Traceability and management of dispersed product knowledge during design and manufacturing

Mohamed Zied Ouertani, Salah Baïna, Lilia Gzara, Gérard Morel

Traceability and management of dispersed product knowledge during design and manufacturing

Mohamed Zied Ouertani [1] [a], Salah Baïna [b], Lilia Gzara [c] and Gerard Morel [d]

[a] Department of Engineering, University of Cambridge, UK

[b] ENSIAS, Rabat, Morocco

[c] Grenoble Institute of Technology – G-SCOP, France

[d] CRAN – UMR 7039, Nancy-University, CNRS, France

[1] Corresponding author *E-mail address*: mzo21@cam.ac.uk

Accepted Manuscript in Computer-Aided Design http://dx.doi.org/10.1016/j.cad.2010.03.006

1. Introduction

Engineering product development in today's industry is becoming increasingly knowledge-intensive and collaborative. As the people, departments, and partners involved in product development tend to be geographically dispersed, there are increasing demands to globally cooperate during the design phase and outsource the manufacturing processes (Li and Qiu, 2006). Due to this geographical and institutional separation between the different systems involved in the product lifecycle, querying and sharing product knowledge is becoming a key issue in the information systems of extended enterprises. In order to avoid lengthy product development cycles, collaboration across distributed and multidisciplinary design teams has become a necessity. Today's knowledge-intensive product development environment requires a framework which effectively enables capture, representation, retrieval and reuse of product knowledge. This is the essence of Product Lifecycle Management (PLM) (Ameri and Dutta, 2005). Moreover, although each process and operation in the product development cycle is able to state information that can be useful for decision making, a holistic view of the product is missing. By holistic view we mean the conceptualisation of what knowledge should be carried within the product. Product specific views used locally for each process, operation or application might be conflicting, redundant or inconsistent; the constitution of a holistic view of the product should improve the product lifecycle management. The concept of PLM is aimed at moving beyond engineering aspects of a product and providing a shared platform for the creation, organization and dissemination of product related information across the extended enterprise. It extends Product Data Management (PDM) functionalities to include the creation of product information as well as management and control of such information.

Currently, the lack of explicit semantics and contexts in the information content to be shared across PLM applications is a major problem (Gerritsen *et al.* 2008). Making data semantics explicit, context aware and sharable among product life cycle applications is a major challenge. For an adaptable organization to function, a product knowledge representation that supports well defined information and knowledge exchange processes among the participants is critical. Indeed, one of the common problems in facilitating an exchange of product knowledge to support collaborative product development is that the actors involved do not have adequate guidance on what kind of knowledge elements should be shared, and how the communication should be structured. Traceability, defined as the ability to describe and follow the life of conceptual or physical artifact, addresses these challenges by providing semantic and structural guidance to knowledge sharing (Mohan and Ramesh, 2007). Our research is based on the premises that an important step towards achieving product knowledge sharing is providing traceability across various product knowledge elements that are used in product development phases, i.e. design and manufacturing. The objective of this paper is therefore to propose a traceability framework that would support tracing and sharing of product knowledge throughout the product development phase of its lifecycle, i.e. product design and product manufacturing. A standardized approach based on the Zachman Framework[2] is adopted to achieve our aim (Zachman 1987).

The remainder of this paper is structured as follows. Section 2 reviews research efforts related to the traceability of product knowledge during the product development process. It first discusses the distinction between data, information and knowledge. Second, it highlights the need for a representation model to trace and share product knowledge. Third, it presents a literature review on traceability for product knowledge management. The objective of this literature review is twofold (i) understanding product knowledge from a design and manufacturing perspectives respectively, and (ii) reviewing current practices and highlights major issues on traceability and knowledge management from design and manufacturing perspectives. Section 3 describes the proposed approach to develop product knowledge traceability. This approach is based on the Zachman Framework. Section 4 provides with the indentified key constructs for product knowledge traceability. Based on this finding, Section 5 formalizes using UML models (2003) the traceability constructs; two different class diagram models are proposed from design and manufacturing perspectives respectively. Section 6 illustrates, by means of two disjoint but complementary case studies, the use of the proposed approach and product knowledge traceability models in design and manufacturing phases respectively. The implementation of the tools, support to the developed models, is not described in this research paper, but references will be provided when appropriate. A discussion is provided in Section 7, while Section 8 concludes the paper.

[2] Zachman Framework is a framework for enterprise architecture, which provides a formal and highly structured way of viewing and defining an enterprise. The Zachman Framework in its initial definition does not advocate any specific diagram or modelling tool. It is also used as reference model for other Enterprise Architectures (DoDAF or IEC 62264)

2. Related works

Substantial research effort has been pursued in knowledge-based systems to support manufacturability analysis in design (Moropoulos, 2003). These typically require the extensive capture of knowledge related to both the product to be designed and its manufacturing environment. This section proposes first to define "product knowledge" in the context of this research work and then presents a state-of-the-art on knowledge representation. Second, the notion of traceability of product knowledge is described. A discussion on related works to product knowledge traceability concludes this section.

2.1.Product Knowledge and Representation

2.1.1. Data, Information and Knowledge

As we move from the industrial age into the information age, knowledge is critical to competitiveness. It is unnecessary for the purpose of this paper to engage in a debate to probe, question or reframe the term knowledge (the authors refer to Alavi and Leidner, 2001 for an extended review of knowledge conceptual foundations). However, in order to leverage knowledge properly, it is necessary to understand its nature accurately. Data, information and knowledge are three concepts which are sometimes used interchangeably. Although it is not always easy to draw sharp borders between them, these concepts have some delicate distinctions.

Within different fields of research many authors have developed definitions for data, information and knowledge (Benyon, 1990; Devine & Kozlowski, 1995; Tomiyama, 1995), and are reviewed extensively by Court (1995) within the context of engineering design.

According to (Ameri and Dutta, 2005), data represents unorganized and unprocessed facts. Information can be considered as an aggregation of processed data which makes decision making easier. Knowledge is evaluated and organized information that can be used purposefully in a problem solving process. Data and information are much easier to store, describe and manipulate than is knowledge.

A distinction is also made by (Gielingh, 2005) between knowledge, information and data; where:

- Knowledge is that what a Knowledge Processor[3] needs to perform a task. Knowledge is a structure of associations between concepts. Concepts are thus the elementary units that form knowledge.
- Information is an *expression* of knowledge. In such an expression, concepts are represented by symbols (such as words) or - in case there is no adequate symbol available - by symbolic structures, such as sentences. Symbols and rules for the arrangement of these symbols are part of a language. Two knowledge processors that wish to communicate must agree about - and have knowledge of - a common language.
- Data are a *manifestation* of information. Data are part of physical reality and form the carriers of information between two knowledge processors. In the case of digital electronic expressions, manifestations often take the form of electric charges or the orientation of magnetic particles.

In this research work, our vision is not different from the previous researchers; moreover it integrates different concepts from each distinction. For the purposes of this work, a distinction between data, information and knowledge is also made. Data is considered to be structured and represent a measure such as a quantity. Information exists when the relationships between *data* are recognized within a specific context describing a fact, where the fact is an occurrence of a measure or inference of some quantity or quality. Knowledge on the other hand is information with added details relating how it should be used or applied.

2.1.2. Product Knowledge Representation

The information and knowledge requirements for global manufacturing decision making have been explored through a multiple-perspective modelling approach comprising IDEF0 activity modelling, IDEF3 process modelling, and Unified Modelling Language (UML) (Dorador and Young, 2000; Kim *et al.*, 2003). A number of reference models and methodologies for modelling generic forms of enterprises are discussed in the research literature, such as CIMOSA (Kosanke, 1999), RM-ODP (ISO/IEC) or CommonKADS (Schreiber *et al.*, 1999).
Product knowledge is defined in (Wu and Huang, 2004) as an integration of the basic product data structure described in the bill of material (BoM) and gradually upgraded throughout the product development cycle. In a broad manner, product knowledge consists of all product information such as order information, design information, manufacturing information and

[3] such as a human being, an electronic knowledge processor or an organization

management information. Research indicates that, in a typical organization, only 4% of organizational knowledge is available in a structured and reusable format and the rest is either unstructured or resides in people's minds (Rasmus, 2002).

A key issue in managing product knowledge is the provision of frameworks to support information/knowledge sharing, because information sharing between participating members is a critical determinant of collaboration. For this purpose, information and knowledge modelling is necessary to build an understanding of the complex information and knowledge relationships which provide the conduit for the effective interactions between different information and knowledge objects.

Rezayat argued that each stakeholder should be able to communicate the rationale behind decisions made in the design and manufacturing of a product to anyone within the extended enterprise (Rezayat, 2000). He then developed a Knowledge-Based Product Development (KBPD) system as an enabler to provide the right information to the right person at the right time and in the right format. Rezayat proposed using the concept of Key Characteristics for defining a communication dictionary for human beings or their agents, and the use of the eXtensible Markup Language (XML) as an enabling tool for making this dictionary function within the web. On the issues of design rationale communication and design intent exchange during the product development, international CAD data exchange standards including ISO 10303 (STEP) have been limited until recently to transferring geometry (Kim et al, 2008). For instance, ISO 10303-108 – Parameterization and constraints for explicit geometric product models – makes possible the capture and transfer of parameter and constraint information, and the representation of 2D sketch. It focuses on a smaller selection of geometric constraints and provides freeform constraint capabilities. On the other hand, the ISO 10303-111 – Elements for procedural representation of solid shapes – provides only representations of operations for the construction of feature-based solid models.

