Product oriented modelling and Interoperability issues

Salah Baïna, Hervé Panetto, Khalid Benali

▶ **To cite this version:**

Product oriented modelling and Interoperability issues

Salah Baïna[1], Hervé Panetto[1], Khalid Benali[2]

[1] *CRAN - UMR 7039, Nancy-University, CNRS*
Campus scientifique, BP 239
F 54506 Vandoeuvre les Nancy, France,
salah.baina@cran.uhp-nancy.fr, herve.panetto@cran.uhp-nancy.fr,

[2] *LORIA - UMR 7503, Nancy-University, CNRS, INRIA*
Campus scientifique, BP 239
F 54506 Vandoeuvre les Nancy, France,
benali@loria.fr

Abstract. The consideration of Product information or Knowledge management, product traceability or genealogy, and product life cycle management implies new strategies and approaches to manage flows of information that relate to flows of material managed in shop floor level. Moreover, 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 a product oriented modelling and a product oriented interoperability approach based on the use of the "Holon" modelling concept as a means for the synchronisation of both physical view and informational views. The Zachman framework is afterwards used as a guideline to establish product oriented interoperability between enterprise systems.

Keywords: Process Modelling, Manufacturing Systems, Enterprise Integration, Systems Interoperability, Model Driven Architecture, Zachman, Models Transformations and Mappings.

1. Introduction

Enterprise integration and the opening of information systems towards integrated access have been the main motivation for the interest around systems interoperability. Integration aspect and information sharing in the enterprise lead to an organisation of the hierarchy of enterprises applications where interoperability is a key issue (see Figure 1).
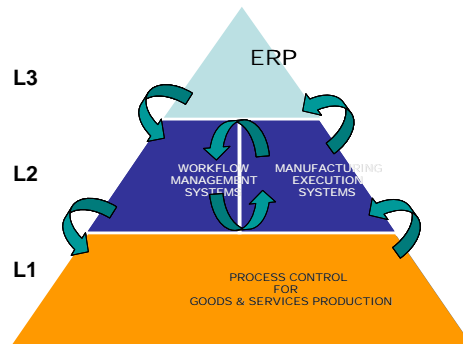
Fig. 1 Manufacturing enterprises common structure

This hierarchy defines the three main levels in manufacturing enterprises:

- L1: Process control level contains all processes that perform routing and physical transformations on the produced goods and services;
- L2: The Execution level performs the processes that manage decision flows (e.g.: Workflow systems) and production flows (e.g.: MES[1], SCE[2]);
- L3: The management system level is responsible for the management of processes that handle all different informational aspects related to the enterprise (e.g.: APS[3], ERP[4] or CRM[5] systems).

To meet traceability, product genealogy and product life cycle management needs, nowadays an enterprise has to manage flows of information that relate to flows of material and that are managed in shop floor level. We assume that the enterprise is composed of two separated worlds (see Fig 2):

(i) On one hand, a world in which the product is mainly seen as a physical object, this world is called the manufacturing world. It handles systems that are tightly related to the shop-floor level,

(ii) On the other hand, a world where the product is seen as a service released in the market. This world is called the business world.

In order to achieve the main objective of the enterprise, "the product" to be specific, the business universe and the manufacturing universe need to exchange information and to synchronise their knowledge concerning the product (good and service). It is assumed that the product (good/service) can play the role of the gateway between both universes, since it represents a common entity between those worlds.

---

[1] Manufacturing Execution System

[2] Supply Chain Execution

[3] Advanced Planning Scheduling system

[4] Enterprise Resource Planning
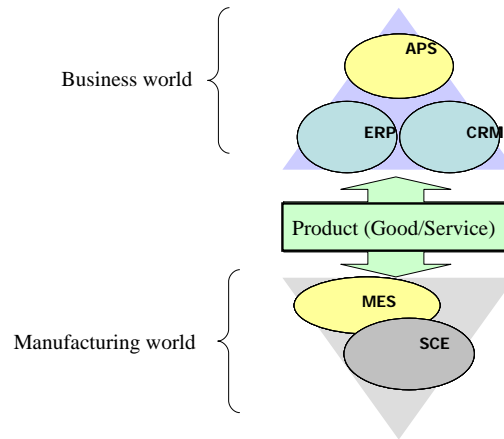
[5] Customer Relationship Management

