the strategic vision in the process running and the technical activities and tools that are really implemented to do so. This alignment of the technical aspects with the strategic ones opposes clearly the business view to the technical view.

To combine both views and to separate the concerns, several approaches have been developed for around fifteen years under the label of model driven approaches. The approaches described below use enterprise models and then require model transformation to ensure the coherence between modelling level from the business view to the simulation.

The Model Driven Architecture (MDA) is the most illustrative even if some approaches were developed before with the same spirit of modelling levels. However, MDA can come in adaptations to specific situations.

4.1.1. Model Driven Architecture

MDA (OMG 2005) was defined and adopted by the Object Management Group (OMG) in 2001. It is designed to promote the use of models and their transformations to consider and implement different systems. It is based on an architecture defining four levels, which goes from general considerations to specific ones:

- CIM Level (*Computation Independent Model*) is describing the system and its environment. It contains function, decision and process models independent from implementation.
- PIM Level (*Platform Independent Model*) models the system sub-set that will be implemented.
- PSM Level (*Platform Specific Model*) focuses on parts of the development platform.
- Coding Level consists in coding Enterprise Software Application.

To complete this description, a Platform Description Model used for the transformation between PIM level and PSM level is added to these four kinds of models corresponding to four abstraction levels.

However, the idea of MDA to propose a multi-level architecture is not new. As presented previously in this paper, it was also at the origin of CIMOSA with the requirement / design / implementation levels using model derivation and model transformation. This spirit was also proposed in MERISE with conceptual / organisational-logical / realizational-physical levels and in GIM with conceptual / structural / realizational levels.

MDA is at the moment significantly used for IT service implementation. In major cases, the final stage of the MDA or other MD Development method is the generation of a Service Oriented Architecture SOA. An execution workflow language can be used like BPEL or simulation components. Obviously, MDA is an approach that must privilege enterprise modelling. Nevertheless the development is mainly IT-directed and the models are already built with this IT goal without placing always the users point of view and requirements at a central position. Moreover, in MDA, no real modelling languages are promoted at each level, leading the practitioner to develop his own architecture of models.

4.1.2. The Model Driven Service Engineering Architecture (MDSEA)

MDSEA is inspired from MDA. This methodology was developed in the frame of the FP7-

MSEE project (Manufacturing Service Ecosystem) (MSEE 2012). One objective of MSEE was to facilitate the transition of companies from a traditional business of product manufacturing to a challenged business of Product and Service manufacturing. MDSEA provides an integrated methodology dealing with modelling languages at various abstraction levels to support Service models and Service System design and implementation.

To better define and implement the service system supporting the service life cycle, it is again necessary to separate the preoccupations, from the user/business point of view to the technical point of view. The business point of view is more focused on the definition of the service product and service system, in particular from the business process, the decision and the information system modelling points of view, while the technical point of view is more focused on the progressive implementation of the service - product and service - system taking progressively the technical constraints into account. The MDSEA defines a framework for service system modelling based on three abstraction levels: BSM, TIM and TSM as well as the dedicated modelling languages at each level. Figure 7 presents the MDSEA.

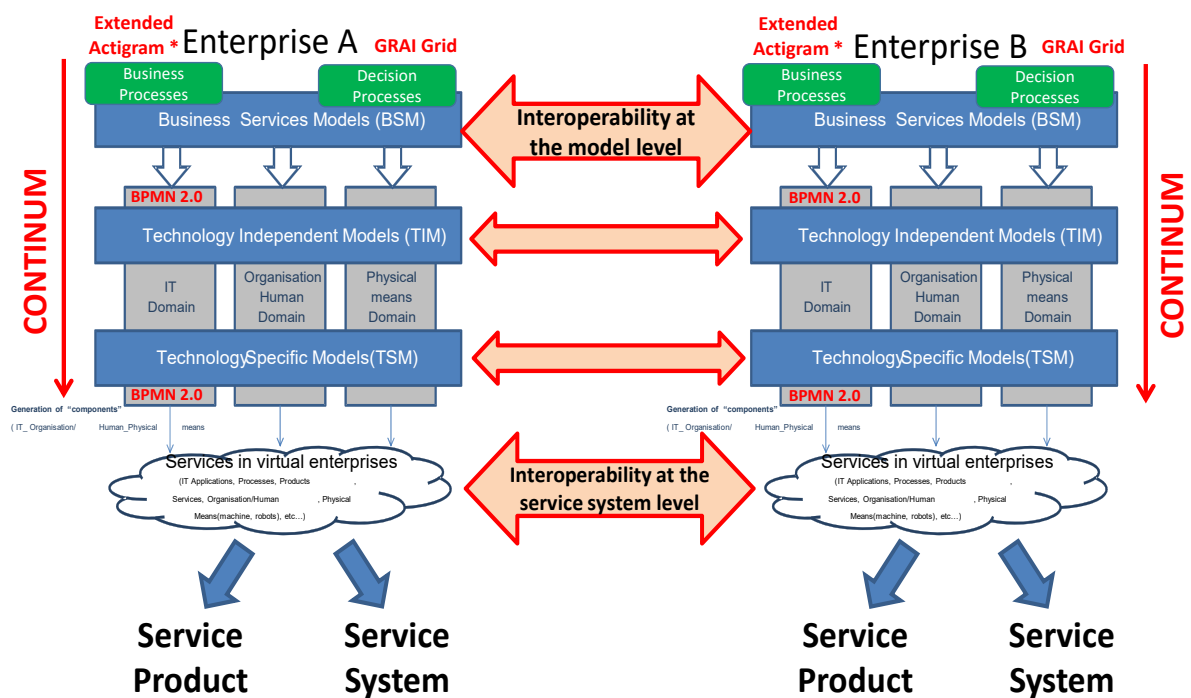


Figure 7. MDSEA: Architecture for interoperability and alignment of service system

Figure 7 shows a declination of MDSEA with two collaborative enterprises. Even if each level is detailed hereafter, the figure shows that the interest of such an architecture is on one hand to develop and implement at the same time a service - product and on the other hand a dedicated service - system coherent with business service models, represented by enterprise models. So, the consideration of BSM models for both entities at the same time ensures the interoperability at the model level that will then be ensured at the service system level, which is required to allow an efficient running of the service system network. It can be seen that from TIM to TSM level the methodology separates three kinds of elements. The resources to achieve the service are categorized into IT, Human and Physical Means in order to tackle the different consideration coming from the different natures of the resources at the implementation step of the service system. Then after implementing the cloud of services, it must be kept separated on one side the service - product that is proposed to the environment of the enterprise and on the other side the system to generate the services (the way to generate) that can be specific to the enterprise.