A set of modelling guidelines for the improved reuse of manufacturing knowledge in decision support systems has been proposed in (Cochrane *et al.*, 2008). However, Cochrane's work mainly focused on the principles of knowledge reuse rather than on knowledge modelling and/or representation. This has led to a classification of the information and knowledge to support global manufacturing coordination decisions into three major types: product-related information, manufacturing capability related information and knowledge, and order-related information (Liu, 2004). The respective information and knowledge models are commonly termed product model (PM), manufacturing model (MM), and order model (OM).

In this research work, our vision considers the different perspectives discussed in the abovementioned research efforts. First, product knowledge is seen as product information with added details relating how it should be used or applied. This knowledge is continuously evolving and upgraded throughout the product development lifecycle. Second, in order to manage the created product knowledge, there is a need for modelling the information and knowledge handled during the development process. This will build the foundation for a traceability approach to support information and knowledge sharing. Moreover, the rationale behind decisions made in the design and manufacturing needs to be traced. This will improve the use of product knowledge in future decision making processes.

2.2.Traceability for Product Knowledge Management

It is necessary for the purpose of this research to understand what knowledge management means before studying the role of traceability for product knowledge management.

According to (Yang and Li, 2008), recent research is increasingly focused on knowledge representation, acquisition and management. Knowledge management is defined as the process by which an organization creates, captures, acquires, and uses knowledge to support and improve its performance (Nonaka, 1994; Kwan and Balasubramanian, 2003). Furthermore, knowledge traceability and management could be considered as a Key Performance Indicator for any organization. This perspective would judge the performance of internal processes as well as the innovation effort of the company.

According to (Davenport and Prusak, 1998), most knowledge management projects have one of the following three aims: (1) to make knowledge visible and show the role of knowledge in an organization, mainly through maps, yellow pages, and hypertext tools; (2) to develop a knowledge-intensive culture by encouraging and aggregating behaviours such as knowledge sharing and proactively seeking and offering knowledge; (3) to build a knowledge infrastructure – not only a technical system, but also a web of connections among people given space, time, tools, and encouragement to interact and collaborate. Traceability can play a crucial knowledge management role in product development—if the necessary facilities[4] to support product knowledge management are developed.

We shall now provide an overview of research efforts on traceability of product information and knowledge in the context of building a knowledge infrastructure to improve the product

[4] A framework for analysing the role of information systems in knowledge management is proposed in (Alavi and Leidner, 2001). According to this framework, an organization is viewed as knowledge system with four sets of socially enacted knowledge processes: construction, storage and retrieval, transfer and application.

development process. This basically deals with how to trace product knowledge during the product development process.

There are different usages for traceability management systems (Viruéga, 2006); such as knowledge and data management, standardisation and product return. According to (Mohan *et al.*, 2008) traceability assists in understanding the relationships that exist within and across software artifacts like requirements, design, and software code modules. These relationships help designers ascertain whether and how the design and implementation satisfy the requirements. Moreover, traceability helps understand the rationale behind the design decisions made during system development.

In the specific case of product traceability, one should not only focus on the design rationale behind the development process, but also on components and raw material used to fabricate the product. Furthermore, the history of all processes involved in product development cycle and delivery should be retained.

Currently available PDM/PLM systems support information exchange between product developers, especially in the later phases of the engineering lifecycle which is characterized by more deterministic and well-known processes (Brandt *et al.* 2008). However, they lack essential capabilities for the management and use of product knowledge, for example design knowledge (Maropoulos, 2003).

Some recent research efforts try to extend the capabilities of PDM/PLM systems for product knowledge traceability during the product development phase. Brandt (2008) proposes an ontology-based approach to knowledge management for supporting creative, non-deterministic design processes. A process data warehouse is introduced to capture and reuse of product knowledge. Gao (2003) describes an integration of a PDM system with ontological methods and tools. This integration aims to provide knowledge management capabilities for the conceptual design phase. One of the limitations of the ontology-based approaches is that it can handle only a relatively small amount of instance data. Szykman (2000) describes a design repository to enable storage and retrieval of design artifact knowledge. It is based on a proprietary knowledge base implementation that can be accessed via a web interface.

The key issue with traditional knowledge traceability approach, in particular from the point of view of industrial application, is that it is labour intensive, both for the product-knowledge-engineering specialists as well as for those whose knowledge they are seeking to acquire (McMahon *et al.* 2004). For example, in Gao (2003) knowledge needs to be entered manually and explicitly, based on ontology instances and static rules; experience knowledge is not recorded. In addition, this approach relies on a domain where the processes are well defined and better understood than in process engineering. In the other hand, all queries to the design repository proposed in Szykman (2000) need to be developed manually and coded as specific algorithms. This design repository is limited to the storage of product data and documents. It does not offer support for the recording and management of the associated work processes and decision-making procedures. In addition to these limitations, the proposed product knowledge representations for knowledge traceability are domain-dependent or lifecycle phase-dependent. Indeed, none of these solutions were applied to different product development phases, such as the design and manufacturing phases.

2.3.Synthesis

Different major points emerge from the above discussion: (1) a great emphasis is given to understanding the difference between data, information and knowledge. (2) It has been commonly stated in the reviewed literature that knowledge is personalised. In order for an individual or a group's knowledge to be useful for others, it must be expressed in such a manner as to be interpretable by the receivers. (3) The success of knowledge sharing depends on many factors such as trust and economic value of information. However, once the participating members in a global enterprise are willing to share knowledge, it is technically crucial to design **common knowledge structures to be able to trace the right knowledge and make it available for making better decisions**. Product knowledge has complicated semantics and heterogeneous constraints. Therefore a product knowledge traceability model should define and represent these semantics in order to subsequently share product knowledge.

Moreover, the role of product knowledge traceability to improve the product development process has been emphasized in the reviewed literature. Product knowledge traceability is achieved by capturing and sharing the knowledge created during the product design and manufacturing processes. However, not only product knowledge has to be traced but also the activities and design rationale behind the decisions made while creating this knowledge. Design rationale is defined as, "…*not only the reasons behind a design decision but also the justification for it, the other alternatives considered, the tradeoffs evaluated and the argumentation that led to the decision*" (Falessi *et al.*, 2006). Meanwhile, we believe that effective traceability would consist of the development of system and framework stating clearly the information related to traceability constructs and links, and the way to implement them in a standardized manner, instead of depending on specific features.

In light of these insights the scope of this research work is defined as follows. The contribution of this paper would be to propose an approach for traceability of product knowledge during product design and manufacturing. The benefit of such an approach, compared to the reviewed research efforts, would be on its holistic view where not only specific information on the product are traced, but also the context where this product has been designed and manufactured. The proposed modelling of the product knowledge considers therefore the semantic and syntax of the traceability constructs.

In order to come up with such approach, we propose to use a standardised strategy based on the Zachman Framework. "*The Zachman Framework*™ typically is depicted as a bounded 6 x 6 'matrix' with the *Communication Interrogatives* as Columns and the *Reification Transformations* as Rows. *The Framework* classifications are represented by the Cells, that is, the intersection between the *Interrogatives* and the *Transformations*. This matrix would necessarily constitute the total set of descriptive representations that are relevant for describing something..." (Zachman, 2008).

Following this standardised approach, traceability constructs are first identified. Then, product knowledge traceability UML models are provided for design and manufacturing which are addressed separately. Finally, case studies illustrate the use of the proposed models. We shall now discuss in detail the adopted standardized approach.

3. Approach outline

In order to follow a standardized approach, the Zachman Framework (cf. Figure 1) is used as a support strategy to build the traceability system. We describe hereafter the "reification" rows, which are the transformation of an abstract idea into an instantiation:

- The planner's view (contextual level) corresponds to an executive summary for a planner who wants an overview of the scope of the system.
- The owner's view (conceptual level) illustrates how the business entities and processes relate.
- The designer's view (logical level) determines the data elements, process flows, and functions that represent the business entities.
- The implementer's view (physical level) accomplishes the detailed design based on the designer's plans.

- The subcontractor's view (out-of-context level) corresponds to the detailed specifications that are given to programmers who code individual modules without being concerned with the overall context or structure of the system.
- The functioning system corresponds to the actual system view.

In the frame of this research work, the reification transformation is defined as follows. First, at the **contextual level**, the traceability constructs and links are described. A formalisation is then provided for the product knowledge traceability constructs at a Zachman **conceptual level**. At a Zachman **Logical level**, product knowledge UML class diagrams are developed. At this stage, two different UML models were developed depending on the product lifecycle phase. For the product design phase, a design-rationale based traceability model is developed, while for the product manufacturing phase, a holonic traceability model is provided. The motivation for this choice is the nature of information and knowledge dealt with during each of these lifecycle phases and how suitable those concepts are (i.e. design rationale and *holon*) to achieving the traceability goal. We should note that the implementation of the proposed product knowledge model is developed under the MEGA[5] Case tool which is a commercial modelling environment that offers several tools for enterprise application design. A Relational Schema for the product database is derived from the developed UML models (the Zachman **Physical level**). A SQL database is then implemented based on the relational schema specifications referring to Zachman's **Out-of-Context level**. Finally, Human Machine Interfaces are accordingly developed to serve the Zachman's **Functioning System level**. In addition to the UML class diagrams, UML use cases and sequence diagrams were developed for the specification of the product knowledge traceability system. However, for the sake of coherency and conciseness of the paper the implementation phase is not described and only the UML class diagrams are described. The authors refer to (Baïna, 2006 and Ouertani, 2007) for further details about these specifications and implementation.

