



HAL
open science

New paradigms for a product oriented modelling: Case study for traceability

Salah Baïna, Hervé Panetto, Gérard Morel

► To cite this version:

Salah Baïna, Hervé Panetto, Gérard Morel. New paradigms for a product oriented modelling: Case study for traceability. *Computers in Industry*, 2009, 60 (3), pp.172-183. 10.1016/j.compind.2008.12.004 . hal-00288123

HAL Id: hal-00288123

<https://hal.science/hal-00288123>

Submitted on 14 Jun 2008

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.

New paradigms for a product oriented modelling:

Case study for traceability

Salah Baïna¹, Hervé Panetto, Gérard Morel

CRAN (Research Centre for Automatic Control) - UMR 7039, Nancy-University, CNRS

Faculté des Sciences et Techniques, BP 239, Vandoeuvre-les-Nancy Cedex, 54506, France

Abstract

Nowadays, product information is being a key issue for traceability and quality management but also for state-of-the-art technologies such as intelligent product technologies. From this statement, product information design and management are becoming one of the most important steps in order to control quality of services and products that aims at customer satisfaction. However, there is no real method that exists to build the exhaustive representation of products information throughout their production cycle. In this paper, a product oriented modelling approach is proposed in order to provide a complete method for products representation. This modelling approach aims at covering all aspects related to product information and its management.

Keywords: Product representation, Intelligent Product, Product oriented approach, Manufacturing systems, ontology.

¹ Corresponding author

E-mail address: salah.baina@cran.uhp-nancy.fr

1 Introduction

For many, a product is simply the tangible, physical entity that they may be buying or selling. However, since the end of the nineties, the nature of products and even the nature of enterprises have changed due to radical changes in the environment where they evolve. Indeed, the enterprise world attended the creation of several enterprise networks or extended enterprises, in particular networks of suppliers and sub-contractors organised around customers. Products developed by these networked enterprises are resulting of a set of geographically distributed and heterogeneous processes.

In marketing terms, a product consists of three separate dimensions, that, when combined, form the finalised product. In order to actively explore the nature of a product further, those three different views should be considered: the *core* product, the *actual* product, and finally the *augmented* product, these are known as the *three levels of a product* [1, 2].

- The *core* product is not the physical product; it represents the benefit of the product that makes it valuable to customer.
- The *actual* product is the tangible, physical product.
- The *augmented* or *increased* product is the non-physical part of the product. It may contain information and detailed representations of the actual product.

Table 1 Example of a product detailed according to its three dimensions.

| Core product | Actual product | Augmented product |
|---|---------------------------------------|--|
| The benefits or solutions provided by the product: | The physical product: | The extended product – features and attributes (tangible and intangible): |
| <input type="checkbox"/> A comfortable night's sleep for two people | <input type="checkbox"/> A double bed | <input type="checkbox"/> Brand name : Comfy <input type="checkbox"/> Price : \$2,000 <input type="checkbox"/> Quality : High <input type="checkbox"/> Size options : Single, Queen <input type="checkbox"/> Colour : Pink <input type="checkbox"/> Dimensions : 120cm x 190cm <input type="checkbox"/> Labelling : Australian made <input type="checkbox"/> Warranty : 2-years <input type="checkbox"/> Delivery : Free delivery |

Regarding this classification and regarding the distributed architecture of production systems, product information management and product lifecycle management lie on heterogeneous systems and multiple data storage systems with potentially conflicting formats. Moreover, the *actual* product, the *augmented* product and the *core* product could be designed, manufactured and sold in different places. Due to this geographical and institutional separation between the different systems involved in the product lifecycle, it makes it difficult to query, to exchange and to maintain consistency of product information inside the extended enterprise. By analogy with the definition of "*Interoperability*" as the ability of two or more systems to exchange information and having the meaning of that information accurately and automatically interpreted by the receiving system [3, 4], we introduce the "*product oriented interoperability*" paradigm as the ability of different enterprise systems to manage, exchange and share product information in a complete transparency to the user and utilizing essential human labour only. In this context, the consideration of an intelligent product capable to embed and manage all or part of its information, in order to launch and organise its own manufacturing or simply to enable access to its informational items, is probably one of the most promising approaches of the last decade. The paradigm of an improved product able of embedding information is a first step towards the evolution of core products.

In this paper, in order to consider information features embedded into the product, a product oriented modelling approach is proposed. This approach enables a structured method for designing product information. The main objective is the synchronisation of product material flows and product information flows in a given manufacturing environment at modelling phase. Through its lifecycle, the product evolves in a dual way, a physical product a informational product that handles all information related to the lifecycle of the product. On one hand, the physical product interacts with physical entities in the manufacturing environment (processes, machines, transport equipments, etc.), on the other hand, digital

product interacts with the computational environment for production control, quality assessment and traceability management.

The structure of the paper is detailed in the following. Section 2 presents some concepts related to intelligent product and information management capabilities; this section aims at comparing our conceptualisation of the product and some definitions of the intelligent product. In section 3, we introduce conceptual foundations that are used in our formalisation of the product oriented modelling approach. Section 4 presents concepts and meta-models of our product oriented modelling approach. In Section 5, model driven interoperability is proposed. This interoperability approach is based on exchange of product oriented models instantiated according to the proposed meta-models. Section 6 shows a case study where our proposal has been used in order to achieve a product traceability management system in a flour production enterprise. Finally, a summary of the paper is given in Section 7.

2 Improving product oriented models

During the last decade, the concept of « Intelligent Product » has been widely used in manufacturing systems. This concept is often used to indicate a product equipped with a specific technology that enables it to develop certain capacities. Indeed, according to [5] an intelligent product possesses all or some of the following features:

1. A unique identity
2. The capacity of communication with its environment
3. The capacity to retain or store data about itself
4. The capacity of using a language to display its features, requirements etc.
5. The capacity of participating in or making decisions relevant to its own destiny

Using these features, [5] categorises product intelligence into two levels:

Level 1: information oriented intelligence that enables the product to be conscious of its status, and to be able to communicate about it. This level of intelligence is ensured by the first three features of the previous list.

Level 2: In this case the product is supposed able to influence and control operations related to its manufacturing. This level of intelligence is said to be decision oriented. To achieve this level of intelligence the product should ensure all features of the list below.

New technologies such as RFID, Auto-ID, UPnP enable identification and information embedding on the product itself (see [6, 7]). Moreover, technologies related to multi agent systems make it possible involving the product in decision making protocols at shop floor level. The product became then an active entity in decision making process, able of taking control of its own destiny (see [8, 9, 10]).