Business Service Model (BSM).

BSM specifies the system, at the global level, describing the service running inside a single enterprise or inside a set of enterprises as well as the links between these enterprises. These models can include business models to specify the value creation. The models at the BSM level must be independent of the future technologies and skills that will be used for the various resources. The languages used at this level must be simple to be used by non experts in modelling but also powerful to describe business aspects. The languages must represent the various points of view of service processes, service process control for the design and operation of the service system. At this level, MDSEA proposes to use GRAI Grid for decision modelling and Extended Actigram* (a process oriented language, Bazoun 2015) for business process modelling instead of BPMN in the sense that the language will not be used only for IT.

Technology Independent Model (TIM).

TIM delivers models at a second level of abstraction independent from the technology used to implement the system. It gives detailed specifications of the structure and functionality of the service system which do not include technological details. More concretely, it focuses on the operational details while hiding specific details of any particular technology in order to stay independent from any technology, used for the implementation. At TIM level, the detailed specification of the components of a service - system (as skills for a human resources or function of a machine or a function of a software) are elaborated with respect to IT, Organization/Human and Physical means involved within the production of the service. This is important to mention that in comparison to MDA or SOMA (Service Oriented Modelling and Architecture), the objective of MDSEA is not only IT oriented and then this requires enabling the representation of human and technical resources from the BSM level. At TIM level, the representations must add information in comparison to BSM models. Of course, the intention is to obtain models at TIM level using transformation mechanisms of BSM model. For instance, transformation mechanisms have been developed between extended actigrams at the BSM level and BPMN 2.0 at TIM level. At TIM level, it can be necessary to add detailed information. Also horizontal transformation mechanisms have been developed between BPMN 2.0 and DEVS still at TIM level (Zacharewicz *et al.* 2016).

Technology Specific Model (TSM).

TSM enhances the specifications of the TIM model with details that specify how the implementation of the system uses a particular type of technology (such as, for example IT applications, machine technology or a specific person with dedicated skill defined at TIM). At TSM level, the models must provide sufficient details to allow developing or buying suitable software applications, hardware components, recruiting human operators / managers or establishing internal training plans, buying and realizing machine devices, for supporting and delivering services in interaction with customers. For instance for IT applications, a TSM model enhances a TIM model with technological details and implementation constructs that are available in a specific implementation platform, including middleware, operating systems and programming languages (e.g. Java, C++, EJB, CORBA, XML, Web Services, etc.). Based on the technical specifications given at TSM level, the next step consists in the realization and the implementation of the designed service - system in terms of IT components (applications and services), physical means (machine or device components or material handling) and human resources and organization ensuring human related tasks/operations.

4.2. Enterprise interoperability: framework, maturity model and model driven approach

The current economic situation, favouring the networking of the economic players for the benefit of a collective project, leads enterprise modelling to be interested in the networks of technico-economic activities the size of which overtakes the statutory borders of the companies involved. In this sense, enterprise modelling plays a major role in the analysis of the complex transactions networks appearing within the theme of the extended enterprise.

Furthermore, as the evolution of the economic environment in which companies have to evolve leads to envisage permanently new organizations (decentralization, transformation of big size structures into networks of communicating companies, etc.), the proposed approaches must be multi-purpose to adapt itself to these various contexts and especially not be dedicated to a type of specific organizations. This situation leads to focus more on enterprise interoperability rather than enterprise integration. That is why interoperability raised up as an important challenge and a scientific domain since the beginning of this millennium.

There are many initiatives to develop solutions for enabling enterprise interoperability. This ranges from conceptual approaches focusing on frameworks, models and methodologies to practical experimentation with tools dealing with IT interoperability platforms, semantic interoperability, utility tools and models interoperability etc. A quite complete state-of-the-art on those approaches and initiatives have been developed under the three main European projects addressing enterprise interoperability: IDEAS (IDEAS 2002), ATHENA (ATHENA 2003) and INTEROP NoE (INTEROP 2003). In this section, focus is given to frameworks, maturity models and model driven approach.

4.2.1. Enterprise Interoperability Framework

Concerning the Enterprise Interoperability Framework development (state-of-the-art on existing works on interoperability frameworks reported by Chen *et al.* 2008 and Chen 2013), initial works such as the IDEAS Interoperability framework (IDEAS 2002), ATHENA interoperability framework (ATHENA 2003), European Interoperability Framework (EIF 2004) provide interesting structuration of various issues and concerns on enterprise interoperability. However they do not address explicitly interoperability problems, and do not allow structuring interoperability solutions in relation with the problems. Framework for Enterprise Interoperability (FEI), initially drafted under the frame of INTEROP NoE (INTEROP 2006) and then accepted as an international standard, is built on the existing ones.

The Framework for Enterprise Interoperability (FEI) has been specified in CEN/ISO 11354-1 (CEN/ISO 2009, Figure 8). It aims at structuring basic enterprise interoperability concepts and issues. The framework has three basic dimensions: *Interoperability concerns* that defines the content of interoperation that may take place at various levels of the enterprise (data, service, process, business), *Interoperability barriers* that identifies various obstacles to interoperability in three categories (conceptual, technological, organisational, as in GIM, Figure 1) and *Interoperability approaches* that represents the different ways to remove barriers (integrated, unified, and federated).