[Insert here Figure 1]

In order to illustrate the proposed approach, consider the case of the development of an Electrical Control Unit (ECU) for the motors powering a plant's central conveyor line. At the "contextual level", the *planner* describes, using natural language, the product knowledge to be traced during the design and manufacturing of the ECU. These requirements might be

[5] MEGA Suite, MEGA International, http://www.mega.com

derived from past experiences, or requirements for maintenance activity to keep the plant running at peak performance. Examples of product knowledge to be carried with the ECU are: design specification, ECU identification number, ECU calibration, plant configuration, manufacturing capabilities or ECU installation and safety procedures. At a "conceptual level", the *owner* details these requirements to trace specific product knowledge and formalizes the business rules to design and manufacture the ECU. Examples are: ECU technical specifications or the motor's calibration and instrument reading. At the "logical level", the *designer* proposes a traceability model to trace the product knowledge. In this paper, UML class diagrams are adopted to model the product knowledge traceability constructs. For the design phase, we propose to use a design rationale based model, while for the manufacturing process, holonic-based model is applied. Finally, the implementer and subcontractor develop and implement the product database to support product knowledge traceability.

Having described the overall approach, the following section discusses first step forward the definition of the traceability constructs.

4. Product knowledge traceability constructs

The approach proposed in this paper to define traceability constructs is inspired by the Zachman Framework. Originally, the Zachman framework provides structured mechanisms to define and represent an enterprise. To give a holistic view of the enterprise, the framework uses a two-dimensional classification model based around (i) six basic communication interrogatives: What, How, Where, Who, When, and Why, and (ii) six distinct model types corresponding to stakeholder groups: Visionary, Owner, Designer, Builder, Implementer and Worker. The adaptation in this work consists of using the basic communication interrogatives to define and represent a *product* and how the different actors interact with this product during the design and manufacturing phases of the product lifecycle.

Here, the primary focus is on managing traceability across the various product knowledge created during the product development process. The proposed formalization is intended to represent knowledge in the following dimensions:

- What product knowledge is created or represented?
- Who are the actors playing different roles in creating, using or modifying product knowledge?
- Where is the product knowledge created and located?

- How is the product knowledge being created or modified?
- Why was certain product knowledge created or modified?
- When was the product knowledge created or modified?

Based on these questions, the product knowledge traceability constructs are formalized in Figure 2 and this will define what product knowledge is to be considered.

[Insert here Figure 2]

As discussed earlier, with product knowledge traceability, unlike the reviewed literature one should not only focus on the design rationale behind the development process, but also on components and raw material used to fabricate the product. Furthermore, history of all processes involved in product development cycle as well as the use, the localisation and the history of given entity during its lifecycle by means of an identification mechanism should be retained. There are several usages of this traceability. In the following, we distinguish two perspectives for traceability:

- *Design perspective*

  From product design and development perspective, traceability is defined by Hamilton and Beeby as the ability to discover the design history of every feature of a product (Hamilton and Beeby, 1991).

- *Manufacturing Perspective*

  From the manufacturing perspective, traceability is seen as a mean to control quality of products and processes during the production cycle. Traceability can also be used to verify the accuracy of processes and their conformability and agreement to national or international standards and calibration (Viruega, 2005). This Manufacturing perspective of traceability can be used for example in the case of Quality Management and Standardisation. It also enables product call back in case of failure or detection of production problems in some parts of the product.

We present in Figure 3 a simplified formalization of both these perspectives of traceability. This representation is then translated for specific usage and specific domains, while considering the different traceability construct previously defined. This is mainly due to the different purposes of enterprise systems.

[Insert here Figure 3]

The consideration of both perspectives is not only due to different usages but also to a different semantic of traceability links. Indeed, while the manufacturing perspective of traceability focuses mainly on structural composition (and decomposition) of products, the design perspective focuses on dependency links between information elements. According to (Ramesh, 2002), the traceability of dependency links allows establishing relationships between different traced items and knowledge assets, which are of interest for organizations. The dependency link criterion remains the most studied in the literature and is the most complicated to treat in collaborative design modelling. Different kinds of dependencies exist between two product information/knowledge[6] (Ouertani and Gzara, 2008), such as dependency at creation and dependency at modification. Two elements (i.e. information or knowledge) are said to be 'dependent at creation' if the creation of one of them depends on the creation of the other; and 'dependent at modification' if the change in one of them implies modifying the second one. In addition to these dependency relationships, two other kinds of dependencies are considered between product elements: redundancy and consistency. Two elements are said to be redundant when both of them describe the same entity and are expressed differently. They are consistent when they obey some relationship that is prescribed to hold between them. A special focus is given in this research work to tracking those dependence links created for instance during the product design process. Indeed, although in most design processes, coordination entails clear communication between designers, the real reason for this coordination is not for communication but for resolving dependencies between product elements (Wang and Jin, 1999).

Having identified and formalized the different traceability constructs and emphasized the role of considering the design and manufacturing perspectives, the following section proposes to develop the product knowledge traceability models.

[6] It is important here to clarify that designers do not operate on product information only. In the context of this paper, product *information* could be used interchangeably with product *specifications*.

5. Product knowledge models

The aim of this section is to provide a detailed description of the proposed product knowledge traceability models. First, a traceability model is developed to trace product knowledge during the design phase. This model aims to represent the product knowledge created and used during a design activity while considering the rationale behind it. Second, from a manufacturing perspective, a holonic model is proposed in order to trace product knowledge.

5.1.Product knowledge model from a design perspective

A traceability approach has been developed and traceability constructs were identified (see Figure 2). This helps for representing traceability among product knowledge produced during the various phases of product development. The theoretical foundation of this approach during the design phase is illustrated in this subsection. Section 6.1 presents an industrial case using the tool developed for this purpose.

- What product knowledge is created or represented? This represents the semantic of captured product knowledge with meta-data. Some examples of product knowledge or knowledge elements are design elements, constraints or requirements. The product knowledge could also be embedded into various documents types such as 3D CAD drawings.
- Who are the actors playing different roles in creating, using or modifying product knowledge? Here, the actors are the sources of tacit knowledge and the actor node shown in Figure 2 may be specialized as project manager, designer, customer or software application.
- Where is the product knowledge created and located? Refers to the location where the product knowledge is manipulated within the knowledge network. During the design phase, designers create knowledge during the design activity they are responsible for.
- Why was certain product knowledge created or modified? This to represent the objectives why product knowledge is created and could be represented in text or matrix form.
- How is the product knowledge being created or modified? This is to represent justification or the design rationale behind decisions and hence the creation of product knowledge during the design phase. It is an explanation of why an artifact, or some

part of an artifact, is designed the way it is (Lee and Lai, 1991). The key issues for the justification concept herein adopted are articulated as questions, with each issue followed by one solution that responds to the issue. Arguments support the adopted solution and could be of Requirement-based, Function-based, Rule-based, Case-based and/or standard.

- When was the product knowledge created or modified? The properties of the time node shown in Figure 2 would be date and time the product knowledge was created or modified.

Figure 4 shows in UML formalism the design rationale model describing all elements that contribute to creation of product knowledge during the design phase.

[Insert here Figure 4]

In this diagram, in addition to the traceability constructs described above, the dependency link previously described is also represented and the completeness, variability and sensitivity attributes are proposed to qualify this link. The *variability* is defined as *the likelihood that the output provided by one task would change after being initially released*. The *sensitivity* is *the degree to which work is changed as the result of absorbing a transferred product element.* The *completeness* attribute is used *to draw the actual product element variation interval.* Four subjective levels (0 to 3) of a completeness, variability and sensitivity were proposed from 0 to 3. This scale was developed using the techniques for constructing subjective attributes, as described by Keeney (1992). The three attributes are then aggregated to form one criterion to express the *dependency degree* between two product elements.
The dependency degree is determined using the following formula:

Dependency degree=Completeness*(1 + (Variability * Sensitivity))

As they are complementary attributes, a multiplicative utility function is used to aggregate the variability and the sensitivity attributes (V*S). Furthermore, *higher* the required product element's completeness is, *longer* the required rework is. Thus, for given variability and sensitivity values, the more the completeness is elevated, the longer the iterations are and higher the dependency degree is. In the case where the variability value is 0 and the completeness value is not 0, the dependency degree value must be different from 0, since a “not null completeness” implies that the creation of one product element depends on the creation of the other.

The designers assess the completeness, variability and sensitivity attributes according to their expertise. However, if they have trouble in assessing these attributes, they can be assisted by questionnaires based on structured expert interviews. Examples of questions are: How fixed are the design requirements? (Variability); What is the design risk when a change occurs? (Sensitivity); and Are other knowledge required to achieve to task? (Completeness).

In order to implement the product knowledge traceability database, a standardized approach has been adopted as described in the introduction. Figure 5 illustrates the correspondence between the Zachman Framework and the proposed traceability model, where a unified relational schema is provided corresponding to the structure of the product database to be produced.

[Insert here Figure 5]

5.2.Product knowledge model from a manufacturing perspective

This section proposes an adaptation of the traceability model for manufacturing traceability domain. This adaptation aims at proposing a model for a generic and unified product representation. The model groups all mandatory traceability constructs that are connected to the product, its composition and its description. Section 6.2 presents an industrial case using the tool developed for this purpose.

In this proposal, the concept of holon is used to formalize the representation of the product. 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. In manufacturing vocabulary, a holon is an autonomous and co-operative building block of a system for transforming, transporting, storing and/or validating information and physical objects (Seidel and Mey, 1994). Several adaptations of the holon concept for the product have been proposed in order to take into account the dual view of the product as Informational and Physical.