This paper focuses on information related to the product and the design of this information. Products considered are assumed to be intelligent of level 1 (information oriented) [5]. Indeed, the main concern of our proposal is how to decide what kind of information is needed on the product, what should be the knowledge embedded on the product and what information should it retain? Since the product is assumed to be a physical-digital entity evolving in a pervasive computing system achieving manufacturing purposes, engineers need adapted tools and methodologies that help them managing this new artefact [11]. From the decisional and operational points of view the product is no more seen as a static inactive entity in the shop floor. Moreover, the product, at a final stage, exists only thanks to interactions between intermediary products and computational or physical processes. The deployment of intelligent products carrying information and communication means needs adapted modelling methods and approaches. Current modelling techniques consider either manufacturing specifications of the product in process oriented models or product information features in computational models. The consideration of intelligent products that combines both information and

computational features in addition to physical components of the manufactured product implies necessarily modelling approaches that integrates all aspects of the intelligent product. In addition, each process and operation in the enterprise is able to state information that can be useful for product related decisions; nevertheless, the holistic view of the product is missing. The holistic view of the product is the conceptualisation of what information should be carried out on the product, on RFID tags for example. Product specific views used locally for each process, operation or application can be conflicting, redundant and inconsistent; the constitution of a holistic view of the product should improve product information management. The holistic view of the product is a complete informational representation that takes into account information relevant for all processes that are involved in the product lifecycle; those processes could be design, manufacturing, commercialising or even end life recycling process. This holistic view of the product can then be seen as the representation of this product as it is in the manufacturing environment. This representation is the result of conceptual product information tracking throughout processes at modelling phase.

In order to enable a synchronous modelling of product material flows and product information flows in a given manufacturing environment different aspects of the product should be considered: the physical product that is subject of physical processes in a manufacturing environment, and the digital or informational product that is aware of all information related to its lifecycle. The design of such a product should take into account all aspects related to specifications of the physical product, those of the informational product, and interactions of both facets of the product with their direct environments. This design should be based on a product oriented approach that handles the different aspects of the product.

3 Holons and BWW ontology for product representation

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. As such, a holon is recursive. It has two main features: autonomy and cooperation. In the manufacturing context, a Holonic Manufacturing System (HMS, [12]) is seen as an autonomous and co-operative building block of a system for transforming, transporting, storing and/or validating information and physical objects [13]. In this context, several adaptations of the holon concept for the product have been proposed in literature [14-16]. For our purpose, and in order to design a unique representation of the product, there are some critical features that are mandatory in the holonic representation of a product.

To conceive our dual view of the product (information and material), we propose an adaptation of the holon concept. In our proposal, a holon is considered as the aggregation that combines both the material part of a given product and the set of information related to the product (see Figure 1). In fact, our holon based modelling aims at achieving a complete representation of the product, defined by a set of properties and their values. This representation can be reused for different concerns and in different applications.

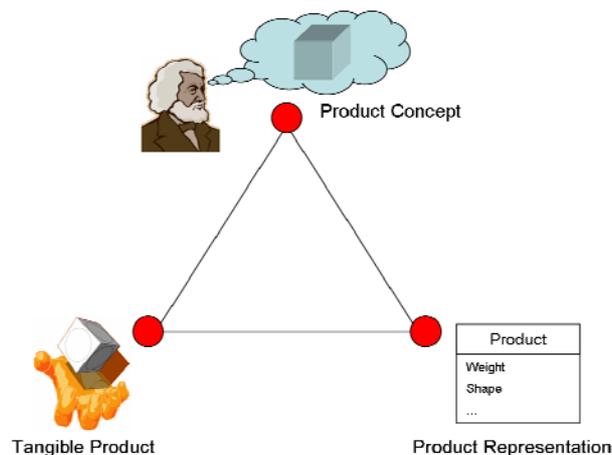


Fig. 1. Relationship between the concept, the reality and the representation of a product.

In figure 1, we illustrate the three dimensions of the product that should be considered in our modelling approach. This classification is an analogy of the three levels of product that have been identified by Kotler in [1]:

- The concept of the product: what one should expect of a given product and how one can imagine its utility.
- The tangible product: the physical object considered as product.
- The representation of the product: the informational part associated to the product.

In our approach, the holon construct is used to represent the concept of the product (the core product) whatever a product illustration is needed.

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 [17, 18] and adapted by Wand & Weber for the information systems field ([19, 20]). 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. They assumed that this correspondence will be greatly supported by using a conceptual modelling language that provides the constructs that are (nearly) the same as the concepts people use to structure their conceptions of the world [20]. To improve the BWW models clarity, Rosemann and Green developed a meta-model of the BWW-model [21]. For the sake comprehensiveness, the following introduces some of the main constructs of BWW [22]:

- Thing: “The world is made of things that have properties”.
- Composite Thing: “A composite thing may be made up of other things (composite or primitive)”.
- Conceivable State: “The set of all states that the thing may ever assume”.

- Transformation of a Thing: “A mapping from a domain comprising states to a co-domain comprising states”.
- Stable State of a Thing: “A state in which a thing, subsystem or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event)”.
- Property: “We know about things in the world via their properties.

In the following, we present some of the important features for product representation; some of them have been inspired directly from the BWW ontology.

4 A generic product representation

In order to propose a methodology for product information design, a meta-model for a generic and unified product representation has to be proposed. This meta-model should group all mandatory features that are connected to the product, its composition and its description. The following introduces those important features.

Product structure. In our context, manufacturing products are represented by holon constructs. Each holon represents what BWW calls a *thing*. Holons are made by successive composition or transformation of raw or intermediary materials. We call product structure, or product composition the tree structure whose nodes represent raw or intermediary materials used to make the product located at the root. Arcs of this tree structure represent composition relationship between different nodes. To take in account this notion of structure, our holon adaptation should consider the link of composition between objects.

Holons can be classified into two categories; (i) simple holons and (ii) complex holons. On one hand, simple holons (elementary holons) are the combination of a single informational part and a single physical part. On the other hand, complex holons (composite 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 a new one, or integrating a set of holons in order to compose a new one. Each composite holon can be defined as the output of the execution of a manufacturing process on one or more less complex holons. If a holon is composed only of one unique holon, then the composition should be seen as a transformation process.

Product features. The informational part of the holon should describe the set of features of a specific product, these features can be distinguished into two categories:

- Characteristics describing intrinsic properties of the physical product. These properties are substantial to the physical part of the product (example: weight, high, material).
- Characteristics describing information that are assigned to the product throughout design phases; each piece of information is of a specific domain and a specific use. (e.g. identifier, production date, etc.)