Fig. 2. Product centric approach

In this paper, we define a holon based approach in order to synchronise views in the business world and in the physical manufacturing world using the holon concept. The paper continues by presenting the usability of the concept of holon in ensuring interoperability enterprise context. Section 2 presents the bases of our holonic process modelling concepts [1] 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 explains how holons can be used as a means for enterprise applications interoperability. In Section 4, an implementation of the holon is proposed. In Section 5, we propose a guideline to model driven and product oriented interoperability based on the Zachman Framework. Section 6 gives conclusions and perspectives for this work.

2. A modelling construct 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 modelling phase. Existing solutions for interoperability in enterprise environment focus mainly on enterprise processes interoperability and interconnection. Throughout product lifecycle, coordination needs to be established between the reality in the physical world where the product evolves as a physical object and the "electronic" world handled by manufacturing information systems where the virtual image of the product evolves as an informational object. Our work aims to provide a product centric approach for enabling interoperability between information systems in the manufacturing environment in order to establish the coherence between the physical products and their informational representations. To take into account this duality (physical things/ informational things), we propose an adaptation of the concept of holon [2] to this specific problem.

The word Holon is a combination of the Greek word *holos*, meaning whole, and the suffix *on* meaning particle or part. A holon is an identifiable part of a system that

has a unique identity, yet is made up of sub-ordinate parts and in turn is part of a larger whole. A Holon has two main features, autonomy and cooperation. Several adaptations of the holon concept have been proposed in several fields. 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 ([3, 4]). 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 our Holon based Process Modelling [5], holons are used to 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.

To formalize our perception of the product using the holon concept, we adapt some constructs from the well known BWW ontology to our specific purpose. Our analysis is based on the ontology initially introduced by Bunge [6, 7] and adapted by Wand & Weber for the information systems field [8, 9]. The BWW ontology has its roots in fundamental problems of conceptual modelling. Wand and Weber recognized that the quality of conceptual models is always dependent on the correspondence between the model and what the model is about.

Our fomalisation distinguish holon characteristics 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 [10];
- Properties related to the holon but which do not correspond to any of the three types of properties; space, shape or time.

Holons can be classified into two categories; *(i)* elementary holons and *(ii)* composite holons:
(i) Elementary holons are the combination of a single informational part and a single physical part.
(ii) Composite holons are the result of the processing and treatment of one or more other holons, this processing can be the aggregation of a set of holons (composite or elementary) in order to compose a new holon or a transformation of one composite holon to obtain a new one.

Figure 3 represents the UML class diagram defining the holon concept meta-model. In order save place and limit the complexity, in this meta-model, we have not represented the many constraints that apply between classes and that are specified using the OCL language as defined in UML specifications [11].

Here is a brief description of this class diagram: The Class *Holon* defines basic attributes for both composite 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 *Composite Holon* is a holon that has been obtained by either by assembling existing holons, or by disassembling existing holons into new ones.