The Bunge-Wand-Weber (BWW) ontology has been used to formalize the holon. The BWW ontology was initially introduced by Bunge (1977, 1979). The BWW ontology has its roots in fundamental problems of conceptual modelling. For the sake of comprehensiveness, the following introduces some of the main constructs of BWW (Fetke and Loos, 2003):

- Thing: "The world is made of things that have properties".
- Composite Thing: "A composite thing may be made up of other things".
- State of a Thing: "A state that the thing may ever assume".
- Transformation of a Thing: "A mapping from a domain comprising states to a co-domain comprising states".
- Property: "We know about things in the world via their properties.

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 this notion of structure into account, our holon adaptation should consider the link of composition between objects. Holons can be classified into two categories:

- Simple holons are the combination of a single informational part and a single physical part which describe the product feature. A product feature can be distinguished into two categories: intrinsic properties of the physical product—called *attributes*, and information that are assigned to the product throughout design phases—called *properties*.
- 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 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.

During its manufacturing phase, a product passes several states collectively describing its history. The management of the set of states of a specific product and relationships between them enable products traceability and genealogy management (Terzi, 2005). The state of a holon is then defined using the tuples (attribute, value) and (property, value). Each operation processed on a holon implies a state change. Figure 6 shows in UML formalism the model describing all elements that constitute the definition of a holon element representing a physical-digital product (Baina *et al.* 2006).

[Insert here Figure 6]

In addition to the use of holons for product representation, this model also emphasizes interactions between holons and processes. Those models show the relevance of product information for processes by linking each process to pieces of information that it uses or produces. This kind of models covers both the "What" and the "How" columns of the conceptual level of the Zachman grid, since it describes the information schema related to products as well as the organisation of and communication among processes. The model-driven approach led by the Zachman framework gives an exact idea about the kinds of models and information that need to be provided at a conceptual level in order to maintain coherence between different representations in 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.

In order to build a complete approach for traceability oriented product modelling a reference method based on the Zachman Framework is described. The Zachman Framework is used as a support strategy in order to achieve the product traceability system. According to Zachman framework, an enterprise product belongs to the scope of the "What" column that describes important objects from an enterprise perspective. Enterprise applications and enterprise systems handle information about the product and one has a specific representation of the product. Using reverse engineering techniques, a precise logical representation of the 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. The different steps to achieve such a generic representation of the product are now described.

The first step 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". These interviews result in several descriptions of the same products but from different viewpoints (material composition, manufacturing information details). After completing 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, product oriented models are derived into logical models formalized in UML

(Zachman Logical level) representing product data models. It is possible to establish several holon based models describing the different viewpoints of the product. Although those diagrams are separated, they use the same holon object that describes a given product.
In order to implement the product traceability database, these logical models are transformed into a unified relational schema corresponding to the structure of the relational database to be produced. Figure 7 synthesizes the correspondence between the Zachman Framework (as a support) and the proposed model driven approach to develop a traceability system.

[Insert here Figure 7]

6. Case Studies

The following sub-sections present two industrial cases where we applied the above described traceability approach. The first case describes the role of traceability of product knowledge in ECM during a turbocharger development process, while the second case describes an implementation of the holonic model and the traceability approach within a flour milling partner to meet quality requirements.
Constructs related to the product knowledge traceability and flows are integrated in the meta-model of MEGA in order to instantiate them in MEGA diagrams. The main motivation of this unique implementation is to present the product knowledge in a unified manner and to make communication transparent between external parties. Furthermore, MEGA output is easily interpreted by any other PLM system: it can be resolved syntactically using an XML document to transfer knowledge. For semantic issues specific mediators between PLM data structures and our implemented models will have to be identified and defined, but this it out of the scope of our article.

6.1.Product knowledge traceability to support Engineering Change Management

6.1.1. Turbocharger development process traceability

In order to store the various records tracing the design process progression, we consider the questions inspired by the Zachman framework and the modelling of the product knowledge traceability constructs in section 5.1: *What* are the traceable items (product specifications)? *Where* are the traceable items (activities handling the product knowledge)? *Who* are the actors playing different roles in the creation, modification and exchange of product knowledge? *Why* and *How* are product knowledge created, modified and/or evolved in the way they are (the design rationale behind the design activity)? and *When* are the product knowledge created, modified and/or evolved (chronology of design activities).

In order to store the various records tracing the design progression in a database system, the product knowledge traceability UML model (see Figure 4) is used as a specification for a traceability tool called DEPNET[7] (Ouertani and Gzara 2008), which can be seen as an *a posteriori* workflow engine to declare the ongoing design process. Figure 8 illustrates the developed tool used to tracing the product knowledge handled during the design progress.

This screenshot partially shows the turbine wheel designer declaring the turbocharger design process's related information to create new product knowledge, i.e. turbine specifications and attributes. Based on the customer requirements, the turbocharger specifications and the impeller-defined attributes (i.e. the activity inputs to create new product knowledge), the turbine designer - referred to as actor in Figure 2 - commences "to define the turbine wheel" as being the objective of the design activity. The justification and the process sequence to create this new product knowledge consist of the following steps. First, the interdependent parameters, namely the wheel, nozzle ring and insert ring materials, the maximum limit of the turbine inlet temperature, and the inlet and outlet turbine pressure are defined in such a way as to achieve the target turbocharger performance. Once these parameters are defined, the turbine designer can begin to define the wheel dimensions and create the 3D CAD drawing of the turbine wheel. Defining the wheel dimensions involves calculating the exducer and inducer diameters. The designer concludes this part of the design by defining the turbine housing—calculating the turbine attributes 'trim'[8] and A/R[9].

[Insert here Figure 8]

The design of a turbocharger is constraint-oriented and comprises many interdependent parts: turbine, impeller, centre hub rotating assembly (CHRA), and so on. Current product knowledge systems are not able to extract these dependencies and the related knowledge from informal and textual descriptions, and hence the dependencies cannot be revealed. Moreover, the interaction of experts in a collaborative product development context may give rise to engineering change request. This change in one part may have consequences for

[7] DEPendencies NETwork

[8] The 'trim' attribute, which is an area ratio used to describe both turbine and compressor wheels, is calculated using the inducer and exducer diameters.

[9] A/R describes a geometric characteristic of all compressor and turbine housings. It is defined as the inlet cross-sectional area divided by the radius from the turbo centreline to the centroid of that area.

another part, and designers cannot always oversee these interdependencies and consequences. There is then a need to support the ECM process providing them the necessary product knowledge to proceed with the engineering change request.

6.1.2. Product knowledge reuse to support ECM

In the case of the turbocharger development process, an engineering change is requested when the CHRA designer is defining its respective part, regarding the "Turbine A/R" specification. The CHRA assembly contains an oil circuit, a cooling system, a shaft that directly connects the turbine and impeller and supported by a bearing. With the current specification of the "Turbine A/R" and conforming to the turbocharger performance requirements, the CHRA designer is not able to balance the CHRA. The dependency network corresponding to this source of conflict is then extracted using the DEPNET tool and is illustrated in the screenshot in Figure 9. This network refers to the precedence relationship existing between the handled product specifications, to define the turbine part.

[Insert here Figure 9]

In addition to this network, the DEPNET tool allows its user to extract the product knowledge created during the design process as described previously in section 3.1. Hence, in addition to the knowledge about the information and process flows needed to achieve the "Turbine A/R" specification, decision makers dispose of the product knowledge available thanks to the traceability framework. Indeed, design engineers are asked to populate the design repository using the interface proposed in Figure 8. The basic interrogative questions—What, Where, How, Who, Why and When Product Knowledge are answered and stored in a database. The screenshot in Figure 10 illustrates a user interface to consult the product knowledge acquired during the calculation of the "Turbine A/R". This knowledge is available to the turbocharger development team.

The screenshots in Figure 9 and Figure 10 visualise the traced product knowledge which is used to support the designer making effective decision about the engineering change request.

[Insert here Figure 10]

The designer has key knowledge elements needed to make the decision on how to proceed with the engineering change order. Figure 10 illustrates the rationale behind the original decision to set the value of the Turbine A/R the way it is. The issue to be resolved when defining the value of the Turbine A/R, the solution selected for this issue, and the arguments for it are represented in the *justification* area of Figure 10. In addition, when the decision is made, who makes it, who is using the current value of the Turbine A/R and how this value is derived are illustrated in the screenshot above. The set of this product knowledge is instrumental for effective decision making.

In this case, the designer chose not to modify the defined specifications of the "Turbine A/R", the turbocharger (T/C) speed limit will be modified instead. The decision making process to come up with this solution is not described in this paper. However, in order to trace the resolution process, the designers have a tool called CO2MED[10] to support this activity which was developed in previous work (Gzara-Yesilbas et al., 2006). The adopted resolution process is iterative and based on negotiation between the different designers and is organized on successive popularization and mediation phases. During the popularisation phase the designer explains the current change request and argues about the motivation for this request, while the mediation phase aims at advocating an alternative solution to the change request. The resolution process is captured using the CO2MED tool and made available for future similar decisions.

The solution selected to modify the "T/C speed limit" specification corresponds to the change of one or more product specifications involved in the design process. Hence, evaluating the impact of the chosen solution is carried out through the propagation of the product specification changes. This propagation is done through a downstream traversal of the dependency network, starting from the product specification 'solution'. In the considered case study, the solution given to the engineering change order involves modifying the "T/C speed limit". A dependency network is then identified, with the starting data being the "T/C speed limit". The impact of modifying this product specification is illustrated in Figure 11, with a forward coverage of the identified network. A list of the product specifications to be modified is established, as well as a list of the activities for applying these modifications and the rationale behind each of them. Furthermore, decision makers dispose of the necessary product knowledge stored for each design situation traced within the DEPNET tool. Such product knowledge is visualized using user interface such as illustrated in Figure 10.