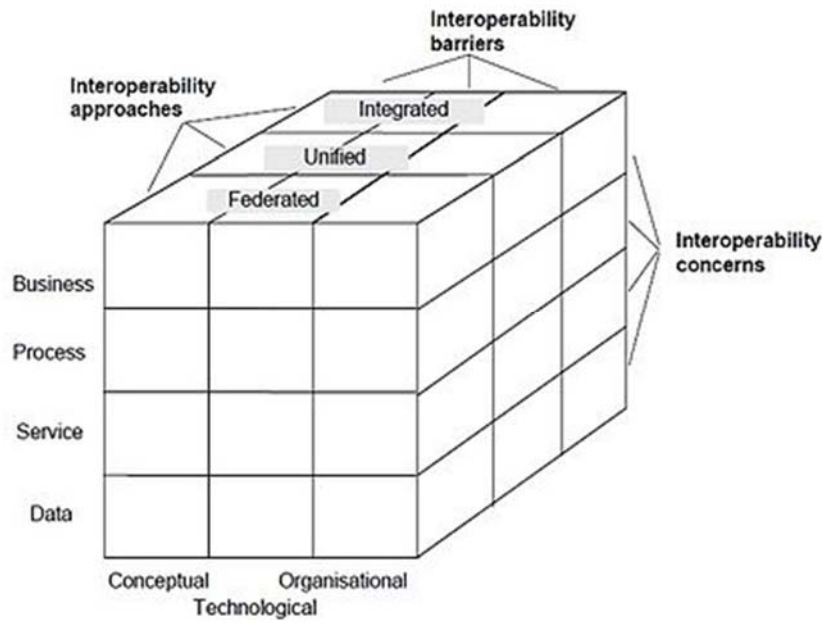


Figure 8. The Framework for Enterprise Interoperability

4.2.2. Maturity Model for Enterprise Interoperability

Concerning the development of enterprise interoperability maturity model, there exist several approaches (Guedria *et al.* 2008, 2009). One can mention some earlier works such as LISI (Levels of Information Systems Interoperability) (C4ISR 1998), (Kasunic 2004), OIM (Organizational Interoperability Model) (Clark and Jones 1999), OIAM (Organisation Interoperability Agility Model) (Kingston *et al.* 2004), LCIM (Levels of Conceptual Interoperability Model) (Tolk and Muguira 2003), EIMM (Enterprise Interoperability Maturity Model) (Athena 2003). Existing maturity models focus, in most cases, on one simple facet of interoperability (data, technology, conceptual, enterprise modelling, etc.). They are complementary rather than contradictory. Consequently, it is necessary to structure the different approaches into one single interoperability maturity model to avoid redundancy and ensure consistency.

The Maturity Model for Enterprise Interoperability (MMEI, Chen 2013) is based on the standard framework (FEI) and inspired from and influenced by some existing maturity models as mentioned in previous paragraph. MMEI has been specified in CEN/ISO 11354-2 (CEN/ISO 2011). It aims at enabling an assessment of an enterprise capability to interoperate with another enterprise. MMEI is built on the FEI described above and covers the whole problem space of the FEI (four interoperability concerns and three kinds of interoperability barriers). Five levels of interoperability maturity are defined as shown in Table 1.

Each level identifies a certain degree of capability to interoperate with another enterprise. The table also gives a high level view of MMEI and shows the main focus for each combination of each maturity level and for each interoperability barrier category. These interoperability barriers are already defined in the framework (FEI) by its interoperability barrier dimension (conceptual, technological, organisational).

Table 1. Maturity Model for Enterprise Interoperability (Chen 2013)

Maturity Levels/ Barriers	Conceptual	Technological	Organizational
Level 4 - Adaptive	Accommodated	Reconfigurable	Agile
Level 3 - Organized	Mapped	Open-architecture	Trained
Level 2 - Aligned	Adhered	Arranged	Flexible
Level 1 - Defined	Modelled	Connectable	Specified
Level 0 - Unprepared	Incomplete	Inaccessible	Inexplicit

4.2.3. Model Driven Interoperability

The Model Driven Interoperability (MDI) methodology was elaborated in the framework of the Task Group 2 (TG2) of INTEROP-NoE (Bourey *et al.* 2007) which proposed an approach, inspired from OMG MDA (see above). The goal is to tackle the interoperability problem at each abstraction level defined in MDA and to use model transformation techniques to link vertically the different levels of abstraction or horizontally to ensure interoperability of models at each level (Figure 9).

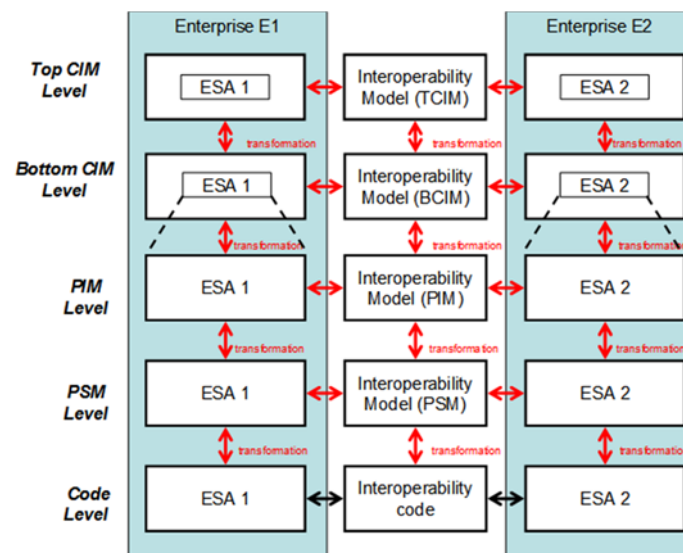


Figure 9. The Model Driven Interoperability framework

The main goal of this methodology is to allow a complete follow-up from expressing requirements to solution coding and also a greater flexibility thanks to the transformations automation. Yet, MDI was focused on pure IT systems in conjunction with SOA.

