Towards a Product Oriented Process Modelling for Enterprise Applications Synchronisation and Interoperability

Salah Baïna, Hervé Panetto, Gérard Morel

HAL Id: hal-00022723
https://hal.science/hal-00022723

Submitted on 13 Apr 2006

Towards a Product Oriented Process Modelling for Enterprise Applications Synchronisation and Interoperability

Salah Baïna, Hervé Panetto, Gérard Morel

CRAN (UMR 7039), University Henry Poincaré Nancy I
F 54506 Vandoeuvre les Nancy, France
salah.baina@cran.uhp-nancy.fr, herve.panetto@cran.uhp-nancy.fr, *gerard.morel@cran.uhp-nancy.fr*

Abstract: Throughout product lifecycle, coordination needs to be established between reality in the physical world (physical view) and the virtual world handled by manufacturing information systems (informational view). This paper presents the "Holon" modelling concept as a means for the synchronisation of both physical view and informational views. Then, we show how the concept of holon can play a major role in ensuring interoperability in the enterprise applications context.

1. INTRODUCTION

Major interoperability solutions in the enterprise handle mainly the interconnection between enterprise activities and processes. However, in manufacturing enterprises the product is becoming more and more important. Indeed, the product has become an active element in taking decisions, coordinating manufacturing processes and establishing interaction with the ambient system (Information systems, shop-floor applications, other products) (Fig. 1). Actually, throughout product lifecycle, coordination needs to be established between product reality in the physical world (physical view) and the virtual world handled by manufacturing information systems (informational view).
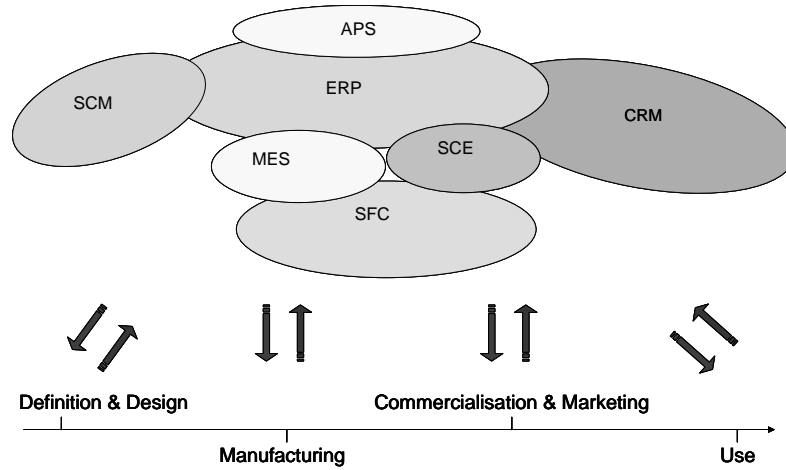
Fig. 1: Coordination between the product and enterprise applications and systems

Due to its increasing role in the enterprise, applications possess at least a restricted view of the product. Hence, it is mandatory to define an approach to maintain the coherence between all those views, when they describe a common shared object, this approach should also respect the coherence of those views and the actual state of the product during its life cycle (design, fabrication, manufacturing, marketing and merchandising, use…).

In this paper, we define a holon-based approach in order to synchronise views in the business world and in the manufacturing world using the holon concept. The paper continues by presenting the usability of the concept of holon in ensuring interoperability in the enterprise context.

Section 2 presents the bases of our holonic process modelling concepts that use the product as a centric entity in process models. Section 3 of the paper gives a brief introduction to interoperability in the enterprise and how holon can be used as a means for enterprise applications interoperability. In Section 4, an implementation of the holon is proposed. Section 5 gives conclusions and perspectives for this work.

2. A modelling concept for product representation

In this section, we introduce the holon as a modelling concept. Afterwards, we will show how this concept can be exploited in order to facilitate taking into account interoperability concerns in the modelling phase.