To distinguish both concepts, we call “attributes” characteristics of the first category, and we call “properties” characteristics corresponding to the second category.

Product states. During manufacturing phases, a product passes throughout several states that describe its history. The management of the set of states of a specific product and relationships between them enable products traceability and genealogy management [23]. The state of a holon is then defined using tuples (attribute, value) et (propriety, value). Each operation processed on a holon implies a state change.

All those constructs have been formalised in a meta-model describing elements that are involved in the definition of a holon representing a physical-digital product [24]. This meta-model has been expressed in the UML class diagram formalism [25].

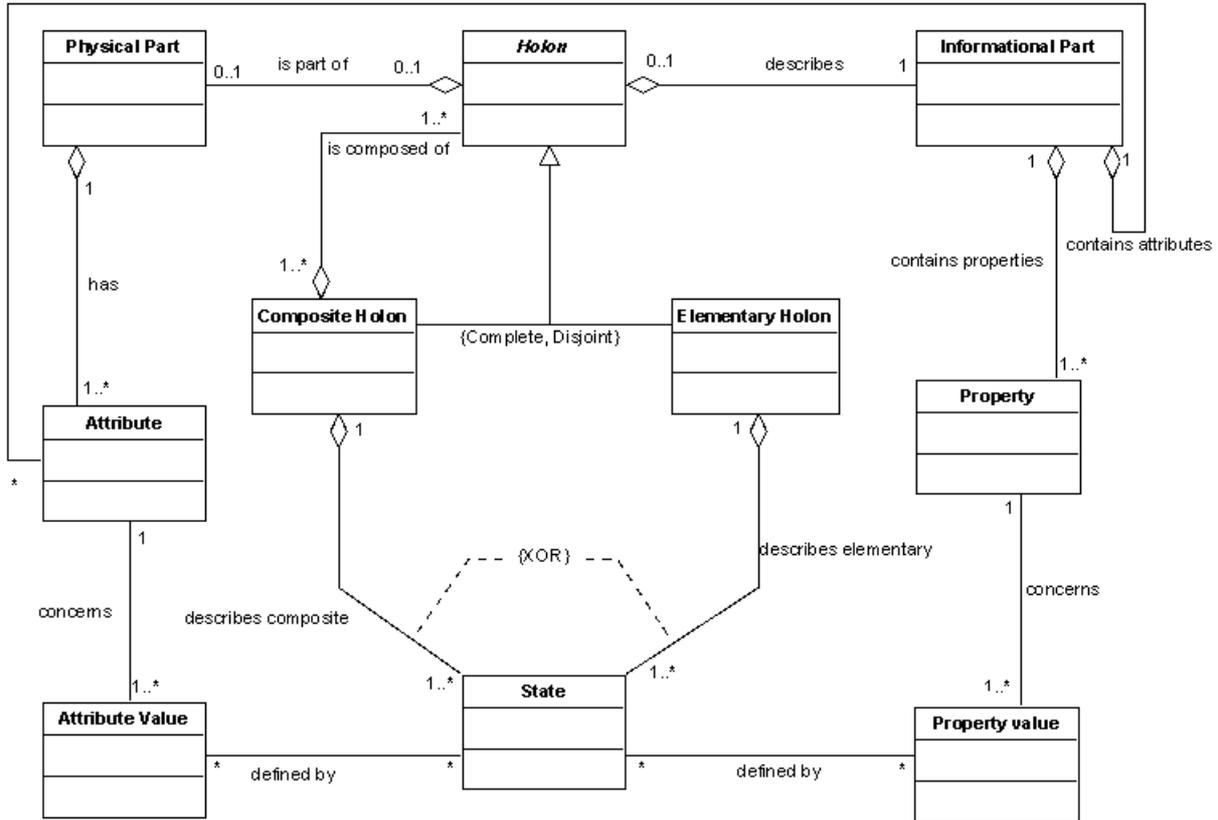


Fig. 2. Holon Meta-model expressed in UML formalism.

In order to preserve the semantic consistency of models resulting of instantiation of this meta-model, we propose a set of semantic constraints that ensure coherence and validity of models. These constraints have been formalized in the OCL formalism which is the language that expresses semantic rules and constraints to be respected by UML classes. OCL is based on object oriented first order logic compatible with UML [26]. The following shows two examples of constraints that have been defined for the holonic modelling.

constraint 1 : The first constraint expresses the fact that only composed holons exist without a physical part that belongs directly to them: « A holon entity that is not linked to a physical part is necessarily a composed holon ».

In OCL this constraint is expressed as following:

```

context Holon
inv constraint1:
self.Physical_Part -> size()=0 implies self.oclIsTypeOf(Composite_Holon) and
self.children ->size() > 0

```

constraint 2 : The second constraint expresses that each elementary holon has necessarily a physical part and an informational part. This constraint ensures that the decomposition of holons ends correctly, and that leafs of the composition tree structure of a given holon are elementary holons.

In OCL this constraint is expressed as following:

```

context Elementary_Holon
inv constraint2:
self.oclAsType(Holon).Physical_Part -> size()=1
and self.oclAsType(Holon).Informational_Part -> size()=1

```

The next section focuses the use of holon based product models as product oriented knowledge bases gathering all product information. Indeed, enterprise systems such as ERP², CRM³ and MES⁴ for example store product information for internal use in their specific data warehouses or databases. Most of the time, they also need to share information [27-30]. Instantiated models can be used as reference models for product oriented interoperability. Indeed, they capitalise information that is to be shared between all different information systems involved in the enterprise.

5 A model-driven approach for product oriented interoperability

In order to establish product oriented interoperability between different enterprise systems we propose a model driven approach based on model transformations [31]. This approach is based on a unified reference model that represents the product.

The principle of reference model approach for interoperability is the following [32]: 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 [33]. The consideration of a new application that needs to interoperate and to exchange product information with other

² ERP: *Enterprise Resource Planning*

³ CRM: *Customer Relationships Management*

⁴ MES: *Manufacturing Execution System*

applications is transparent to existing applications. Adding a new specific local model implies no changes for existing models, but only definition of new transformation rules between the new local model and the reference model. Those transformation rules describe relationships between concepts and entities of the local models (source) and reference model (target) [33, 34]. These rules are defined at meta-model level in order to enable transformations at model level (*cf.* Figure 3).