An originality of this MDI framework is also to decompose the CIM level into two levels:

- The TCIM (or TOP CIM) level aims at describing the enterprise from a business perspective using business models and classical enterprise modelling languages, and
- The BCIM (or BOTTOM CIM) level aims at extracting from the TCIM models the part of the processes only dedicated to IT system.

The interest of this decomposition is to focus the transformation between CIM and PIM only

on the IT part of the company, MDI being focused on this part. However, the main problem is to determine real criteria to define exactly the items that must be kept in the BCIM models.

4.3. Model transformation

As mentioned previously (chapter 4.1.2), transformation mechanisms have been developed between Extended Actigrams* (EA*) at the BSM level and BPMN 2.0 at TIM level. Horizontal transformation mechanisms have also been developed between BPMN 2.0 and DEVS still at TIM level (Zacharewicz *et al.* 2016) which are not presented hereafter. The transformation of an EA* into a BPMN target model has been based on meta-modeling. Figure 10 is a particularization of the meta-model approach to the context of transformation of EA* models into BPMN2.0 models.

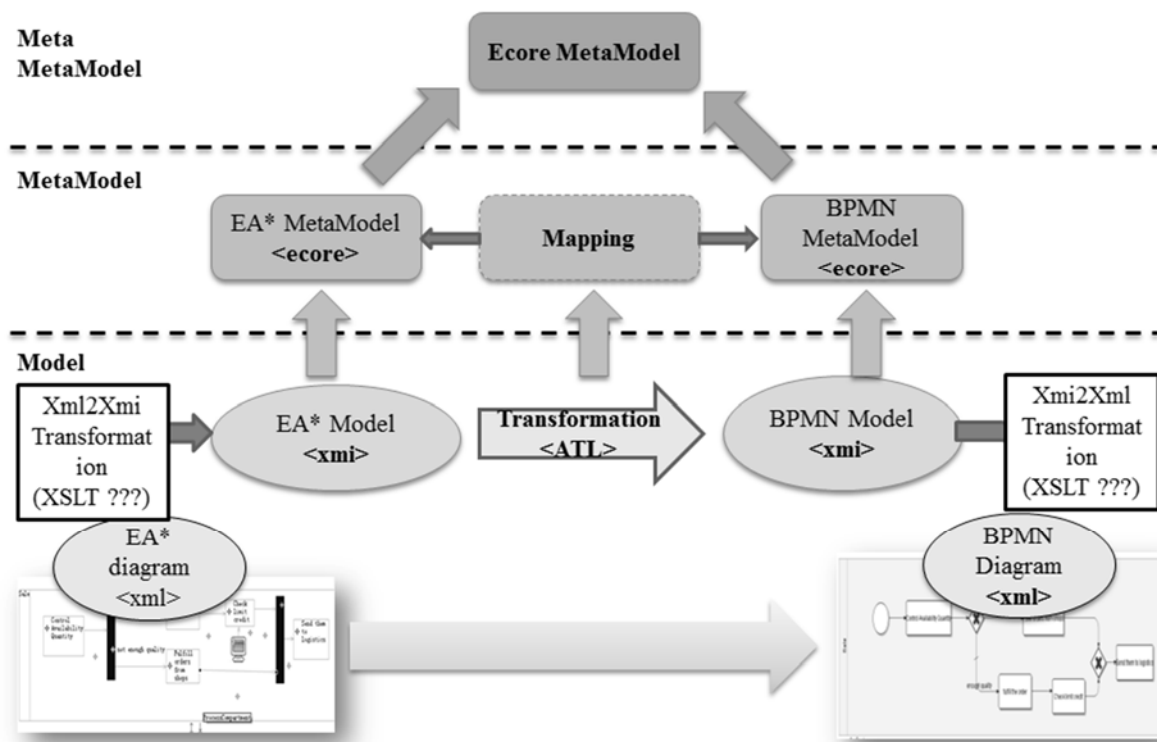


Figure 10. Transformation architecture of EA* into BPMN

The first step consists in the identification of the source and target meta-models (respectively Extended Actigram Star and BPMN2.0 meta-models), and the languages used for the model representation. The ecore meta-model is used as a meta-meta-model which defines the structures of the Extended Actigram star and BPMN meta-models. Both EA* and BPMN2.0 meta-models are defined in ecore. In addition XML Meta-data Interchange (XMI) is used to save source and target models.

After defining the meta-models and the description languages, a mapping between meta-models is built. This mapping can be characterized by a table defining constructs matching. Later on, the mapping rules described in the table will be implemented using Atlas Transformation Language (ATL).

The Eclipse “BPMN modeler” plugin is used to visualize and validate BPMN target models. It requires a specific xml format with graphical elements definition. To address this issue, we rely

on XSLT transformation sheet which will transform the BPMN models (XMI format) resulting from the ATL transformation into xml format that conforms to the BPMN modeler requirements.

4.3.1. Mapping of concepts

The mapping of concepts proposed for the transformation creates correspondences and links between concepts and their relations from EA* to BPMN language. It is a translation of constructs and their relations from one meta-model to another one. As a result, deep analysis and understanding of the EA* and BPMN meta-models, represent the main issue to start the translation and drawing of the links. The mapping is accompanied with conditions to manage the creation of relations between concepts. The model transformation was developed in this case using ATL.