In the manufacturing context, a Holonic Manufacturing System (HMS) is an autonomous and co-operative building block of a system for transforming, transporting, storing and/or validating information and physical objects [1]. In this paper, we adapt the *holon* concept definition to solve the problem of synchronisation between physical views and informational views of the same objects. We define the holon then as an aggregation of an information part and a physical part.

In Holonic Process Modelling ([2][3]), holons represent products; the physical part of the holon represents the material part (also called physical view) of the product and the informational part of the holon represents the informational part (informational view) of the product. We consider that the information about a holon is distinguished into two categories:

- Attributes describing the current state of the holon. The state of a holon contains three kinds of attributes: space attributes, shape attributes, and time attributes [4];
- Properties related to the holon but which do not correspond to any of the three types of properties; space, shape or time (for example, a product's reference).

Holons can be classified into two categories; *(i)* elementary holons and *(ii)* complex holons:

- Elementary holons (holons) are the combination of a single informational part and a single physical part.
- Complex holons are the result of the processing and treatment of one or more other holons, this processing can be a transformation of one holon to obtain two or more new ones, or assembling a set of holons in order to compose a new one. Each complex holon can be defined as the output of the execution of a manufacturing process on one or more holons.

Next section will try to formalise the *holon* concept using the well-known BWW model ([5], [6]). Afterwards, we will propose the *holon* meta-model which is the core of our approach.

2.1. The holon concept formalisation using BWW ontology:

The approach presented in this paper is based on the BWW model (Bunge, Wand and Weber) which is inspired of the original metaphysical theory developed by Mario Bunge in [7], [8]. Bunge's Treatise on Basic Philosophy devotes two volumes to ontology, in which he articulates a set of high-level abstract constructs that are intended to be a means of representing all real-world phenomena. In this paper, we focus mainly on the notions of thing, kind, state, property and attribute. Interested readers may refer to [7], [8], for more details. In Bunge's ontology, there are two kinds of objects: *concrete things* or simply *things*, and *conceptual objects* or *constructs (composite things)*. All objects have properties. If the objects are conceptual, their properties are called *attributes*. If the objects are things, their properties are called *substantial properties* (or simply *properties*). A substantial property is a feature that some things possess even if we are ignorant of this fact. An attribute is a feature assigned to some object in order to manipulate them in some specific models. An attribute may or may not represent a substantial property. Every property is possessed by some thing. Some properties are inherent properties of things (called *intrinsic properties*), other properties are properties of pairs or, in general, n-tuples of things (termed *mutual properties*).

An arbitrary collection of things do not need sharing a given set of properties. When they do, however, and no thing outside the collection has the properties of interest, the collection is called a *kind*.

The properties of a functional schema of a thing (representing value properties of the thing) are also called *state variables* because their values contribute to characterising or identifying the *states* the thing of interest can be in. An ontological hypothesis is that, every thing is at a given time in a state.

In the following, we establish the matching between the holonic meta-model and the BWW ontology (see Table 1).

Table 1: Matching between holon concepts and BWW concepts.