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.
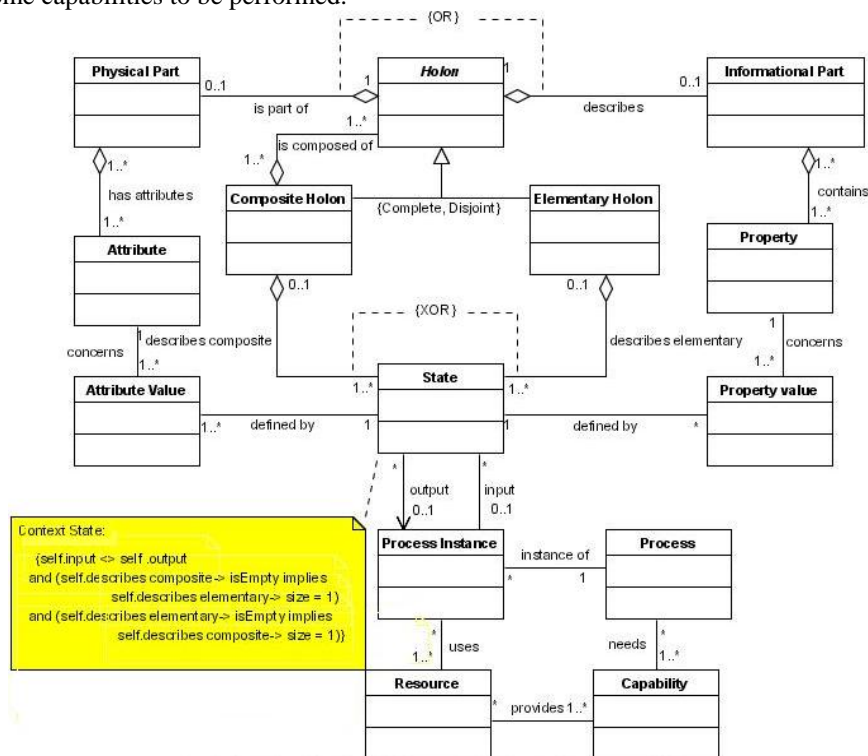
Fig. 3: Class diagram for the Holon model

3. Holon based models 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 [12]

Interoperability can be classified into two categories considering the enterprise hierarchy model:

"Horizontal Interoperability" is the interoperability between applications from the same conceptual level in the enterprise. This first category of interoperability aims to synchronise models that were created in different enterprises even those managed by different modelling systems (e.g.: enabling organisational interoperability between two systems used in two different organisations).

"Vertical interoperability" is the interoperability between applications from different enterprise levels. The objective of this category of interoperability is to maintain coherence between information that is handled in two different level of the enterprise (e.g.: ensuring coherence between organisational models of the enterprise and the process models used at shop floor level).

The following introduces the Levels of Conceptual Interoperability Model (LCIM). Similar to the technical approaches, five levels of interoperability are defined [13]. The focus lies on the data to be interchanged and the interface documentation, which is available. The layers are defined as follows:

Level 0 - System Specific Data: No interoperability between two systems. Data is used within each system in a proprietary way with no sharing. The component (or application) is a black box.

Level 1 – Documented Data: Data is documented using a common protocol and is accessible via interfaces. The component is a black box with an interface.

Level 2 – Aligned Static Data: Data is documented using a common reference model based on a common ontology, i.e., the meaning of the data is unambiguously described. This is also possible by using metadata standards or by using standard reference models. The component is a black box with a standard interface.

Level 3 – Aligned Dynamic Data: The use of the data within the federate/ component is well defined using standard software engineering methods such as UML. This shows the use of data within the otherwise unknown "black box behind the interface," also known as white box.

Level 4 – Harmonized Data Semantic: connections between data that are not related concerning the execution code is made obvious by documenting the conceptual model underlying the component.

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. Vertical interoperability with the MDA approach

In this section, we introduce an approach for interoperability based in a model driven architecture (MDA) [14]. 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.
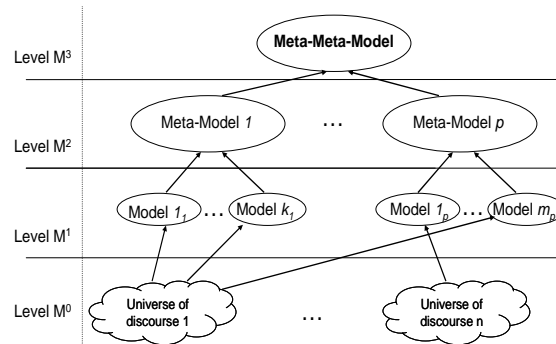

Fig. 4 The four-level ontological approach.

Figure 4 shows the four-level ontological approach levels for modelling that are used in the MDA. As it is explained in [15], 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 this context, applications interoperability may be solved by a top-down approach based on the four levels of the MDA. Indeed the MDA approach for interoperability relies on meta-models mapping to determine, establish and measure interoperability between applications. Several research works have been done in order to resolve meta-models mappings, more generally ontology mappings problems [16].