4.3.2. Case Study

The concept of electronic marketplace (e-marketplace) has been based on grouping new and/or used products coming from several sellers on a unique Internet platform and under the same e-commerce catalog. The result is a trading platform as the e-market itself does not sell nor buy directly physical products or services traded on the platform. On one hand e-marketplaces and supplier directories are B2B Internet platforms. On the other hand, e-marketplaces offer a portal for B2C interface. Several enterprises took this opportunity to extend their offer. In France, major e-selling platforms have initiated e-commerce configurations since 2010 and are moving progressively to integrate this service offer. This selling concept fits the service orientation. It proposes two service interfaces, one dedicated to sellers and the other one to customers. Dedicated services can be provided. On the B2B side, the broker can choose or remove new merchants for his platform based on existing performances on the platform as visibility, customer satisfaction, low cost delivery, etc. A solution of export product catalogue is embarked on each marketplace for a win/win strategy between merchant and broker. The merchant can reuse its catalogue and be part of several marketplaces.

In this context, the case study presents one of the processes run within an e-marketplace collaborative network. It details the process of purchasing products using a marketplace website. The marketplace is maintained by a broker agent that offers services to customers who choose, configure, and buy their products online. On the other hand, sellers are targeting customers and selling their products via the broker. As a result a collaborative network, formed of the broker website, sellers, and delivery companies, offers a service to online customers. The goal of this section is to demonstrate the support provided by the tool described previously. The business model is assisted in its transformation for generating the service platform that will be implemented.

Figure 11 is a transformation example of a private sale e-marketplace purchase process model. It formalizes the business considerations captured with EA* language, then, in order to prepare the definition of the electronic platform, it transforms the model to BPMN 2.0 diagram. In detail, in this model the customer log in with his user account, browses available brands, chooses a brand, browses brand available product, and configures his product (colour, size, etc.). When the customer terminates the configuration, the broker agent verifies product's availability and delivery details from the seller company. These details will be transferred to the customer, who will decide to validate his basket or not. Then he can either pay for his products or go back to choose other products.

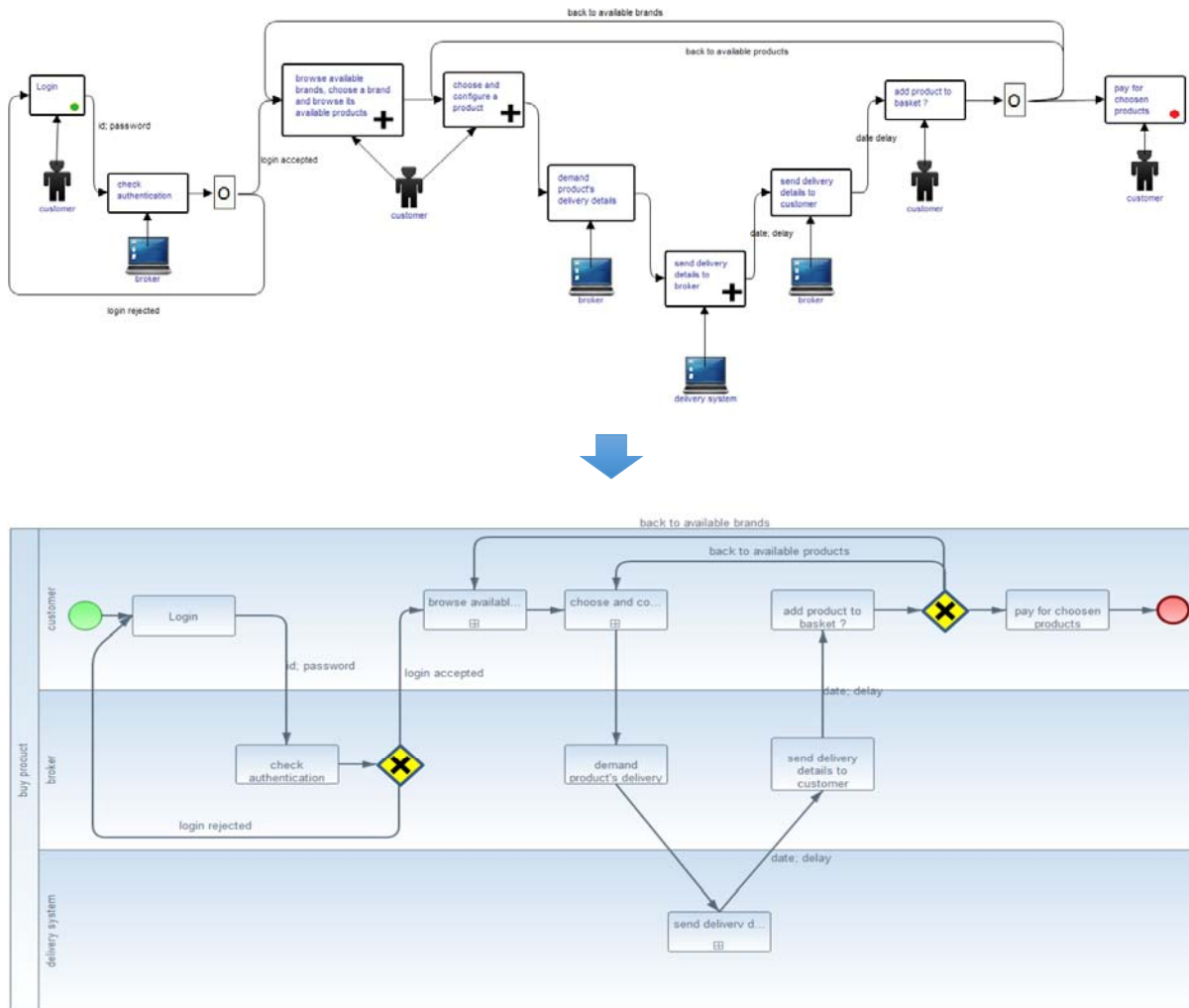


Figure 11. Example of model transformation from EA* to BPMN (Bazoun 2015)

The figure shows that the tool has been able to identify the partners and to isolate the service they solicit or generate. In the BPMN model, the lanes represent on the upper side the B2C link and service required. On the lower side the collaborative network within a B2B relation where a competition is done between sellers to provide better proposition to customer demand (eg. about the delivery time and price). The fact that the electronic service is not ready at this step needs to be taken into account.