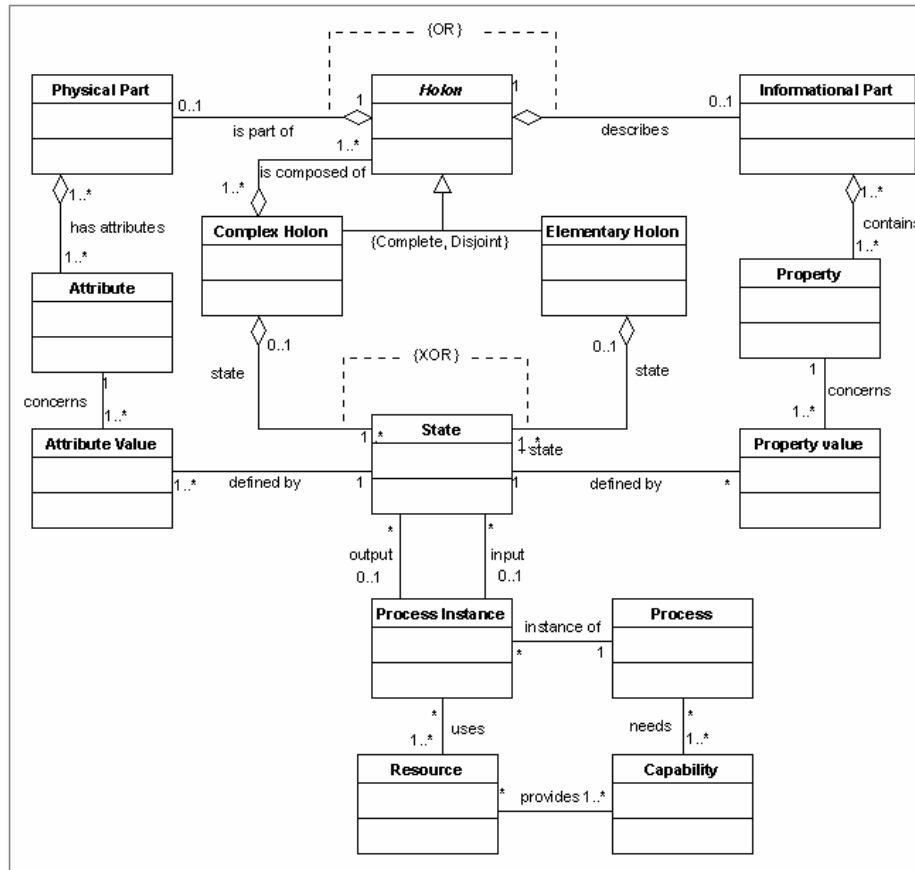| Holonic	BWW
Holon	Thing
Class of holons	Class
Holonic Attribute	Intrinsic Property
Holonic Property	Mutual, Hereditary or Emergent Property
State	State

2.2. The Holon definition:

Fig. 2 represents the formalisation of holons in UML class diagram; here is a brief description of this class diagram: The Class *Holon* defines basic attributes for both complex and elementary holons. A *Physical Part* is a reference to the physical part encapsulated in a *holon*. An *Elementary Holon* is defined as a holon with no indication about his lifecycle. For example a product, produced by external manufacturing systems does not give information about the processes needed for its manufacturing. A *Complex Holon* is a holon that has been obtained by:
-   Assembling existing holons,
-   Disassembling an existing holon into new holons.

The *state* class defines the different states that have been observed during the processing phase of the holon. Every manipulation of a holon through a process (Process Instance) implies a change in the state of the processed holon. A *Property* of a *holon* contains information that can not be handled only using its *state*. The *Process instance* refers to the execution of a process on a single holon, this class enables description of the execution of the process with high level of detail (e.g.: elapsed time,