R Lemesle, in [17], explains how models transformation can be resolved by establishing transformation rules between meta-models. Those transformation rules define a mapping that guides model transformations from the instances of the source meta-model to instances of the target meta-model. Those mappings are the bases for applications interoperability. In the MDA approach for applications interoperability, we consider that each application is based on a specific 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).

In Order to use the MDA approach for interoperability in the holonic context, we need to define roles played by the holon in this architecture, and to position the holonic modelling approach in terms of models, meta-models and universe of discourse: M2, M1 and M0 (see Fig. 6). In the holonic context, the universe of discourse M0 concerns "The Manufacturing Enterprise Product Universe", to describe this universe of discourse we use holonic models (M1) that are instantiations of the holonic meta-model defining holons and their relationships with other entities in their environment (M2).

On the bases of the conceptual model of each enterprise system (M2), transformation rules can be defined between the holon based representation of the product in on hand, and each one of applications conceptual levels on the other hand. Establishing transformation rules between those models enables exchanging product information between different enterprise applications and systems. This information exchange capacity is what we call product oriented interoperability.
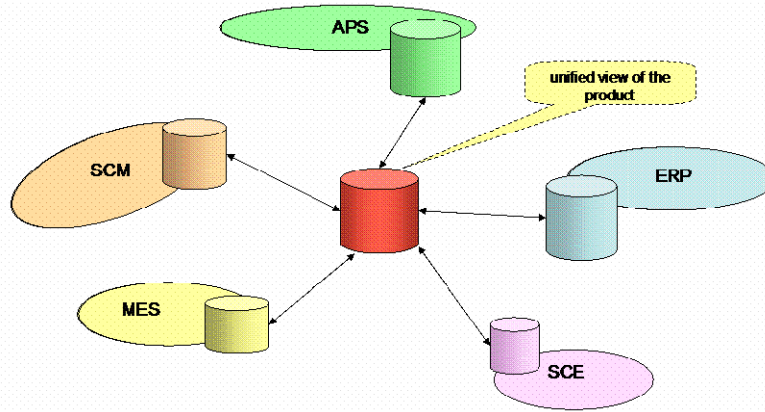
Fig. 5. A unified product view to federate enterprise information systems[6].

Our approach for product oriented interoperability is a model driven approach based on a unified product representation (reference model), and a set of semantic transformation rules. Defining interoperability mappings between the holonic meta-model and other meta-models that handle product information enables the holonic meta-model to play the role of a reference model between those meta-models. Indeed, the holonic meta-model can be seen as a reference model for product representation.

The principle of reference model approach for interoperability is the following: there can be as many local specific models as needed (e.g. one for each system), local models remain as they are, and only transformation rules are defined in order to enable information synchronisation between local models and the reference model.

---

[6] APS: *Advanced Planning System* ; SCM: *Supply Chain Management* ; SCE : *Supply Chain Execution*

Fig. 6. Holon introduction in the MDA four ontological levels

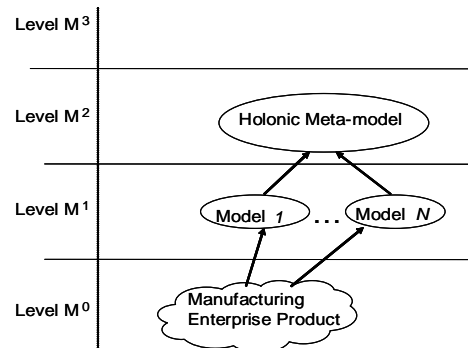In the next section, an implementation of the holonic model and the interoperability mappings is proposed. This implementation relies on a commercial computer assisted software engineering (CASE) tool.

3.2. Holon based models for horizontal interoperability

Horizontal interoperability problem occurs when two or several systems or applications from the same level in the enterprise hierarchy (see figure 1) need to exchange information or data in order to perform a common objective. For example, we consider the case of a manufacturing shop-floor where several manufacturing systems need to cooperate in order to achieve a common goal, the release of the final product to be specific. In this section, we show how the use of the holon concept in the modelling phase, enables considering vertical interoperability concerns at modelling time; in the aim to facilitate resolving interoperability problems during engineering phase. To model manufacturing shop-floor, we use a minimal business process meta-model composed 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.