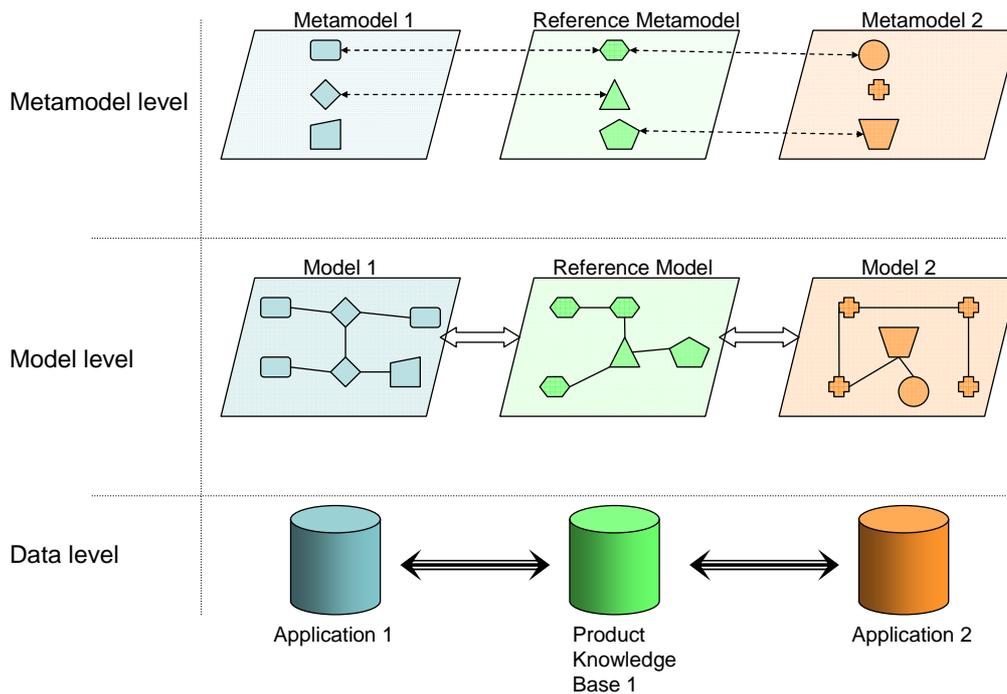


Fig. 3. A model driven approach for product oriented interoperability.

On the bases of the conceptual model of each system, 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 [31]. This interoperability aims at enabling different systems to manage, exchange and share product information in a complete transparency to the user and utilizing essential human labour only [3].

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. This interoperability enables product information mappings between the different applications using a holonic model as a gateway.

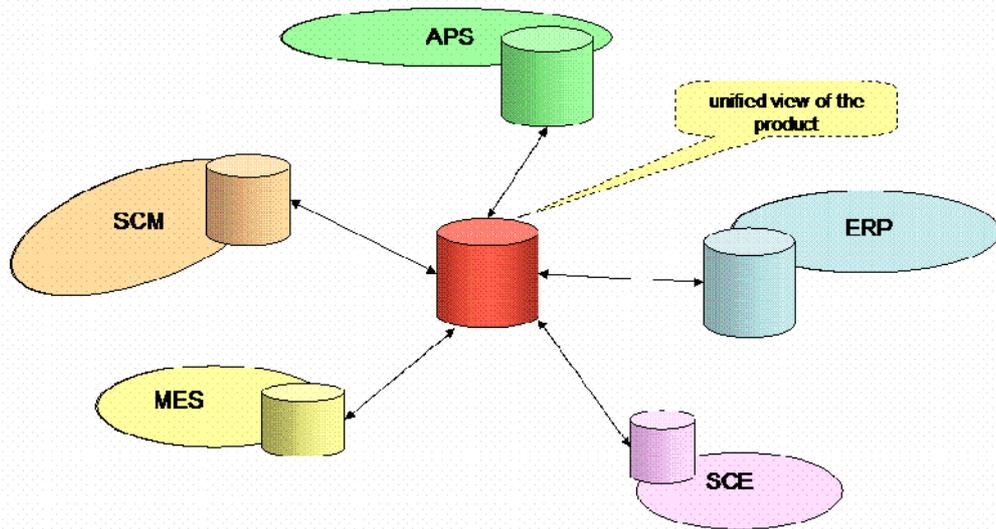


Fig. 4. A unified product view to federate enterprise information systems⁵.

This unified representation is then translated for specific usage and specific applications. The need of transformations is due to the difference of concerns of enterprise applications and systems. Indeed, while manufacturing oriented enterprise systems are interested in physical features of products, business oriented systems are more interested in a service oriented representation of the product more related to informational aspects of products[35].

While well known MDA (Model Driven Architecture) [36, 37] and MDE (Model Driven Engineering)[38] [35] 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

⁵ APS: Advanced Planning System ; SCM: Supply Chain Management ; SCE : Supply Chain Execution

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 [39]. 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 operates | 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. 5. 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.

Many works have been done in trying to map different enterprise architectures frameworks to the Zachman Framework [40-43]. The objective of our work is to propose a product oriented model-driven approach based on the Zachman framework, our proposal takes into account, not only product information models but also models of manufacturing processes that interact with the product among its lifecycle [44, 45]. The Zachman framework is being considered as an important departure from traditional techniques in such areas as software engineering, system engineering and data engineering.

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 this context, we define a product oriented process model as a conceptual model that aims on one hand at defining relations and exchanges and interactions between processes of a manufacturing environment; on the other hand it aims at specifying information related to a product during its lifecycle and its evolution through the set of processes involved in its manufacturing. Regarding this, a product oriented process model combines constructs defined in the holonic meta-model for product representation and constructs defined in standard

process diagrams such as processes, activities, flows and messages. Figure 6 shows a subpart of the meta-model describing constructs involved in our product-oriented process modelling approach.