4.4. Simulation

The question regarding models and then implementation of systems and solutions is related to the time required to validate the approaches and to obtain an ROI on the real system. This time could be too long considering the project duration. For this reason, traditional enterprise specifications based on conceptual modelling such as enterprise modelling benefits more and more from simulation techniques in helping to extend the models and overpassing design flaws, by adding dynamic dimension. It helps learning system behaviour, providing training and correcting potential problems before a final implementation of the real system. It is becoming a standard practice for building complex systems.

Simulation is still mostly used for running models of enterprise processes or anticipating decision level forecast. These categories of simulation are different by nature targeting different

audience. They are not used in combination or coupled at execution time. Nevertheless, the current demand within the context of industry 4.0 reports that coupling of real systems and virtual parts (with IoT, virtual reality, CPS) pushes enterprises using more embedded, interoperable and loosely coupled simulations and organizational models to anticipate the correct behaviour of the whole system. To make these benefits happen, modelling and simulation have to be aligned in the enterprise systems development lifecycle in a model driven approach (vertical alignment) and interoperable with other simulations (horizontal alignment). For instance, efforts have been done in proposing simulation as an integrated step in the MDSEA cycle (Zacharewicz *et al.* 2016). In this approach, the model transformation concerns the transformation of BPMN 2.0 models at the TIM or TSM level into DEVS Models directly usable for simulation. The originality of this transformation coupled with MDSEA is to reuse the existing models instead of developing new simulation models from scratch and then to verify the hypothesis of BSM models in terms of performance improvement.

At the technical level (such as MDSEA TSM), in the current context of open-source software development, service-oriented architectures (SOAs), HPC and Model Driven Approaches, modelling and distributed (cloud) simulation will assume even more significance in assessing business process management effectiveness, alignment between business process models and configuring future enterprise information system services (Wang *et al.* 2017). Alignment for interoperability between the client and vendor business strategies in a Supply Chain ecosystem remains critical too. These considerations are identified by MDI.

Despite these efforts, modelling and simulation are still mostly used as offline tools in isolation from each other. Also the decision level is not connected gathering at run time with process information or either the operative level is also not fully able to feed with live inputs and to receive live outputs from simulation. In the future, the current advances can already promise that coupling between real and virtual at run time will be enhanced thanks distributed simulation and IoT. Then the most important challenge concerns the integration of the human in the simulation loop. The human can be “augmented” thanks to the results of simulation but also viewed as an actor of the simulation process to be trained. No doubt that the model driven approaches will facilitate to close the loop from the simulated part to the real world.

5. CONCLUSION AND FUTURE WORK

During the last thirty years, enterprise modelling has been recognised as an efficient tool to externalise the knowledge of the company in order to understand its operations, to analyse its running and to design new systems from several points of view: functions, processes, decisions, resources, information technology. This paper has shown that enterprise modelling methods evolved all along this period, based on theoretical concerns but also on practitioner matters. After a phase of divergence with many developments in parallel and then attempts of reconciliation (from modelling frameworks towards UEML), this is now the phase of usage when enterprise modelling is less and less used as a stand-alone approach but rather as a component of larger engineering methods (such as MDA and enterprise engineering). Several model driven approaches have been defined and experimented from business modelling through simulation in order to get an efficient usage of these methods. Then the coupling of enterprise modelling with simulation is a very promising approach. In this way, because the system is not only considered during its design phase but also during its exploitation, enterprise modelling does not stay as an approach usable only during the design time but moves closer to runtime. This trend is fully consistent with the industry 4.0 objectives. In this sense, enterprise models

move to be used during the total enterprise life cycle. This trend provides a very exciting scientific challenge for the next future.

In this paper, we tried to show that many current challenges address the models transformation:

- Translation of a model at a given abstraction level into a model at a lower level within the framework of systems design (vertical transformation, MDA),
- Translation of static models in simulated models,
- Translation of models from a language to another one (horizontal transformation),
- Establishment of the coherence of models related to different semantics (interoperability).

So, we consider that the main challenge today concerns the models transformation and, more precisely, the development of methods of transformation that allow a fluid and fast process with verifiable results.

The models transformation is based on a meta-modelling activity. So, behind the problem of the models transformation, and in a more theoretical perspective, the challenge concerns the development of meta-modelling methods which would allow to meta-model a language in a systematic way. It is to be noted that these methods do not exist today and that the meta-modelling activity is a work burdensome, specific to each situation and with an uncertain result.

This domain constitutes a real domain of research for the future years.

Bibliography

- Abrial J. R. 1974. Data Semantics in Data Base management, Klimbie, Koffeman (eds), North Holland, p. 1-57.
- AMICE. 1993. *CIMOSA: Open Architecture for CIM*. Berlin: Springer.
- ATHENA. 2003. Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications. FP6-2002-IST-1, Integrated Project.
- Bazoun, H. 2015. Contribution to a methodology for service systems modeling and engineering through a model driven approach: architecture, transformation and model simulation. PhD thesis, university of Bordeaux.
- Bouaza, M. 1995. *La norme STEP*. Paris: Hermès.
- Bourey, J.P., R. Grangel, G. Doumeingts, and A. Berre. 2007. Deliverable DTG2.3 from the INTEROP project. *Report on Model Driven Interoperability*. Available from: <http://interop-vlab.eu> (accessed May 2014)
- Braesch, C., A. Haurat, and J. M. Beving. 1995. L'entreprise-système. in *La modélisation systémique en entreprise*, Paris: Hermès.
- C4ISR. 1998. Levels of Information Systems Interoperability (LISI). Architecture Working Group (AWG).
- CEN/ISO. 2009. Part 1: Framework for enterprise interoperability, Draft International Standard ISO/DIS 11354-1.
- CEN/ISO. 2011. Part 2: Maturity model for assessing enterprise interoperability, CEN/ISO 11354-2, Draft for proposed activation of WI 00310094, document N270.
- Chapurlat, V., M. Larnac, E. Lamine, and J. Magnier. 1999. Definition of a formal analysis framework for existing enterprise modelling approaches. Proceedings of the *Annual conf. of ICIMS-NOE, Life cycle approaches to production systems: management, control, supervision (ASI)*, Leuven, Belgium, September 22-24.
- Chen, D., and F. Vernadat. 2001. Standardisation on Enterprise Modelling and Integration:

- Achievements, on-going Works and Future Perspectives. Proceedings of the 10^{ème} *symp. Information Control in Manufacturing (INCOM)*, Vienna, Austria, September 20-22.
- Chen, D., B. Vallespir, and G. Doumeingts. 1997. GRAI integrated methodology and its mapping onto generic enterprise reference architecture and methodology. *Computers in industry* 33(2): 387–394.
- Chen, D., B. Vallespir and G. Doumeingts. 2002. Designing Manufacturing Systems: Contribution to the Development of an Enterprise Engineering Methodology (EEM) within the Frame of GERAM. Proceedings of the *IFAC world congress*, Barcelona, Spain, July 21-26.
- Chen, D., B. Vallespir, and G. Doumeingts. 2002. Developing an unified enterprise modelling language (UEML) – Roadmap and requirements. Proceedings of the *3rd IFIP Working conf. on infrastructures for virtual enterprise (PROVE)*, Sesimbra, Portugal, May 1-3.
- Chen, D. 2007. Enterprise-control system integration - an international standard. *International journal of production research* 43(20): 4335-4357.
- Chen, D., G. Doumeingts, and F. Vernadat. 2008. Architectures for Enterprise Integration and Interoperability: Past, Present and Future. *Computers In Industry* 59(7): 647-659.
- Chen, D. 2013, Framework for Enterprise Interoperability and Maturity Model (CEN/ISO 11354). Interoperability for Enterprise Software and Applications, H. Panetto and N. Boudjilda eds., John Wiley and Sons Inc.
- Clark, T., and R. Jones. 1999. Organisational interoperability maturity model for C2. Proceedings of the *Command and Control Research and Technology Symposium*, Washington, US.
- Di Leva, A., F. Vernadat, and D. Bizier. 1987. Information system analysis and conceptual database design in production environments with M*. *Computers in Industry* 9(3): 183-217.
- Doumeingts, G., B. Vallespir, M. Zanettin, and D. Chen. 1992. GIM, GRAI Integrated Methodology - A Methodology for Designing CIM Systems. Version 1.0, LAP/GRAI, Université Bordeaux 1.
- Doumeingts, G., B. Vallespir, and D. Chen. 1995. Methodologies for designing CIM systems: A survey. *Computers in Industry* 25(3): 263–280.
- EIF. 2004. European Interoperability Framework. CompTIA: European Industry Association, white paper - ICT Industry Recommendations, <http://www.comptia.org>.
- GERAM. 1999. GERAM: Generalised Enterprise Reference Architecture and Methodology. Version 1.6.1, IFIP–IFAC Task Force on Architectures for Enterprise Integration, march.
- GRPGT5. 1999. La modélisation d'Entreprise : le Point de Vue Productique. Working group #5 «Enterprise Modelling» of the Groupement pour la Recherche en Productique, December.
- Guedria W., Y. Naudet, and D. Chen. 2008. Interoperability Maturity Models - Survey and Comparison. Proceedings of the *IFAC/IFIP workshop Enterprise Integration, Interoperability and Networking (EI2N)*, Monterrey, Mexico, November 12-13.
- Guedria W., D. Chen and Y. Naudet. 2009. A Maturity Model for Enterprise Interoperability, Proceedings of the *IFAC/IFIP workshop Enterprise Integration, Interoperability and Networking (EI2N)*, Vilamoura, Portugal, November 3-4.
- Hay D. 1997. The Zachman Framework: An Introduction, The Data Administration Newsletter.
- IDEAS. 2002. Roadmaps, Annex 1 – Description of Work. IDEAS Consortium: Interoperability Development for Enterprise Application and Software, Thematic Network.
- INTEROP. 2003. Proposal Part B. Interoperability Research for Networked Enterprises Applications and Software, Network of Excellence.
- INTEROP. 2006. Enterprise Interoperability – Framework and knowledge corpus. INTEROP DI Deliverable DI 1.2, December.
- ISO. 2006. Enterprise integration - Framework for enterprise modelling, ISO 19439:2006. ISO/TC 184/SC 5.