Site: a geographic place where the enterprise is established. Sites can express a special kind of places such as agency, office and factory, or can also express precise geographic places.

Flow: is a set of elements (data, information, energy, material ...) that are exchanged between processes

To those entities, we add the notions of *Holon* which *represents* products instances. As we see in section 2, a *holon* is described by *properties* and *attributes* that are mandatory for controlling the execution of a process 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 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).

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 the 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 the attribute (or property) or updates it.

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

Using those interfaces, interoperability of processes using can then be defined as explained in the following:

Definition 2: A process P is said interoperable with a system S (composed of processes) iff each input of P is declared as an output of one of his predecessors in S.

The precedence relation between processes is defined as following:

Definition 3: The relation of precedence is 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.

Using the holonic modelling constructs in manufacturing context, enables the considered process interoperability to be concerned at modelling time and not during the engineering phase. This interoperability is a vertical integration of processes, since all process (informational and physical) involved in the studied system are from the same enterprise level, the process control level to be specific. The obtained interoperability is categorised into level 1 of the LCI model, it defines interfaces for shop floor process, that are seen as black boxes, since the designer does not know in advance their internal structure and characteristics.

4. Holons in Action

To experiment the holonic approach defined above in real case we have implemented this approach into a commercial CASE tool named MEGA Suite[7]. MEGA is an enterprise process modelling environment that contains a business process analysis and process modelling and design tools. MEGA has its own meta-model that described all concepts and objects ready to use in MEGA, 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

---

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

Diagrams, Workflows). 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 are based on a meta-model inspired from BPMN[8]. MEGA offers tools that enable customizing the meta-model; we used these tools to embed our own holon meta-model into the existing meta-model of MEGA in order to test the usability of our proposal.

The example presented in Figure 7 shows an example of models that can be designed using the holon modelling concept to represent products in a manufacturing process model. For the sake of simplicity, this example contains only one single process that takes a holon flow as input, and produces a holon flow as output.
In this example, we show using the implementation of the holonic concepts in MEGA, how a process can be connected to information and data concerning holons (inputs or outputs). The holons in this example represent products (finished or not).

To experiment the holon models interoperability with other enterprise modelling frameworks using the MDA approach, two examples have been chosen; UEML and B2MML. UEML [18, 19] is the Unified Enterprise Modelling Language, it is used at the organisational level of the enterprise. B2MML [20] is an implementation of the part 1 of the IEC FDIS 62264 standard [21] developed for interfacing the manufacturing control and execution systems with higher level systems. According to the MDA interoperability approach defined in section 3, we now define an example of mappings from the holonic meta-model to The 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 [22]. The UEML is an Interlingua between Enterprise Modelling tools. The meta-model of UEML1.0 [19] 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 Mark-up Language (B2MML) is an XML implementation of the IEC 62264 part 1. This standard 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 information see Baïna, *et al* [5]. Vertical interoperability that is established by using those mappings is classified in the Level 2 of the LCI model. (see section 3).

To implement the mappings from the holonic models designed in MEGA and the other formats, we first define an extraction format that expresses data extracted from MEGA holon models in order to reuse it in other tools and frameworks based on other meta-models (UEML, B2MML, etc.). To represent the extracted data, we choose the XML language [23]; since it is considered as the standard application data exchange language by the W3C. MEGA Suite enables XML files generation in respect to a specific structure. XML structures for UEML [18], and B2MML [20] are used to transform the mappings defined below into XSLT rules that can be applied on the files generated by MEGA in order to restructure them into files that respects the UEML structure or the B2MML structure.