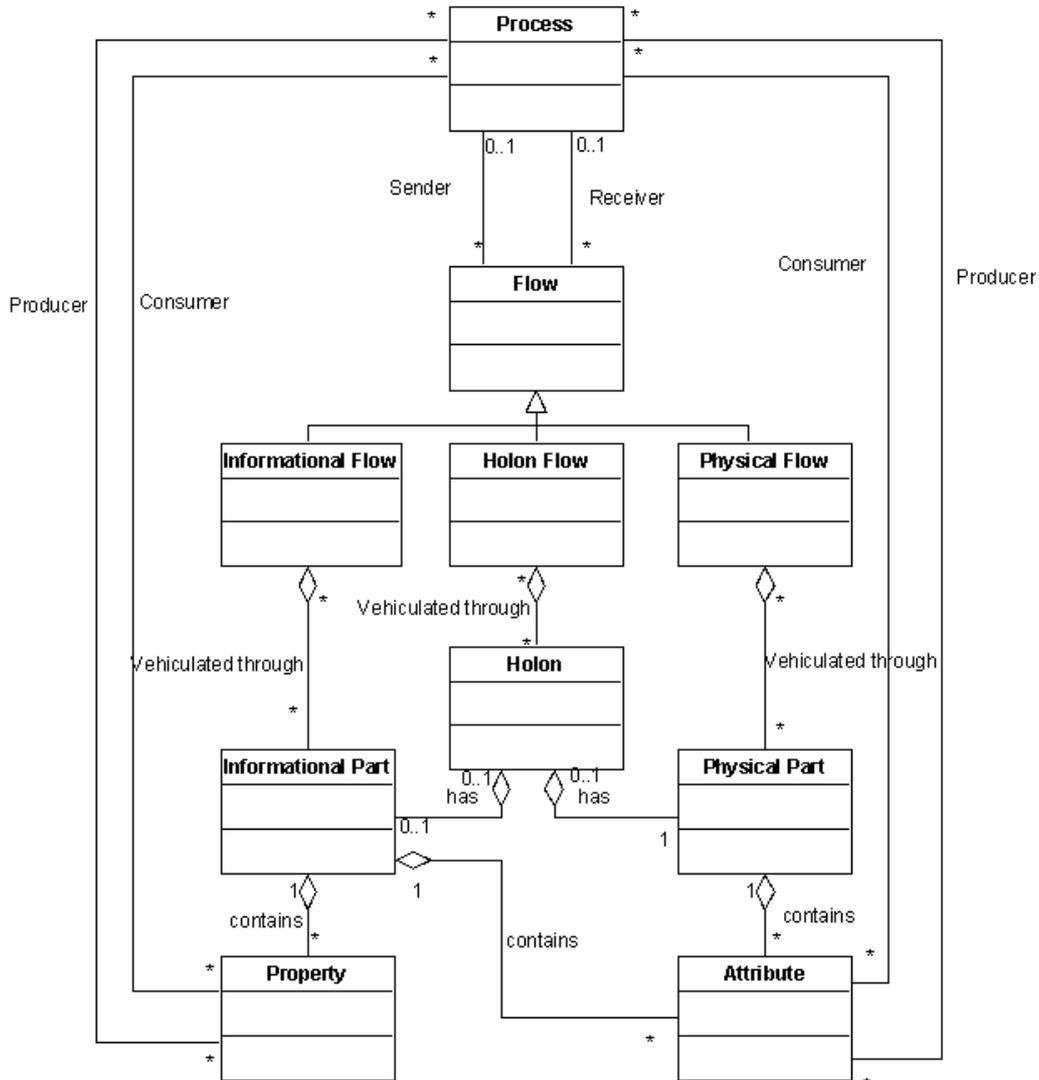


Fig. 6. Part of the product oriented process modelling meta-model.

In addition to the use of holons for product representation, this meta-model emphasizes also interactions between holons and processes. Those models show relevancy of product information for processes by linking each process to pieces of information that it uses or produces. According to the Zachman framework, this kind of models covers both the "What" and the "How" columns of the conceptual level of the Zachman grid, since it describes information schema related to products and in the same time organisation of processes and

exchanges between them. The model-driven approach leaded by the Zachman framework gives an exact idea about kinds of models and information that need to be provided at conceptual level in order to keep coherence between different representations in the different enterprise systems. At the logical level of the "What" column, automatically generated UML class diagrams can be used to express informational models related to the product.

6 Case Study

The proposed case study is the result of a collaboration project with a flour milling enterprise. The company transforms wheat into flour and packs flour into bags of 25 or 50 Kg. To meet quality and traceability requirements, the company has decided to improve product information tracking at shop floor level. From this statement, collaboration has been started in order to model the actual flour milling system in a one of the ten mills of the company. The purpose of this modelling is to specify information related to each important enterprise object that is involved in products release (resources, customers, raw material, etc.). These entities are starting blocks for product traceability system implementation.

Interviews with employees of the company have been performed in order to identify precisely the parts of operational system that should be covered by our modelling approach. These interviews have permitted identification of processes, actors, exchanges and events involved in flour manufacturing and bagging. In order to take into account not only physical objects traffic but also informational parts related to these objects (products, documents, orders), we decided to apply our product-oriented process modelling approach based on holon concept and its features. An implementation of our meta-model has been proposed under MEGA⁶ Case tool which is a commercial modelling environment that offers several tools for enterprise application design and development. Constructs related to holons and flows have integrated in

⁶ MEGA Suite, MEGA International, <http://www.mega.com>

the meta-model of MEGA in order to instantiate them in MEGA diagrams. In this case study, holons represent objects (products and documents) between exchanged processes throughout all phases of flour milling activity. Those objects are described using attributes, and properties that define explicitly pieces of relevant information. To take into account traceability issues, both dependency between objects and structural composition of objects have to be modelled as well. Links between objects enable tracking the set of elements related to a given product, a given delivery or a given customer order. Our approach starts from the first interaction with the customer; this interaction is launched by the arrival of a "*product order*" sent by the customer to the customer relationship management. From this event, the tracking of all exchanges, activities and processes resulting from the interaction with the customer are modelled using our product oriented approach in order to map processes, equipments and humans that acts during product manufacturing cycle. This first model obtained is a very generic one representing a global view of the production system.



Fig. 7. Start point of products tracking.

Figure 7 presents the contextual part of our analysis, describing very abstract view of what our use case is about. From that model, we start investigating in order to define all flows and processes that interact with different phases of preparation, planning, and execution of production. The aim is to map enterprise activities and processes that are involved in product release. During this modelling phase, we apply our product oriented approach by focusing mainly in objects and their features (attributes or properties). The resulting models are to be considered as conceptual level maps for the product oriented approach for enterprise modelling.