start and end of the treatment, used equipment, needed personal). A *Process instance* input is a holon state A *Process* describes an internal process that is performed inside the studied domain. The *Resource* class describes resources needed to perform a process instance. A *resource* can be a material resource, a software resource or a human resource. Each *resource* provides a set of capabilities, and each *process* needs some capabilities to be performed.
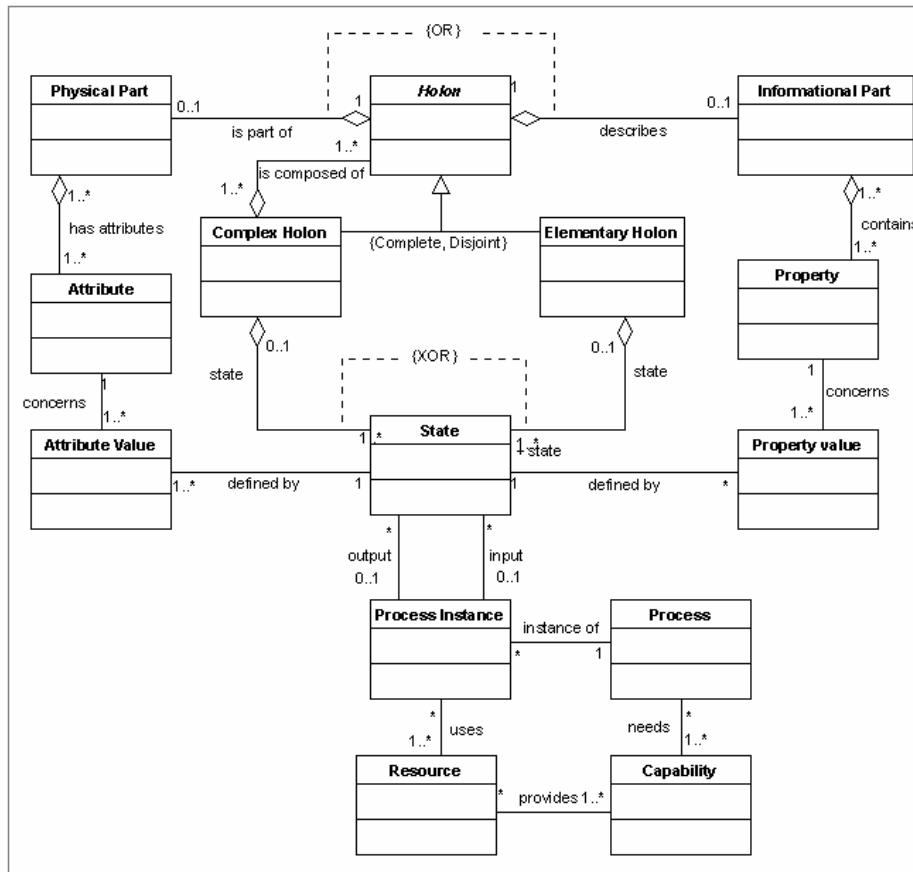
Fig. 2: Class diagram for the Holon model

For the sake of simplicity, some complex constraints have not been represented in this meta-model; formalising those constraints is out of the scope of this paper. Ongoing work handles the specification of all the constraints of this meta-model using the OCL (Object Constraint Language)..

In the next section, we will show how this approach can be used to express models based on the holon concept defined in section using models related to existing data exchange standards and other unified languages. This transformation will enable model and data exchange between applications that are based on holon models and different enterprise applications from different levels in the enterprise.

3. Holons and interoperability

The ISO/IEC 23821 Information Technology Vocabulary defines interoperability as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." The IEEE STD 610.122 standard defines interoperability as "the ability of two or more systems or components to exchange and use information". In this paper, interoperability definition is adapted from the two previous definitions as:

Definition 1: Interoperability is the ability to communicate, to cooperate and to exchange models between two or more applications despite differences in the implementation languages, the execution environments, or the models abstraction [9]

In order to take into account interoperability requirements during modelling phase in the context of manufacturing systems, we introduce, in this section, the holonic modelling approach for interoperability. Existing interoperability standards and most of existing techniques that enable business process or workflow interoperability are based on a message exchange paradigm (e.g. Wf-XML, BizTalk, FIPA ACL.). These solutions resolve only the particular case of syntactic interoperability (messages vocabulary, messages format, data types, etc). In this section, we show how the holon concept can be used as a means for resolving interoperability issues. First, we will show the use of the holon to handle horizontal interoperability concerns at modelling time. Second, the case of vertical interoperability is studied.

3.1. Holon in action for procedure driven interoperability

Major interoperability approaches are based on a remote procedure call paradigm; these approaches handle the interoperability concerns at runtime, this is called procedure driven interoperability. In this context, it is assumed that the required procedure and the provided procedure perform exactly the same functionalities. Thus, the main objective of this category of interoperability is the matching and the adaptation of input and output parameters.

In this section, we show how the use of holons concept in the modelling manufacturing environments, enable considering procedure driven interoperability concerns; this will facilitate resolving procedure driven interoperability problems during engineering phase. To Model manufacturing shop-floor, we focus mainly in a minimal set of four Entities:

- *Actor*: represents a person or a group of persons that act in someway on processes or in the information system of the enterprise. An a actor can be internal or external to the enterprise.
- *Process*: is a value chain that provides a good or a service to an internal or external customer.
- *Flow*: is a set of elements (data, information, energy, material, ...) that are exchanged between processes
- *Holon* which *represents* Products instances.