[10] COllaborative COnflict Management in Engineering Design

[Insert here Figure 11]

6.2.Product knowledge traceability to support manufacturing management

This section presents a manufacturing traceability study resulting from the collaboration with a flour milling enterprise. It describes an implementation of the holonic model and the traceability approach detailed in Section 3.4. The proposed method is based on production process urbanisation and cartography. The advantage of this method in comparison to the other approaches is the association between processes and information, this association ensures information use traceability. Indeed, contrary to existing traceability databases in the host enterprise, the proposal presented in this section has been obtained by a methodological approach based on a product oriented modelling of the enterprise's activities.

The partner 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 the 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.

The aim of the interview process is not to introduce the holon concept to end users since it would influence the comprehensibility of the approach by this users. The holon concept appears only in the formalisation phase, where it helps to express relationships and their types between different categories of products (raw, intermediate, finished). In order to take into account not only physical object 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 concepts and features. An implementation of the model has been proposed using the MEGA Case tool. Constructs related to holons and flows are integrated in 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 the flour milling activity. Those objects are described using attributes and properties that explicitly define relevant pieces of 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 with the first interaction with the customer. This interaction is triggered by the arrival of a "product order" to the department of 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, equipment and humans involved in the product manufacturing stage. The first model obtained is very generic and represents a global view of the production system.

[Insert here Figure 12]

The Figure 12 presents the contextual part of our analysis and is modelled using the MEGA tool. It describes a very abstract view of what our use case is about. Based on this model, we define all flows and processes that interact with the different production phases: preparation, planning, and execution. The aim is to map the enterprise activities and processes that are involved in product release. During this modelling phase, we apply our product-oriented approach by focusing mainly on objects and their attributes or properties.

[Insert here Figure 13]

Figure 13 illustrates the result of the process cartography phase. The output model generated using the MEGA tool describes the shop-floor processes and the flows exchanged between those processes. Each flow is decomposed in terms of holon instances carried out from one process to another. In this case study, customer orders are grouped by type of flour to be produced, production bags of each category of flour are organized into "Bagging tasks". Each task is decomposed into "Missions" and each mission is executed by a set of "Executed tasks" (or subtasks). Each subtask corresponds to a specific "Production Order" called PO Bagging. The result of the execution of those Production orders is a set of flour palettes. Each palette corresponds to a certain category of flour and is described by a set of information.

An automated tool extracts data from conceptual level models and generates a UML diagram describing "Logical Models" that represents entities (products, documents, and important objects for the enterprise) and relationships between them (cf. Figure 14). Indeed, after implementing product oriented models based on holon concept in the MEGA modelling environment, transformation and generation rules were expressed using mega functionalities in order to regenerate UML class diagrams that formalise products dependencies and features in a more standardized syntax. These UML class diagrams represent the relational structure of the core database of the targeted product traceability management system. The transformation rules are presented in the following table.

[Insert here Table 1]

Since the whole holon model is not yet implemented, the translation is not completed yet and some concepts are still missing, for example state. However, the resulting class diagram is detailed sufficiently to be used as relational schema for a traceability management system (cf. Figure 14).

[Insert here Figure 14]

The class diagram in Figure 14 illustrates the identified objects during the mapping stage. For each object type, the model represents the dependency links, the characteristics of each object (i.e. its attributes) as well as the cardinalities between the association classes as defined during the mapping process. Based on this class diagram, an SQL database is automatically extracted to store the information being handled during the production process.

In order to automatically populate the obtained database, basic routines have been created to extract the data already stored in spreadsheets within the different involved departments of the company. We were able to extract a sample of 15000 product orders, 60 production recipes of flour as well as 2000 bagging programmes.

Different queries were developed in order to analyse the accuracy of each of the stored pieces of product information and knowledge and have been used for quality management purpose. The first query finds the delivered product orders of the milling of wheat. The second query finds the palettes having a problem with the bagging process. The last query is to find the palettes with ID tagging issues.

These different queries supported the quality controller in retrieving the concerned palettes as well as the related issues: (i) which palette has been delivered; (ii) which bagging process needs intervention; and (iii) which palette has an ID tagging issue. However, for all three queries the quality controller was unable to find the client for these palettes, because the existing traceability system of the company does not record the link between the palette and the customer. Based on this approach, the diagnosis phase has concluded that improvement of data collection is needed. Moreover, the missing information and product knowledge that need to be traced has been identified.

7. Discussion

The aim of this section is to first discuss the proposed approach with related research efforts. Second, limitations of the proposed approach are presented. A novel approach for traceability and sharing of product knowledge during the product development process has been proposed in this paper. It defines traceability constructs toward product knowledge traceability during the design and manufacturing of a product. This is in-line with the four elements of traceability several researchers agreed on (Cheng and Simmons, 1994), (Steele, 1995) and (Töyrylä, 1999):

- physical lot integrity: how large a lot is and how well lot integrity or separateness is maintained determines the resolution or precision of the traceability;
- data collection: two types of data are identified. Lot tracing data records the movements, and process data records important process data;
- product identification and process linking to determine the product composition; and
- reporting, to retrieve data from the system.

It is believed that this research has important implications for academic and industrial works on traceability of product knowledge during the design and manufacturing phases. Past research on product development and traceability has been isolated from one another. Indeed, traceability of product knowledge such as proposed by Szykman (2000) or Gao (2003) consider only knowledge created during the design phase of a product. Moreover, the experience knowledge is not considered in either papers; the design repository proposed in Szykman (2000), for example, only supports the traceability of product data and documents. On the other hand, Gao's proposal is based on a predefined ontology instances of product knowledge which makes it domain-specific.

This paper proposes a standardized approach to allow the synergistic integration of these areas to trace product knowledge during the design and manufacturing of a given product. For example, research on developing or improving engineering change management methods and tools may use a traceability framework to organize artifact managed within product lifecycle management tools. Moreover, it should be noted that even a team that did designed or manufactured the original product might benefit from traced knowledge, especially in larger projects. The knowledge will help product developers to understand the rationale behind past decisions.

Meanwhile, this research effort has some limitations. According to (Cheng and Simmons, 1994), traceability has diverse dimensions of investigation and classification: Internal and External Traceability. Internal traceability is the traceability inside the factory and the production system, and External, which follows the product the product into its relationships with customers, maintainers, service providers, etc. Regarding this classification, although the proposed traceability approach aims to consider sub-parts suppliers, not all the PLM lifecycle is considered. Only the design and manufacturing phases are addressed in a disjoint, not an integrated, manner.

Another dimension is found in (Jansen-Vullers et al., 2003) where traceability is classified into Backward and Forward Traceability. Backward traceability records information and data on the past history of the product. Forward traceability explains what will happen to a certain product, in terms of operations and processes; this information is written before performing any operation. Compared to this classification, the proposed research effort only addressed the backward traceability. Although we believe that forward traceability would be very useful for decision makers, it has not been tackled in this paper. This will need further development of the exception handling procedures during design and manufacturing.

8. Conclusion and future prospects

This paper proposed an approach to trace and share product knowledge during the design and manufacturing phases of the product lifecycle.

The novelty of this approach consists first of the use of a standardized strategy, based on the Zachman Framework, to build up the traceability approach; and second of addressing the traceability issue from both design and manufacturing perspectives respectively. Constructs to define the product knowledge were proposed and then formalized using UML. The generated UML models were afterwards used as specifications to develop a tool support to tracing product knowledge during design and manufacturing. Two different case studies were finally used to validate those tools.

The research effort presented is based on the premises that an important step towards achieving product knowledge capture and sharing is providing traceability across various product knowledge elements that are used in product development phases. Traceability can play a crucial role to support product knowledge management if the facilities discussed here are developed. The proposed approach for product knowledge traceability in this paper is generic and could be implemented in PLM systems currently in use. Indeed, most of the information required to track the product knowledge during the product development process already exists, dispersed in the different enterprise information systems. However, as to the question of whether product (lifecycle) knowledge can be fused into existing PLM systems, further points remain to be considered to achieve this. These points include (i) research and standardization towards a richer product data/information/knowledge exchange, (ii) stepping up to semantically richer, ontology-based product descriptions, and (iii) integration of non-engineering data.

Future work for the generalization of the proposed approach must be done with caution in light of its limitations. First, we aim to integrate the design and manufacturing perspectives into a single model. Next, we seek to integrate and fully automate, where appropriate and feasible, the proposed traceability tool. Finally, detailed empirical studies for various types of product developments projects are sought in order to further validate the traceability approach proposed in this paper.

References

1. Alavi, M., Leidner, D. E. (2001). Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues. MIS Quarterly, 25(1), pp.107-136

2. Ameri F. and D. Dutta, Product lifecycle management: Closing the knowledge loops. In Computer-aided Design & Application (2005), Vol. 2, No. 5, pp 577,590.

3. Baïna, S. Model Driven Interoperability: A Product oriented approach for enterprise systems interoperability. 2006 PhD Thesis, Nancy University, France

4. Benyon D.. *Information and data modelling*, Alfred Waller Ltd., Henley-on-Thames UK (1990).

5. Brandt S.C., J. Morbach, M. Miatidis, M. Theißen, M. Jarke and W. Marquardt, An ontology-based approach to knowledge management in design processes, Computers & Chemical Engineering, Volume 32, Issues 1-2, 2008, Pages 320-342

6. Bunge M., Treatise on Basic Philosophy. Volume 3, Ontology I: The Furniture of the World. Volume 3, Ontology I: The Furniture of the World. (1977), Boston: Reidel.

7. 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.