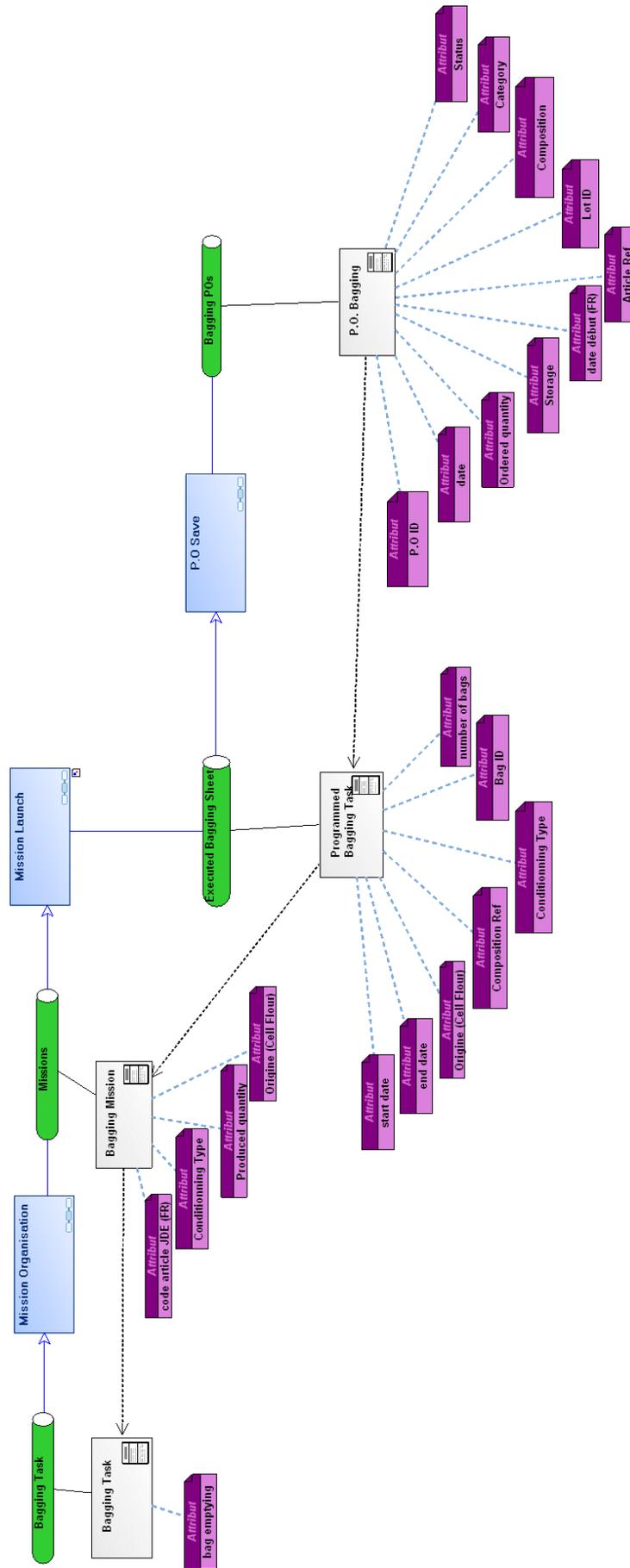


Fig. 8. Example of a product oriented model.

To go further, we implemented an automated generation tool that extracts data from conceptual level models in order to generate "*Logical Models*" representing only entities (products, documents, and important objects for the enterprise) and relationships between them. To formalize this "*Logical Model*", we chose a UML Class Diagram notation. In our case study, the resulting Class Diagram has been used as basis for the development of a traceability management database. Contrary to already existing traceability databases in the host enterprise, the one we propose has been obtained by a methodological approach based on a product oriented modelling of activities of the enterprise. Thanks to our approach, a diagnosis phase has been achieved using the obtained database and data collected in shop floor. This diagnosis has concluded that improvement of data collection has to be done. In deed, some pieces of information are not well filled, these implies important loss of information and loss of connections between elements. Improvements proposed enable a consistent and reliable traceability management.

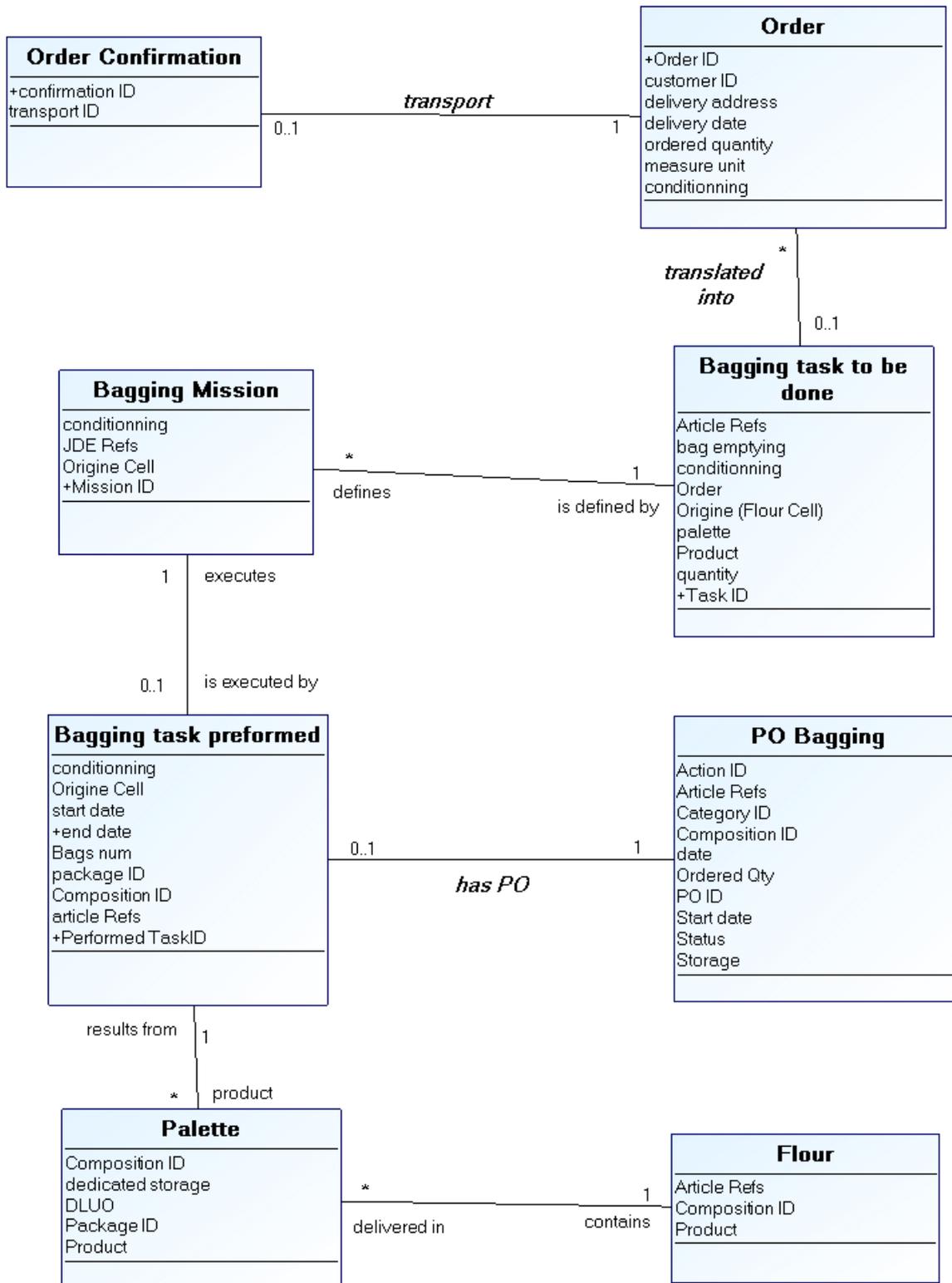


Fig. 9. Automatically generated class diagram.