As we shown in section 2.2, a *holon* is described by *properties* and *attributes* that are mandatory for controlling the execution of processes on the *holon*. To manipulate those pieces of information we assume that each *process* is indeed composed of two interdependent sub-processes:

(i) An informational process which is responsible of manipulating, updating and controlling the information concerning the product (holon), this informational process can be implemented by an application that is performed on the information contained in the product.

(ii) A physical process that performs all physical transformations on the material of the product.

Those two sub-processes are performed in an atomic operation (both are executed or none of them). Two types of relationships between a process and a piece of information (property or attribute) have been identified: production and consumption:

- Production: we say that a process produces an attribute (or property) when that attribute did not exist before the execution of the process;
- Consumption: a process is said to be consumer of an attribute (or property) when it uses that attribute (or property) or updates it.

The specification of relationships between processes and pieces of information during modelling phase enables identifying the interfaces of processes at modelling time. The interface of a process defines its inputs and outputs.

The precedence relation can be considered between processes; this relation is defined as explained in the following.

Definition 2: The relation of precedence is a partial order between processes; we say that a process P1 precedes a process P2 (*P1 $<_{Pred}$ P2*) if it exists a path composed of flows and processes that leads from P1 to P2. In the case cyclic systems, occurrences of execution of processes should be considered; example *$P1_i <_{Pred} P2_i$* the ith execution of P1 occurs before the ith execution of P2.

Interoperability of processes can then be defined as explained in the following:

Definition 3: A process P is said interoperable with a system S (composed of processes) if and only if each input of P is declared as an output of one of his predecessors in S.

Using the holonic modelling concepts in manufacturing context to define process interfaces and their relationships with data handled by the product (holon), enables the consideration process interoperability concerns at modelling time and not during the phase of engineering.

3.2. Holon Interoperability with the MDA approach

3.2.1. Introduction to the MDA approach for interoperability

In this section, we introduce an approach for interoperability based in a model driven architecture (MDA) [10]. The main objective of this section is to show how models based on the holon concept defined in section 2 could be expressed and transformed into models based on existing data exchange standards and other unified languages.
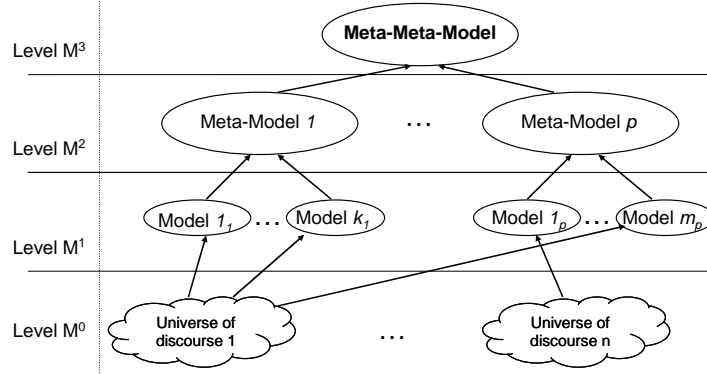

Fig. 3: The four-level ontological approach.

Fig. 3 shows the four-level ontological approach levels for modelling that are used in the MDA. As it is explained in [11], the lowest level M^0 presents different subjects for modelling, called universe of discourse. The level M^1 contains different models of each universe of discourse. The next level M^2 presents domain specific meta-models: one meta-model for each of the domains of interest relevant for the M^1 models. And finally, M^3 level presents a meta-meta-model designed to allow the definition of all the existing in the scope of the meta-models.

In case of MDA, each application is a based on a repository or model defined in the M^1 level which is based on a specific meta-model defined in M^2. Application interoperability can then be resolved either by interconnecting applications together using a level M^0 exclusive reasoning, or by establishing a top-down approach for resolving applications interoperability based on the four levels of the MDA. Several research works have been done in order to resolve meta-models mapping problems. Lemesle [12] explains how models transformation can be resolved by establishing transformation rules between meta-models. Transformation rules define a mapping that guides model transformations from the instances of the source meta-model to instances of the target meta-model.