8. Cheng and Simmons, Traceability in Manufacturing Systems, International Journal of Operations & Production Management, 1994, Vol.14, No 10, pp. 4-16

9. Cochrane *et al.*, Knowledge reuse in manufacturability analysis, Robotics and Computer-Integrated Manufacturing, Volume 24, Issue 4, 2008, Pages 508-513.

10. Court, A. W. (1995). *Modelling and classification of information for engineering design*. Ph.D. Thesis, University of Bath, UK

11. Davenport, T. H., and Prusak, L. *Working Knowledge: How organizations manage what they know*, Harvard Business School Press, Boston, 1998.

12. Devine D.J. and S.W.J. Kozlowski , Domain-specific knowledge and task characteristics in decision making. *Organisational Behaviour and Human Decision Processes* Volume 64 Issue 3 (1995), pp. 294–306

13. Dorador, J.M. and Young, R.I.M. Application of IDEF0, IDEF3, and UML methodologies in the creation of information models. *INT. J. Computer Integrated Manufacturing*, 2000, **13**(5): 430-445.

14. Falessi, D.; Cantone, G.; Becker, M. “Documenting design decision rationale to improve individual and team design decision making: an experimental evaluation”. In Proceedings of the 2006 ACM/IEEE international Symposium on Empirical Software Engineering. Pages: 134-143.2006.ACM Press.

15. Fettke P. and P. Loos. *Ontological Evaluation of Reference Models using the Bunge-Wand-Weber Model*. in *Proceedings of the 9th Americas Conference on Information Systems*. 2003. August 4-5, 2003. Florida, USA

16. Gao, J. X. Aziz, H. Maropoulos, P. G. Cheung, W. M., 2003, Application of product data management technologies for enterprise integration, International Journal of Computer integrated manufacturing, Vol 16; Part 7/8, pages 491-500.

17. Gerritsen B., W. Gielingh, H. Nowacki, R. Anderl and W. Dankwort, 2008, Editorial – Special Issue on Current State and Future of Product Data Technologies (PDT), Computer-Aided Design, Vol. 40, Issue 7.

18. Gielingh W., *Improving the performance of construction by the acquisition, organization and use of knowledge*. PhD thesis, Delft University of Technology, October 2005.

19. Gzara-Yesilbas, L., Rose, B. and Lombard, M. (2006), Specification of a repository to support collaborative knowledge exchanges in IPPOP project, Computers in Industry, 57, pp 690-710

20. Hamilton V.L. and and Beeby, M-L., 1991, Issues of Traceability in Integrating Tools, Proc. IEE Colloquium Tools and Techniques for Maintaining Traceability During Design, London, UK, 2 Dec 1991.

21. ISO/IEC 10746-1. Reference model for open distributed processing. Part 1: Overview. International Organization for Standardization, Geneva (Switzerland) (1998)

22. ISO 10303-108, Industrial automation systems and integration — Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models, International Organization for Standardization, Geneva (Switzerland) (2005).

23. ISO 10303-111, Industrial automation systems and integration — Product data representation and exchange: Integrated application resource: Elements for the procedural modelling of solid shapes, International Organization for Standardization, Geneva (Switzerland) (2007).

24. Jansen-Vullers M.H., van Dorp C.A., Beulens A.J.M., (2003), Managing traceability information in manufacture, Intl Journal Of Information Management, 23, pp 395-413

25. Keeney R.L. (1992), Value-focused thinking: A path to creative decision making, Harvard University Press, Cambridge (MA)

26. Kim, C.H., Weston, R. H., Hodgson, A. and Lee, K. H. The complementary use of IDEF and UML modelling approaches. *Computers in Industry*, 2003, **50**(1): 35-56.

27. J. Kim, M. J. Pratt, R. G. Iyer, R. D. Sriram, Standardized data exchange of CAD models with design intent, Computer-Aided Design, Volume 40, Issue 7, 2008, Pages 760-777

28. Kosanke K., F. Vernadant and M. Zelm, CIMOSA: enterprise engineering and integration, *Computers in Industry* **40** (1999), pp. 83–97

29. Kwan M.M. and P Balasubramanian (2003), Process-oriented knowledge management: a case study. Journal of the Operational Research Society, Vol. 54, No 2, pp 204-211.

30. Lee, J. and Lai, K. What's in design rationale. Human-Computer. Interaction, 6 (3–4), 251–280 1991.

31. Li W.D. and Z.M. Qiu, State-of-the-art technologies and methodologies for collaborative product development systems, Intl. J. Prod. Res. (2006), Vol 44, No 13, pp 2525-2559.

32. Liu S., 2004, Manufacturing information and knowledge models to support global manufacturing co-ordination. PhD Thesis, 2004, Loughborough University, UK.

33. Maropoulos P.G., Digital enterprise technology-defining perspectives and research priorities, *Int J Comput Integr Manuf* **16** (7–8) (2003), pp. 467–478.

34. McMahon C., A. Lowe and S. Culley, (2004), Knowledge management in engineering design: personalization and codification, Journal of Engineering Design, Vol 15, No 4, pp 305-325.

35. Mohan K., B. Ramesh, 2007, Traceability-based knowledge integration in group decision and negotiation activities, Decision Support Systems, Volume 43, Issue 3, pp 968-989

36. Mohan K., P. Xu, L. Cao and B. Ramesh, 2008, Improving change management in software development: Integrating traceability and software configuration management, Decision Support Systems, Volume 45, Issue 4, November 2008, Pages 922-936.

37. Nonaka, I. (1994). *Dynamic Theory of Organizational Knowledge Creation*. Organizational Science, Vol. 5, n°1, 1994, pp. 14-37.

38. Ouertani. M.Z., DEPNET: A methodology to support design conflict management based on product information dependencies tracking. 2007. PhD Thesis, Nancy University, France.

39. Ouertani M.Z. and L. Gzara, Tracking product specification dependencies in collaborative design for conflict management, Computer-Aided Design, Volume 40, Issue 7, July 2008, Pages 828-837

40. Ramesh B., Process Knowledge Management with Traceability. IEEE Software, vol. 19, issue 3, pp. 50-52. IEEE. 2002.

41. Ramesh B. and M. Jarke, (2001), Towards reference models for requirements traceability, IEEE transactions on software engineering. Vol 27, No 1, pp 58-93

42. Rasmus D. (2002), Collaboration, contents and communities, an update, Giga Information Group Inc., May 2002.

43. Rezayat M., 2000, Knowledge-based product development using XML and KCs, Computer-Aided Design, Vol 32, Issues 5-6, Pages 299-309

44. Schreiber G, Akkermans H, Anjewierden A, de Hoog R, Shadbolt N, Van de Velde W, Wielinga B. The CommonKADS methodology, 1999. ISBN 0-262-19300-0 (http://www.commonkads.uva.nl)

45. 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.

46. Steele D.C.A., Structure for lot-tracing design. *Production and Inventory Management Journal* **36** 1 (1995), pp. 53–59

47. Szykman S., R.D. Sriram, C. Bochenek, J.W. Racz and J. Senfaute, "Design Repositories: Engineering Design's New Knowledge Base," IEEE Intelligent Systems, vol. 15, no. 3, pp. 48-55, May/Jun, 2000

48. Terzi S., Elements of Product Lifecycle Management: Definitions, Open Issues and Reference Models. (2005), PhD Thesis Université Henri Poincaré Nancy I and Politecnico di Milano (Italie), 27 mai 2005.

49. Tomiyama T., A design process model that unifies general design theory and empirical findings. *ASME Design Engineering* **83** 2 (1995), pp. 329–340.

50. Töyrylä I., 1999, *Realising the potential of traceability.* Ph.D. Thesis, Helsinki University of Technology, Espoo, Finland

51. UML, *The Unified Modelling Language specifications 1.5*. 2003, OMG.

52. Viruéga, J.-L., *Traçabilité: Outils, méthodes et pratiques*. 2005: Éditions d'Organisation. ISBN : 2-7081-3260-1.

53. Wang KL, Jin Y. Modelling dependencies in engineering design. In: The ASME design theory and methodology conference, DETC99/DTM-8778. 1999

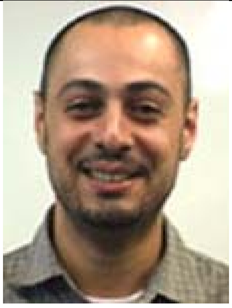
54. Wu H.J. and J.L. Huang, 2004, Automating design requirement acquisition for product knowledge management systems, in proceedings of IEEE International Conference on Systems, Man and Cybernetics, 10-13 Oct, The Hague, Netherlands.

55. Yang C. and H. Li (2008), Study on product knowledge management for product development, in proceedings of the 7th World Congress on Intelligent Control and Automation, 25-27 June, Chongqing, China

56. Zachman J.A, 1987, A Framework for Information Systems Architecture, IBM Systems Journal, Vol. 26, No. 3, 276-292.

57. John Zachman's Concise Definition of the The Zachman Framework™". Zachman International. 2008. http://www.zachmaninternational.com/concise%20definition.pdf.