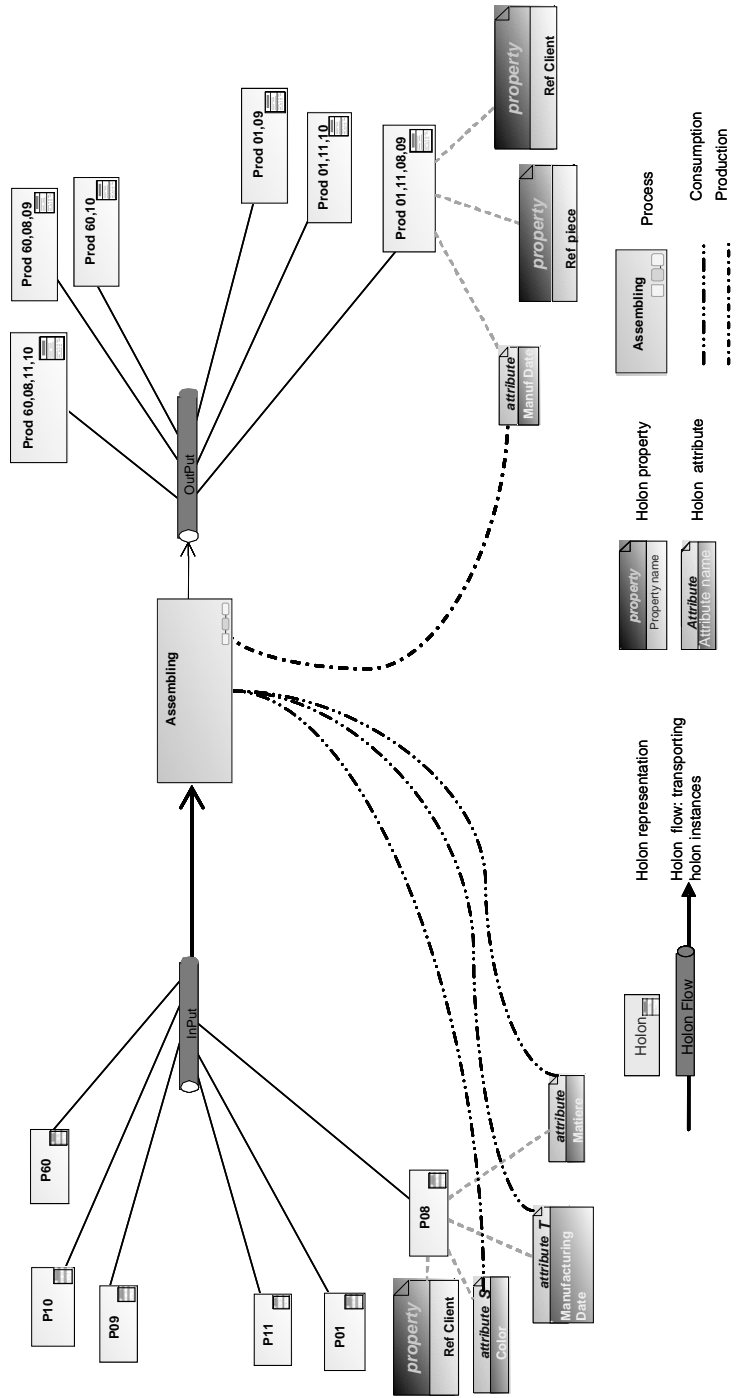8 Business Process Modelling Notation, *www.bpmn.org*

Fig. 7 An example of models containing holons.

5. Model driven Interoperability in the Zachman framework

While well known MDA (Model Driven Architecture) [24] and MDE (Model Driven Engineering) [25] propose frameworks based on OMG specifications, highlighting different abstraction levels of a given system from application development point of view, we contribute to this domain by proposing a framework that combines various points of view of a given system linking manufacture views of a system at the "business to manufacturing" level.

Our approach provides a methodology for product information design and collection, starting from the conceptual model of the product and the manufacturing processes that interact with it, until the definition of product information data models that can be used for product information management (product quality control, traceability or genealogy). The model-driven approach proposed in this paper is derived from the Zachman framework. The Zachman Framework provides a highly structured way of defining and representing an enterprise. It uses a two dimensional classification model based around the 6 basic communication interrogatives (What, How, Where, Who, When, and Why) intersecting 6 distinct model types which relate to stakeholder groups (Visionary, Owner, Designer, Builder, Implementer and Worker) to give a holistic view of the enterprise.

	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner	List of things important to the business	List of processes that the business performs	List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} Business Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Fig. 8. Grid representation of the Zachman framework.

- How: process and functions performed in the enterprise.
- What: important data and objects of the enterprise.
- Who: human actors in the enterprise.
- Where: places, sites and locations where enterprise activity is performed.