In order to use the MDA approach for interoperability in the holonic context, we need to position the holonic modelling approach in terms of models, meta-models and universe of discourse (M^2, M^1 and M0). In the holonic context, the universe of discourse M^0 concerns "Manufacturing enterprise". To describe this universe of discourse we use holonic models (M^1) that are instantiations of the meta-model defining holons and relationships with their context (M^2).

To ensure interoperability between applications that handle those different meta-models, we should first define mappings that enable transforming one instance of a meta-model in an instance of another meta-model. Let us consider two applications A and B; A and B are interoperable, if and only if there is a mapping from the meta-model of A (M_A) to the meta-model of B (M_B) and a mapping form M_B to M_A. Those mappings ensure that we can build a model compatible with A from a model used by B (and vice versa).

3.2.2. Holons and MDA for interoperability

The universe of discourse concerned by the holon is "Manufacturing product environment", the holon meta-model is identified as a specific model for modelling this universe of discourse, the holon meta-model is an element of the M^2 level. The "Manufacturing product environment" can also be described using other modelling approaches and languages dedicated to specific domains in the enterprise; for example UEML and B2MML. UEML is the Unified Enterprise Modelling Language; it is used at the organisational level of the enterprise ([13], [14]). B2MML [15] is an implementation of the part 1 of the IEC FDIS 62264 standard [16] developed for interfacing the manufacturing control and execution systems with higher level systems. Both UEML and B2MML are elements in the M^2 level of the MDA hierarchy.

To experiment the holon models reusability in other enterprise modelling frameworks, these two previous languages have been chosen; UEML and B2MML. The next section defines mappings from the holonic meta-model to UEML and B2MML meta-models.

Mapping Holon with the Unified Enterprise Modelling Language. The Unified Enterprise Modelling Language (UEML) is the result of the UEML project [13]. The UEML is an Interlingua between Enterprise Modelling tools. The meta-model of UEML1.0 [17] defines the set of most relevant concepts and notions for Enterprise modelling.

Mapping with the B2MML language and the IEC 62264 standard. Business to Manufacturing Markup Language (B2MML) is an XML implementation of the IEC 62264 part 1. The IEC 62264 is the standard specifying the exchange of data and models interfacing the shop floor level into the enterprise. It is composed of six different parts designed for defining the models and interfaces between enterprise activities and control activities. Each model concerns a particular view of the integration problem. Those models show increasing detail level in the manufacturing system.

The detail of those mappings has been published in other papers, for further details see [3]. In the next section, an implementation of both modelling interoperability approaches is presented.

4. Implementation

To experiment the holonic approach (modelling and mappings) defined above in real case, we have implemented this approach into a commercial CASE tool named MEGA Suite[1]. MEGA Suite is an enterprise process modelling tool, which provides a business process analysis tool, modelling tools and design environments. MEGA Suite has its own meta-model that described all concepts and objects ready to use in business process models, and all relationships that exist between those concepts. This meta-model can be customized and specialised for specific users' needs. MEGA Suite can be used to define, describe and exploit several kinds of diagrams (e.g.: Business process Diagrams, UML Diagrams, Workflows, to name only a few). In our contribution, we focus only on business process diagrams; indeed they seem to be the most adequate choice for holon integration. Business Process diagrams in MEGA Suite are based on a meta-model inspired from BPMN[2]. MEGA offers tools that enable customizing the meta-model; we used these tools to embed our own holon model into the meta-model of MEGA Suite in order to test the usability of our proposal. To implement the mappings from the holonic models designed in MEGA Suite and the other formats, we chose the XML formalism to represent data extracted from our holonic models. The XML interface enables generating different XML files based on specific XML schema; each file corresponds to a specific standard or language. The B2MML and the UEML XML schemas are used to generate valid and reusable documents. Afterwards those documents can be imported in further applications compatible with UEML or B2MML formats. The translation from the XML generated directly from the holonic diagrams, to the different XML schemas is performed by a translator developed in JAVA programming language and uses XSLT rules to transform an XML input file describing a holonic model to an XML output file in UEML or B2MML format.