Vitae

	Dr. Mohamed Zied Ouertani is currently a Postdoctoral Researcher with the Engineering Department, Cambridge University. He is a member of the Institute for Manufacturing and currently leading the research projects "Strategies for Asset Information Management" and "Cambridge Strategic Performance Information Lab", funded by the EPSRC Innovative Manufacturing Research Centre (IMRC). Zied obtained a PhD in computer science in 2007 from Nancy University for a dissertation on product information management support to the engineering change process. His research interests include product lifecycle management, collaborative engineering, engineering change management, engineering asset management and strategic performance management.
	Dr. Salah Baina is Professor Assistant at ENSIAS Engineer School of Computer Science and System Analysis in Rabat, Morocco. He received his PhD on computer science and production engineering in 2006 at Nancy-University. The subject of his PhD thesis was "Model Driven Interoperability: A Product Oriented Approach for Enterprise Systems Interoperability". He also obtained in 2003 an Engineering degree in Computer Science and Applied Mathematics at ENSIMAG, Grenoble France. He is skilled in information systems interoperability and modelling. His research interests are Information Systems Interoperability, Business/IT Alignment and Business Process Modelling and refactoring. He participated in the FP6 INTEROP NoE (Interoperability Research for Networked Enterprises Applications and Software)
	Dr Lilia Gzara is an Associate Professor at Grenoble Institute of Technology, department of Industrial Engineering. She is also member of the G-SCOP laboratory, which is an associate unit of the CNRS (French National Centre for Scientific Research). She received her PhD from the Grenoble Institute of Technology in 2000 and she occupied since then research/teaching positions at Polytechnic Montreal, Nancy University and Grenoble Institute of Technology. She co-directed many PhD and master thesis and participated in five national research projects (funded by the French government). She is member of the design society and the French chapter of the International Council on Systems Engineering. Her research interests include Information systems engineering, Business Process Management, Product Lifecycle Management and Knowledge management in collaborative design context.

	Dr. Gérard Morel is Professor at Nancy University and co-director of CRAN, the Research Centre for Automatic Control of Nancy, which is an associate unit of the CNRS (French National Centre for Scientific Research). He directed about 30 PhD Thesis and Accreditations to Supervise Research. He published over 150 articles in the area of 'Systems and Automation Engineering'. He holds research positions in national and international research networks and serves in several positions in IFAC and with the European Commission. He is currently Europe Editor of the International Journal of Intelligent Manufacturing and Associate Editor of the IFAC International Journal of Real-Time Automation on Engineering Applications of Artificial Intelligence. Gérard acts as an expert for the French Agency for the Evaluation of Research and Higher Education. He is also Director of a master degree on 'Systems Engineering' as well as member of the French chapter of the International Council on Systems Engineering.

	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner		Product Knowledge Traceability Constructs (Description)				
Enterprise Model {conceptual} Business Owner		Product Knowledge Traceability Constructs (Formalization)				
System Model {logical} Designer	Product Knowledge Models - UML Class Diagrams (Design Rationale Based / Holonic-Based Models)					
Technology Model {physical} Implementer		Product Database Schema (Relational Schema)				
Detailed Representation {out-of-context} Subcontractor		Product Database Implementation (SQL)				
Functioning System		Product Database Management System				

Figure 1. A Product knowledge traceability approach

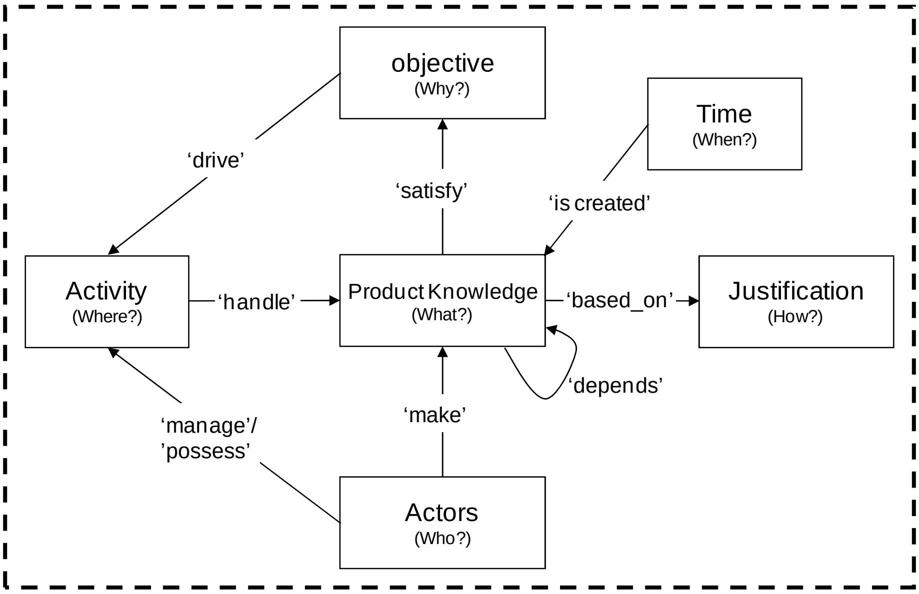


Figure 2. Formalization of product knowledge traceability constructs

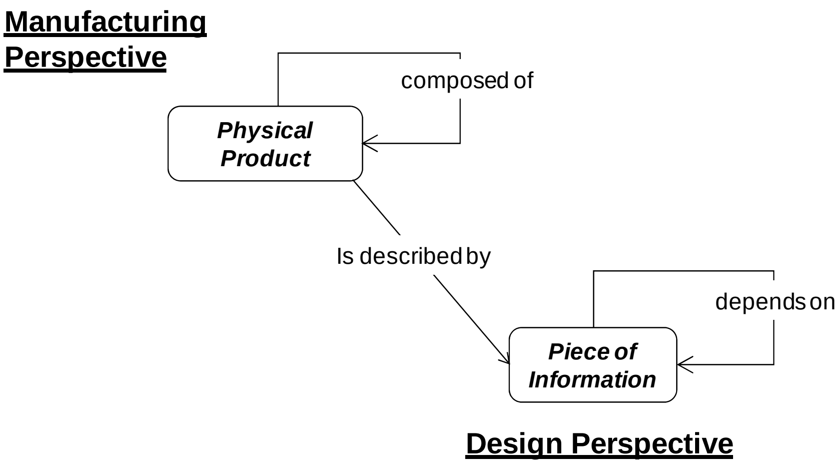


Figure 3. Relationship between design perspective and manufacturing perspective of traceability.