- Why: motivations that lead business and manufacturing behaviour.
- When: events that launch activities in the enterprise.

Each cell (or artefact) in this classification must be aligned with the cells immediately above and below it. All the cells in each row also must be integrated with each other. However, cells will not be aligned diagonally.

By definition of Zachman framework, the product of a given enterprise belongs to the scope of the "What" column that describes objects that are important from the enterprise point of view. Enterprise applications and enterprise systems handle information about the product; each one of those systems has a specific representation of the product. Using retro-engineering techniques, a precise logical representation of product view handled by each system can be produced. However, a generic representation of the product is needed at the conceptual enterprise model level to unify all logical views of the product and to enable then a unified product modelling approach.

In Figure 9, holon based modelling approach are integrated into the Zachman framework in order to propose a simplified guide for product oriented interoperability based on the Zachman grid.
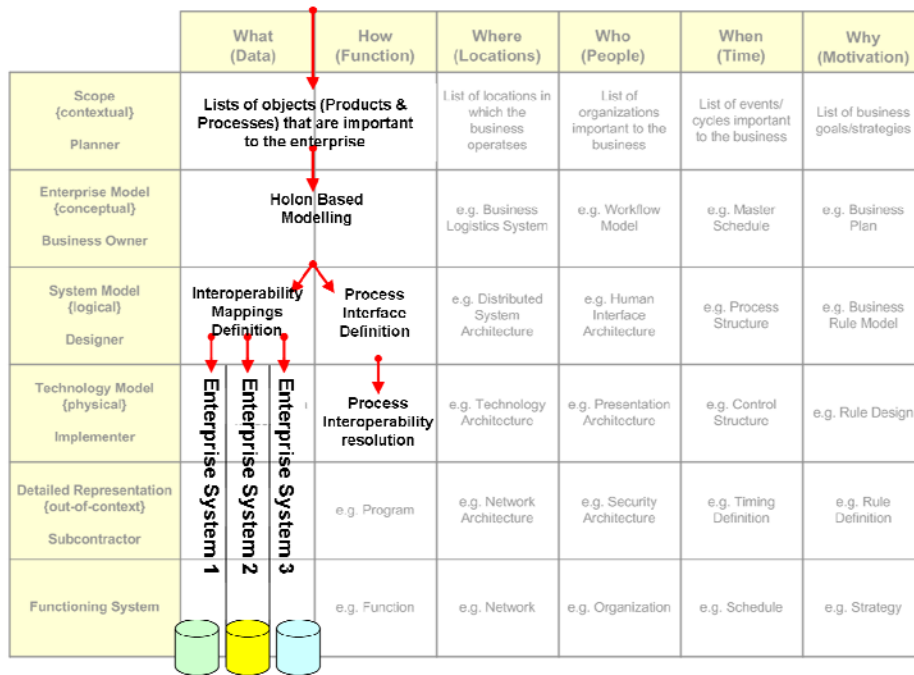
	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner	Lists of objects (Products & Processes) that are important to the enterprise		List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} Business Owner	Holon Based Modelling		e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	Interoperability Mappings Definition	Process Interface Definition	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	Enterprise System 1 Enterprise System 2 Enterprise System 3	Process Interoperability resolution	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor		e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System		e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

Fig. 9. Zachman framework and Holons in action for interoperability

6. Conclusion

In this paper, we defined an approach for specifying the holon modelling construct. The objective of holon based models is the synchronisation between the physical objects and their informational views in manufacturing environment. Then, we introduced how the holon approach can be used for enterprise interoperability issues. Afterwards, an implementation of our approach in a commercial CASE tool is presented. We also establish a translation mechanism based on meta-model mappings that enable applications using the holonic meta-model to exchange models with other applications based on different meta-models, this mechanism is based on the MDA approach for interoperability.