To summarize the approach used in the case study developed in this section, we propose a support based on the Zachman Framework. The Zachman Framework has been used as a strategy support in order to achieve the product traceability system. Indeed, the Zachman

Framework in its initial definition does not advocate any specific diagram or modelling tool, we tried to place each one of the models and modelling approaches that have been used during our case study in the more adequate Zachman cell where it fits. The use of the Zachman Framework ensures a pragmatic application of the MDA. Indeed, Zachman's *contextual* and *conceptual* levels represent CIM (computational independent models) Level in MDA, the PIM (plate-form independent models) Level of the MDA is represented by *logical* level in Zachman, and the PSM Level (plate-form specific models) are represented by the *physical* technology models and the *out-of-context* detailed representations. Zachman

The first activity concerns the interviewing of enterprise employees in order to collect information that should help the achievement of our product traceability system. This task determines objects and processes that are in the scope of our system. It corresponds to the "*contextual level*". After achieving interviews, conceptual product oriented models based on our holonic modelling approach have been designed. These models fit in the conceptual level of the Zachman Framework, they cover the *What* and the *How* columns of the Zachman grid. Using generation rules implemented in MEGA, product oriented models are derived into logical models formalized in UML (*Zachman Logical level*) representing product data models. In order to implement product traceability database, the logical models are then transformed into a relational schema corresponding the structure of the relational database to be produced. The use of SQL [46] facilities in MEGA enables an automatic generation of script code needed to create the product traceability database.

Figure 10 synthesizes the correspondence between the Zachman Framework (as a support) and the model driven approach that we propose to achieve traceability systems.

| | What (Data) | How (Function) | Where (Locations) | Who (People) | When (Time) | Why (Motivation) |
|--|--|-------------------------------|--|---|---|-----------------------------------|
| Scope {contextual} Planner | Lists of important processes and objects (no formalism) | | List of locations in which the business operates | 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 | Product Oriented Models (Holon models) | | e.g. Business Logistics System | e.g. Workflow Model | e.g. Master Schedule | e.g. Business Plan |
| System Model {logical} Designer | Product Data Models (UML ClassD) | 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 | Product Database Schema (Relational Schema) | 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 | Product Database implementation (SQL) | e.g. Program | e.g. Network Architecture | e.g. Security Architecture | e.g. Timing Definition | e.g. Rule Definition |
| Functioning System | Product Database management System | e.g. Function | e.g. Network | e.g. Organization | e.g. Schedule | e.g. Strategy |

Fig. 10. Used strategy and mapping with the Zachman Framework.

7 Conclusion

In this paper, a product oriented modelling approach has been proposed, in order to provide a structured strategy to build a complete and consistent representation of product information. Product representation resulting of our methodology can be used as a repository for product information management (traceability management, product information design, product information storage). The product oriented modelling approach has been inspired from the use of holon construct in product modelling. Our adaptation of the holon concept defines it as an aggregation of a tangible physical part of a given product, and the informational part that corresponds.

To validate our approach, an implementation and a case study application has been proposed.

Our Approach has been implemented into the MEGA suite modelling environment to enable instantiation of product driven models. Moreover, to provide a fully model-driven approach, we propose throughout the Zachman framework in order to guide user progression and the use of adequate models.

The case study has been performed in collaboration with a French leader enterprise in flour milling. The objective of the collaboration project was the application of our product oriented modelling approach to achieve a unified product representation and to implement it using a database that stores product data. The resulting database is used as a source for product traceability management.

The tool proposed enables product oriented modelling that covers both process requirements, and product data representation. The combination of both issues enables a unified representation of products and important objects for the business process management by tacking into account information that are used and produced at each point of the process execution.

References

1. Kotler, P., *Marketing management: analysis, planning, implementation, and control*. 1988: Englewood Cliffs: Prentice-Hall International.
2. Khosrowpour, M. *Managing information technology resources in organizations in the next millennium*. in *Information Resources Management Association International Conference*. 1999. Hershey, PA, USA.
3. Wegner, P., *Interoperability*. ACM Computing Survey, 1996. **vol. 28**(1): p. pp 285–287.
4. Vallespir, B., D. Chen, and Y. Ducq, *Enterprise Modelling for interoperability*, in *The 16th IFAC World Congress*. 2005: July 4-8, 2005. Prague, CZ.
5. Wong, C.Y., et al. *The intelligent product driven supply chain*. in *IEEE International Conference on Systems, Man and Cybernetics*. 2002 October 10-13 2004 The Hague, The Netherlands.
6. McFarlane, D., *Auto-ID Based Control - An Overview*. 2002, Auto-ID Center, University of Cambridge.
7. McFarlane, D., et al. *The Intelligent Product in Manufacturing Control and Management*. in *The 14th IFAC World Congress*. July 5-9. 2002. Beijing, China.