- Kamsu-Foguem B. and V. Chapurlat. 2006. Requirements modelling and formal analysis using graph operations. *International Journal of Production Research* 44(17): 3451-3470.
- Kasunic M., and W. Anderson. 2004. Measuring systems interoperability: challenges and opportunities. Software Engineering Measurement and Analysis Initiative, Technical note, cmu/sei-2004-tn-003, Carnegie Mellon University, US, April.
- Kosanke, K., F. Vernadat, and M. Zelm. 1999. CIMOSA: enterprise engineering and integration. *Computers in Industry* 40(2-3): 83-97.
- Lillehagen, F., and J. Krogstie. 2002. Active Knowledge Models and Enterprise Knowledge Management, in *Enterprise Inter- and intra-organizational integration*, Kluwer Academic Publishers, Kosanke K. et al. (ed.) .
- Menzel, C.P., and R.J. Mayer. 1998. The IDEF Family of Languages. in *Handbook on architectures of information systems*, P. Bernus, K. Mertins, and G. Schmidt ed., Berlin: Springer.
- Mertins, K., and F.W. Jaekel. 1998. MO2GO: User Oriented Enterprise Models for Organisational and IT solutions. in *Handbook on architectures of information systems*, P. Bernus, K. Mertins, and G. Schmidt ed., Berlin: Springer.
- MSEE. 2012. Annex 1 description of work. Manufacturing Service Ecosystem Project, FoF-ICT-2011.7.3, July 29th, available at: <http://msee-ip.eu/> (accessed May 2014).
- Neaga E.I. and J.A. Harding. 2005. An enterprise modeling and integration framework based on knowledge discovery and data mining. *International Journal of Production Research* 43(6): 1089-1108.
- OMG. 2005. MDA Guide Version 1.0. <http://www.omg.org/cgi-bin/doc?omg/03-06-01>
- Panetto, H., F. Mayer, and P. Lhoste. 2001. Unified Modelling Language for meta-modelling: towards Constructs definition. Proceedings of the 10th symp. *Information Control in Manufacturing (INCOM)*, Vienna, Austria, September 20-22th.
- Petit, M., J. Goossenaerts, M. Gruninger, J.G. Nell, and F. Vernadat. 1997. Formal Semantics of Enterprise Models, Proceedings of the *Int. Conf. on Enterprise Integration and Modelling Technology (ICEIMT)*, Turin, Italy, October 28-30, K. Kosanke and J. Nell ed., Berlin: Springer.
- PTA. 1992. Norme expérimentale Base-PTA. CCGA, ISMCM/CEST, experimental standard Z 68-901, september.
- Salvato, G., I.J. Leontaritis, P. Winstone, M. Zelm, D. Rivers-Moore, and D. Salvato. 1999. Presentation and Exchange of Business Models With CIMOSA-XML. *Computers in Industry* 40(2-3): 125-139.
- Scheer, A.W., and M. Nüttgens. 2000. ARIS architecture and reference models for business process management. in *Business Process Management*, Berlin: Springer.
- Scheer, A.W. 1994. Business process engineering: reference models for industrial enterprises. Berlin: Springer.
- Scheer, A.W., and K. Schneider. 1998. ARIS - Architecture of Integrated Information Systems. in *Handbook on architectures of information systems*, P. Bernus, K. Mertins, and G. Schmidt ed., Berlin: Springer.
- Scheer, A. W., 2000. Business Process Frameworks. Berlin: Springer.
- Shorter, D. 2000. Enterprise Engineering and Integration - The Standardization Scene. CEN TC310/WG1, April.
- Tardieu, H., A. Rochfeld, and R. Colletti. 1983. La méthode MERISE, Principes et outils. Paris: Les éditions d'organisation.
- Tolk, A., and J.A. Muguira. 2003. The levels of conceptual interoperability model. In proceedings of the *2003 Fall Simulation Interoperability Workshop*, Orlando, FL, USA, September 14-19.
- Vallespir, B., and G. Doumeingts. 2007. The GRAI Method Part 1: global modelling. Lecture

- presentation, Interop Network of Excellence.
- Vallespir, B., and G. Doumeingts. 2007. The GRAI method Part 2: detailed modelling and methodological issues. Lecture presentation, Interop Network of Excellence.
- Vallespir, B., G. Doumeingts, and D. Chen. 2001. Problems and Research orientation for UEML: A point of view. Oral presentation at *IFAC-IFIP Task force Interest group on UEML*, Vienna, Austria, September 19.
- Vallespir, B., C. Braesch, V. Chapurlat, and D. Crestani. 2003. L'intégration en modélisation d'entreprise : les chemins d'UEML. Proceedings of the 4th *Conférence Francophone de Modélisation et Simulation (MOSIM)*, Toulouse, France, April 23-25.
- Vallespir, B. 2003. Modélisation d'entreprise et architecture de conduite des systèmes de production. Habilitation à Diriger des Recherches dissertation, Université Bordeaux 1. December 19th.
- Vernadat, F. 1998. The CIMOSA Languages. in *Handbook on architectures of information systems*, P. Bernus, K. Mertins, and G. Schmidt ed., Berlin: Springer.
- Vernadat, F. 1999. Techniques de Modélisation en Entreprise : Application aux Processus Opérationnels. Paris, Economica, Collection Gestion.
- Vernadat, F. 1999. Unified Enterprise Modelling Language (UEML). Oral presentation at *IFAC-IFIP Task force Interest group on UEML*, Paris, France, December 16th.
- Vernadat, F. 2001. UEML: Towards a Unified Enterprise Modelling Language. Proceedings of the 3rd *Conférence Francophone de Modélisation et Simulation (MOSIM)*, Troyes, France, April 25-27.
- Vernadat, F. 2002. UEML: Towards a Unified Enterprise Modelling Language. *International Journal of Production Research* 40(17): 4309-4321.
- Wang, Y., G. Zacharewicz, M.K. Traore, and D. Chen. 2017. Use of Fuzzy Clustering for Discrete Event Simulation Model Construction. Proceedings of the *IFAC World Congress*, Toulouse, France, July 9-14.
- Williams, T.J. 1996. An Overview of PERA and the Purdue Methodology. in *Architectures for Enterprise Integration*, P. Bernus, L. Nemes, and T.J. Williams ed., London: Chapman et Hall.
- Zacharewicz, G., H. Bazoun, J. Ribault, Y. Ducq, and H. Boyer. 2016. SLMToolBox: enterprise service process modelling and simulation by coupling DEVS and services workflow, *Indersciences International Journal of Simulation and Process Modeling* 11(6): 453-467.
- Zachman J.A. 2008. The Concise Definition of The Zachman Framework, Zachman International, Inc.
- Zelm, M. 1995. CIMOSA: a primer on key concepts, purpose and business value. Technical report, CIMOSA association, Stuttgart, Germany.
- Zelm, M. 2001. Enterprise modelling - constructs comparison (state of the art). Technical report, CIMOSA association, Stuttgart, Germany.