7. REFERENCES

1. Morel, G., H. Panetto, M. B. Zaremba, and F. Mayer, "Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues," *IFAC Annual Reviews in Control*, vol. 27/2, pp. 199-209, 2003.
2. Koestler, A., *The Ghost in the Machine*. London: Arkana, ISBN 0140191925, 1967.
3. McFarlane, D. and S. Bussmann, "Developments in holonic production planning and control," *International Journal of Production Plannig and Control*, vol. 11, pp. 522-536, 2000.
4. Seidel, D. and M. Mey, "IMS - Holonic Manufacturing Systems: Glossary of Terms," in *IMS - Holonic Manufacturing Systems: Strategies*, vol. 1. University of Hannover, Germany: Seidel D. and Mey M. (eds), 1994.
5. Baïna, S., H. Panetto, and G. Morel, "A holonic approach for application interopearbility in manufacturing systems environment. ," in *the 16th IFAC World Congress*. July 4-8, 2005. Prague, CZ., 2005.
6. Bunge, M., *Treatise on Basic Philosophy. Volume 3, Ontology I: The Furniture of the World*, vol. Volume 3, Ontology I: The Furniture of the World. Boston: Reidel, 1977.
7. Bunge, M., *Treatise on Basic Philosophy. Volume 4, Ontology II: A World of Systems*, vol. Volume 4, Ontology II: A World of Systems. Boston: Reidel, 1979.
8. Wand, Y. and R. Weber, "On the ontological expressiveness of information systems analysis and design grammar," *Information Systems Journal*, vol. 3, pp. 217 - 237 1993.
9. Wand, Y. and R. Weber, "On the deep structure of information systems," *Information Systems Journal*, vol. 5, pp. 203 - 223, 1995.
10. Panetto, H. and J. F. Pétin, "Metamodelling of production systems process models using UML stereotypes," *International Journal of Internet and Enterprise Management. ISSN: 1476-1300*, vol. 3/2, pp. 155-169 2005.
11. UML, "The Unified Modelling Language specifications 1.5," OMG 2003.
12. Kalfoglou, Y. and M. Schorlemmer, "Formal Support for Representing and Automating Semantic Interoperability," in *1st European Semantic Web Symposium (ESWS'04)*. Heraklion, Crete, Greece, 2004, pp. 45-61.

13. Tolk, A. and J. A. Muguira, "The Levels of Conceptual Interoperability Model.," in *Simulation Interoperability Workshop*. Orlando, Florida, USA, 2003.

14. Mellor, S. J., S. Kendall, A. Uhl, and D. Weise, *Model Driven Architecture*: Addison-Wesley Pub Co. ISBN 0201788918, 2004.

15. Naumenko, A. and A. Wegmann, "Two Approaches in System Modelling and Their Illustrations with MDA and RM-ODP.," in *ICEIS 2003, the 5th International Conference on Enterprise Information Systems*. April 23-26, 2003. Angers, France, 2003, pp. 398-402.

16. Kalfoglou, Y. and M. Schorlemmer, "Ontology Mapping: The State of The Art," *The Knowledge Engineering Review, Cambridge University Press*, vol. vol. 18, pp. pp. 1-31, 2003.

17. Lemesle, R., "Transformation Rules Based on Meta-Modelling," in *EDOC'98*. November 3rd-5th, 1998. La Jolla, California, 1998, pp. 113-122.

18. Berio, G., V. Anaya, N. Boudjlida, J. Krogstie, and M. Petit, "D3.2: Core constructs, architecture and development strategy," UEML TN IST- 2001- 34229 March 2003 2003.

19. Panetto, H., G. Berio, K. Benali, N. Boudjlida, and M. Petit, "A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models.," in *IFAC INCOM 2004 Symposium*. April 5th-7th, 2004 . Bahia, Brazil., 2004.

20. B2MML, "The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 2.0," *http://www.b2mml.org* 2003.

21. IEC 62264, "IEC FDIS 62264-1:2002, Enterprise-control system integration," Geneva, Switzerland 2002.

22. UEML, "Unified Enterprise Modelling Language (UEML) Thematic Network," in *IST-2001-34229*, 2002.

23. XML, "Extensible Markup Language (XML) 1.0," World Wide Web Consortium 1998 1998.

24. OMG/MDA, "http://www.omg.org/mda."

25. Kent, S., "Model Driven Engineering," presented at In IFM 2002, Third International Conference on Integrated Formal Methods, Turku, Finland, 2002.