8. Caridi, M. and S. Cavalieri, *Multi-Agent Systems in Production Planning and Control: An Overview*. Production Planning and Control, 2004. **Vol. 15** (No. 2): p. 106-118.
9. Montalvo, A.C. and A.G. Higuera. *RFID enhanced multi-agent based control for a machining cell*. in *Proceedings of Stimulating Manufacturing Excellence in Small and Medium Enterprises 2005 (SMESME'05)*. 2005.
10. Bussman, S., N.R. Jennings, and M. Wooldridge, *Multiagent System for Manufacturing Control. A Design Methodology*. Springer Series on Agent Technology, 2004.
11. Siegemund, F. and C. Flörkemeier. *Interaction in pervasive computing settings using bluetooth-enabled active tags and passive RFID technology together with mobile phones*. in *IEEE PerCom 2003*. 2003: IEEE Computer Society.
12. HMS, *HMS Requirements*, in <http://hms.ifw.uni-hannover.de/>: HMS Server. 1994.
13. Seidel, D. and M. Mey, *IMS - Holonic Manufacturing Systems: Glossary of Terms*, in *IMS - Holonic Manufacturing Systems: Strategies*. 1994, Seidel D. and Mey M. (eds): University of Hannover, Germany.
14. Bajic, E. and F. Chaxel, *Towards a holon-product oriented management.*, in *The 4th IFAC Workshop on Intelligent Manufacturing Systems (IMS'97)*. 1997: Seoul, Korea.
15. Van Brussel, H., et al., *Reference Architecture for holonic manufacturing systems: Prosa*. Computers in Industry, 1998. **37**(3): p. 255-274.
16. Valckenaers, P., *Special issue on Holonic Manufacturing Systems*. Computers in Industry, 2001. **46**(3): p. October.
17. 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. 1977, Boston: Reidel.
18. Bunge, M., *Treatise on Basic Philosophy. Volume 4, Ontology II: A World of Systems*. Vol. Volume 4, Ontology II: A World of Systems. 1979, Boston: Reidel.
19. Wand, Y. and R. Weber, *On the ontological expressiveness of information systems analysis and design grammar*. Information Systems Journal, 1993. **3**: p. 217 - 237
20. Wand, Y. and R. Weber, *On the deep structure of information systems*. Information Systems Journal, 1995. **5**: p. 203 - 223.
21. Rosemann, M. and P. Green, *Developing a meta model for the Bunge–Wand–Weber ontological constructs*. Information Systems Journal, 2002. **Vol. 27**(2): p. pp 75-91.
22. Fettke, P. and P. Loos. *Ontological evaluation of the specification framework proposed by the "Standardized Specification of Business Components" memorandum - some preliminary results*. in *1st International Workshop on Component Engineering Methodology*. 2003. Erfurt, Germany, Overhage, S. and Turowski, K. (Hsrg.).
23. Terzi, S., *Elements of Product Lifecycle Management: Definitions, Open Issues and Reference Models*. 2005, PhD, Université Henri Poincaré Nancy I (France), Politecnico di Milano (Italy). May 27th, 2005.
24. Baïna, S., H. Panetto, and G. Morel, *A Towards a Product Oriented Process Modelling for Enterprise Applications Synchronisation and Interoperability.*, in *IFAC/IFIP I-ESA'06, 2nd Interoperability for Enterprise Software and Applications Conference*. 2006: March, 2006. Bordeaux, France

25. UML, *The Unified Modelling Language specifications 1.5*. 2003, OMG.
26. Warmer, J. and A. Kleppe, *The Object Constraint Language: Precise Modeling with UML*. 1999: Addison-Wesley. ISBN 0-201-37940-6.
27. Selk, B., S. Kloeckner, and A. Albani, *Enabling interoperability of networked enterprises through an integrative information system architecture for CRM and SCM*, in *The 3rd Business Process Modelling Conference*. 2005: September 5-8, 2005. Nancy, France.
28. Dassisti, M., H. Panetto, and A. Tursi, *Product-driven Enterprise Interoperability for Manufacturing Systems Integration. Proceedings of the BPM2006 Business Process Management Workshops*, in *2nd ENEI Workshop, Springer Verlag*, . 2006, Lecture Notes in Computer Science, LNCS 4103, 249-260, ISBN 3-540-32595-6: Vienna, Austria, September 4, 2006. .
29. Molina, A., D. Chen, H. Panetto, F. B. Vernadat, and L. Whitman, *Enterprise Integration and Networking: Issues, trends and vision*, in *International Conference on Enterprise Integration and Modeling Technology (ICEIMT'04)*. 2004, Kluwer Academics Publisher: Toronto, Canada.
30. Linthicum, D.S., *Enterprise Application Integration*. 1999: Addison Wesley. ISBN 0201615835.
31. Baïna, S., H. Panetto, and K. Benali, *A Product Oriented Modelling Concept: Holons for systems synchronisation and interoperability.*, in *ICEIS'06, 8th International Conference on Enterprise Information Systems*. 2006: 23 - 27, May 2006. Paphos, Cyprus.
32. Vernadat, F.B., *Enterprise modelling and integration: principles and applications*. 1996: Chapman & Hall. ISBN: 978-0-412-60550-5.
33. Lemesle, R., *Transformation Rules Based on Meta-Modelling*, in *EDOC'98*. 1998: November 3rd-5th, 1998. La Jolla, California. p. 113-122.
34. Blaha, M. and W. Premerlani. *A Catalog of Object Model Transformations*. in *3rd Working Conference on Reverse Engineering*. 1996. November 8-10, 1996. Monterey, California.
35. Baïna, S., H. Panetto, and G. Morel, *A holonic approach for application interoperability in manufacturing systems environment*, in *the 16th IFAC World Congress*. 2005: July 4-8, 2005. Prague, CZ.
36. Mellor, S.J., S. Kendall, A. Uhl, D. Weise, *Model Driven Architecture*. 2004: Addison-Wesley Pub Co. ISBN 0201788918.
37. OMG/MDA, <http://www.omg.org/mda>.
38. Kent, S. 2002. in *In IFM 2002, Third International Conference on Integrated Formal Methods*. Model Driven Engineering. Turku, Finland: LNCS 2335. Springer-Verlag.
39. Zachman, J.A., *A Framework for Information Systems Architecture*. IBM Systems Journal 1987. **vol. 26**(3): p. pp. 276-292.
40. Noran, O., *An analysis of the Zachman framework for enterprise architecture from the GERAM perspective*. IFAC Annual Reviews in Control. **27/2**, December 2003: p. 163-183
41. O'Rourke, C., N. Fishman, and W. Selkow, *Enterprise Architecture Using the Zachman Framework*. Thomson Learning, Inc. Boston, MA. ISBN: 0-619-06446-3. 2003.
42. Osvalds, G., *Use of UML in Modeling Enterprise and Systems Architecture.*, in *INCOSE Chesapeake Chapter Meeting*. 2003: Columbia.
43. Sowell, P.K., *Consolidated Mapping of C4ISR Framework Products to Federal Framework Models*. . 1999, The MITRE Corporation: McLean, Virginia.

44. Panetto, H., S. Baïna, and G. Morel, *Mapping the IEC 62264 models onto the Zachman framework for analysing products information traceability: a case study* Journal of Intelligent Manufacturing, Springer Verlag. ISSN: 0956-5515., 2006.
45. Baïna, S. and G. Morel, *Product Centric Holons For Synchronisation And Interoperability In Manufacturing Environments.*, in *The 12th IFAC Symposium on Information Control Problems in Manufacturing*. 2006: May 17-19, 2006. St-Etienne, France.
46. Groff, J.R. and P.N. Weinberg, *The Complete Reference SQL 1999*: Osborne.