5. Conclusion

In this paper, we defined a modelling concept for representing products and their associated information in business process diagrams. This modelling concept enables verifying the coherence between the physical objects and their informational views. The second part of this paper proposes a product oriented interoperability approach for enterprise applications interoperability, this approach aims, on one hand, at maintaining the synchronization the physical objects and their informational views, on the other hand it aims at maintaining the coherence between all information concerning the product but owned and managed by different applications in different repositories. Finally, we briefly presented the implementation and integration of the whole approach in a CASE tool.

---

[1] MEGA Suite, MEGA International, *www.mega.com*

[2] Business Process Modelling Notation, *www.bpmn.org*

6. REFERENCES

1. Seidel, D. and Mey, M. 1994, IMS - Holonic Manufacturing Systems: Glossary of Terms, In Seidel D. and Mey M. (eds), IMS - Holonic Manufacturing Systems: Strategies Vol. 1, March, IFW, University of Hannover, Germany.
2. Morel G., Panetto H., Zaremba M.B. and Mayer F. (2003). Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues. *IFAC Annual Reviews in Control*. 27/2, 199-209, December.
3. Baïna.S, H. Panetto and G. Morel, 2005. A holonic approach for application interoperability in manufacturing systems environment. *In Proc of the 16th IFAC World Congress, Prague, July 4-8, 2005.*
4. Panetto H. and Pétin, J.F., 2005. Metamodelling of production systems process models using UML stereotypes, *International Journal of Internet and Enterprise Management*, 3/2, 155-169 - Inderscience Publisher, ISSN: 1476-1300, 2005.
5. Wand, Y. and R. Weber. *An Ontological Model of an Information System* . IEEE Transactions on Software Engineering. November, pp. 1282 -92, 1990.
6. Wand, Y., Weber, R.: On the deep structure of information systems. Information Systems Journal 3 (1995) 203---223
7. Bunge, M., 1977. *Treatise on Basic Philosophy (Volume 3), Ontology I: The Furniture of the World*, Boston: Reidel, 1977.
8. Bunge, M., 1979, *Treatise on Basic Philosophy (Volume 4), Ontology II: A World of Systems*, Boston: Reidel, 1979.
9. Kalfoglou, Y. and Schorlemmer, M., 2004. Formal Support for Representing and Automating Semantic Interoperability. In *Proceedings of 1st European Semantic Web Symposium (ESWS'04)*, pages pp. 45-61, Heraklion, Crete, Greece.
10. Mellor S.J., Kendall S., Uhl A. and Weise D. (2004). *Model Driven Architecture,* Addison-Wesley Pub Co, March, ISBN: 0201788918.
11. Naumenko, A., Wegmann, A., (2003). Two Approaches in System Modelling and Their Illustrations with MDA and RM-ODP. In *ICEIS 2003, the 5th International Conference on Enterprise Information Systems (2003), p. 398-402.*
12. Lemesle, R., 1998. Transformation Rules Based on Meta-Modelling. *EDOC'98, La Jolla, California, 3-5 November 1998, p. 113-122*.
13. UEML. (2003). Unified Enterprise Modelling Language (UEML) Thematic Network. IST-2001-34229, www.ueml.org.
14. Panetto H., Berio, G., Benali, K., Boudjlida, N. and Petit, M., 2004. A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. *Proceedings of IFAC INCOM 2004 Symposium, April 5th-7th, Bahia, Brazil*
15. B2MML, 2003 The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 2.0, 2003..
16. IEC 62264, 2002. IEC FDIS 62264-1:2002. *Enterprise-control system integration, Part 1. Models and terminology*, IEC, Geneva.
17. Panetto H., G. Berio, K. Benali, N. Boudjlida and M. Petit (2004). A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. *Proceedings of IFAC INCOM 2004 Symposium*, April 5th-7th, Bahia, Brazil.