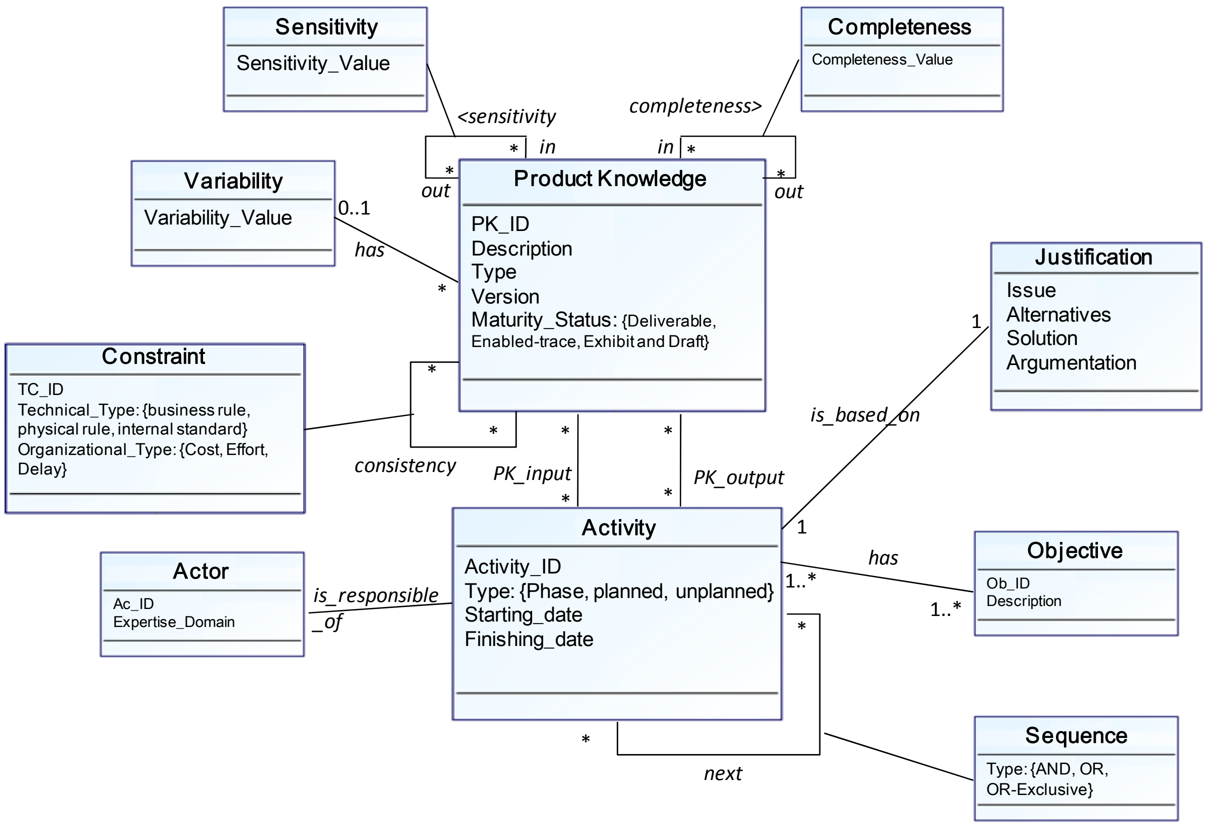


Figure 4. Design rationale model expressed in UML formalism

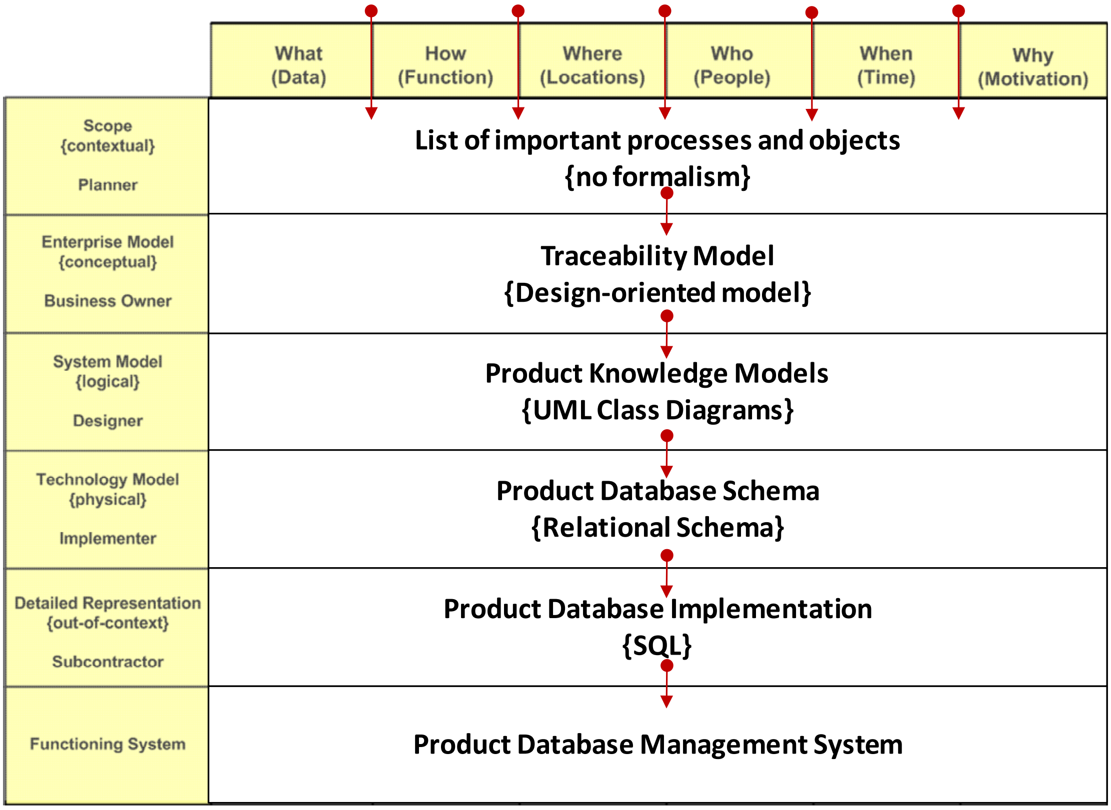


Figure 5. Used strategy and mapping with Zachman Framework in the context of product design

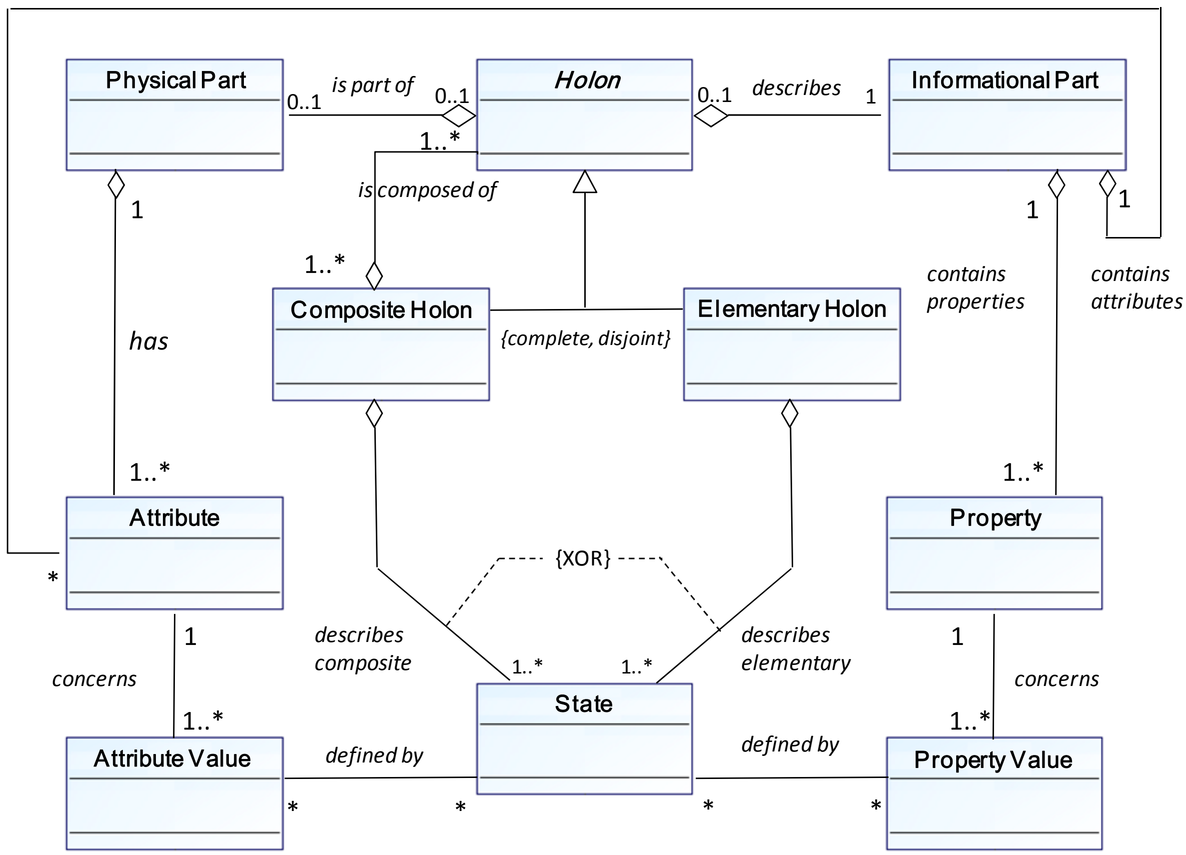


Figure 6. Holon model expressed in UML formalism (Baina, 2006)

	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 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	**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

Figure 7. Used strategy and mapping with the Zachman Framework in the context of product manufacturing

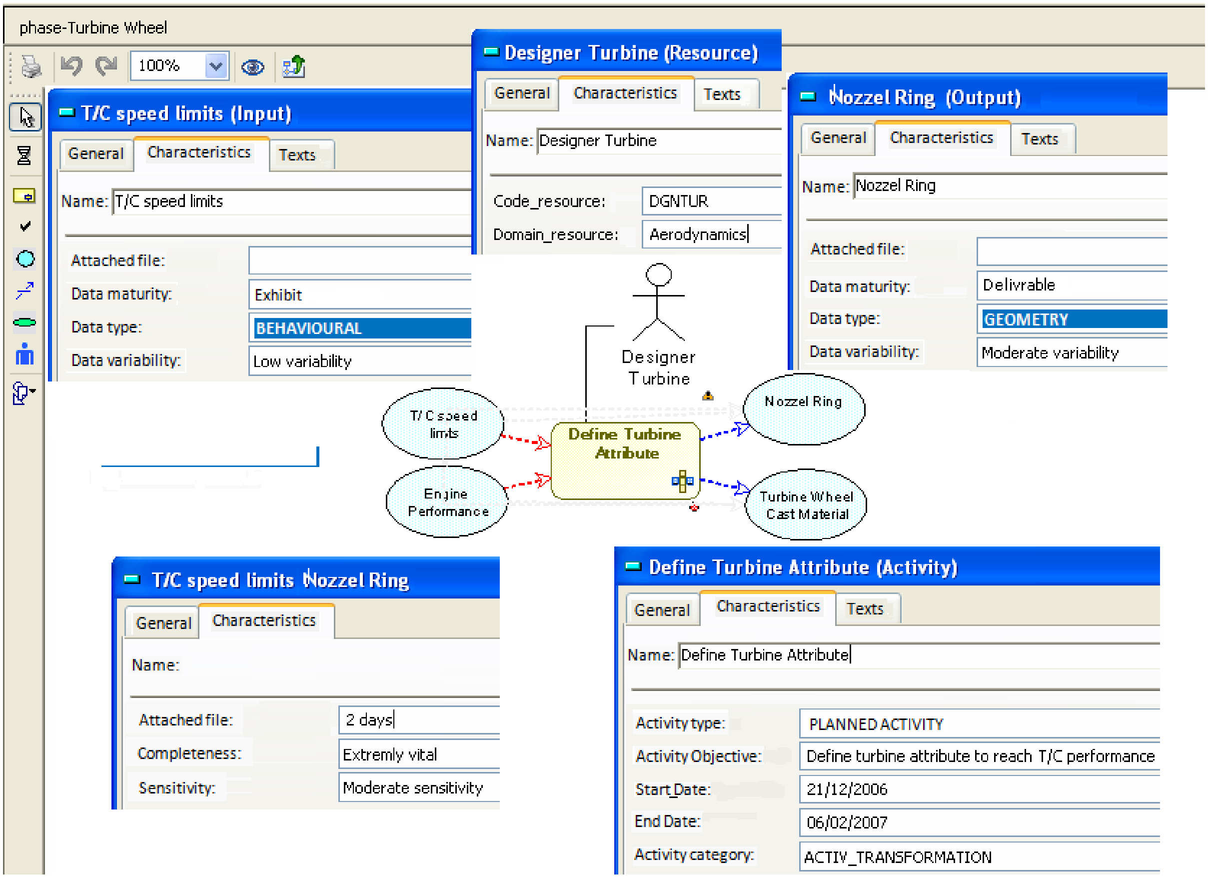


Figure 8. Snapshot of the product knowledge traceability tool

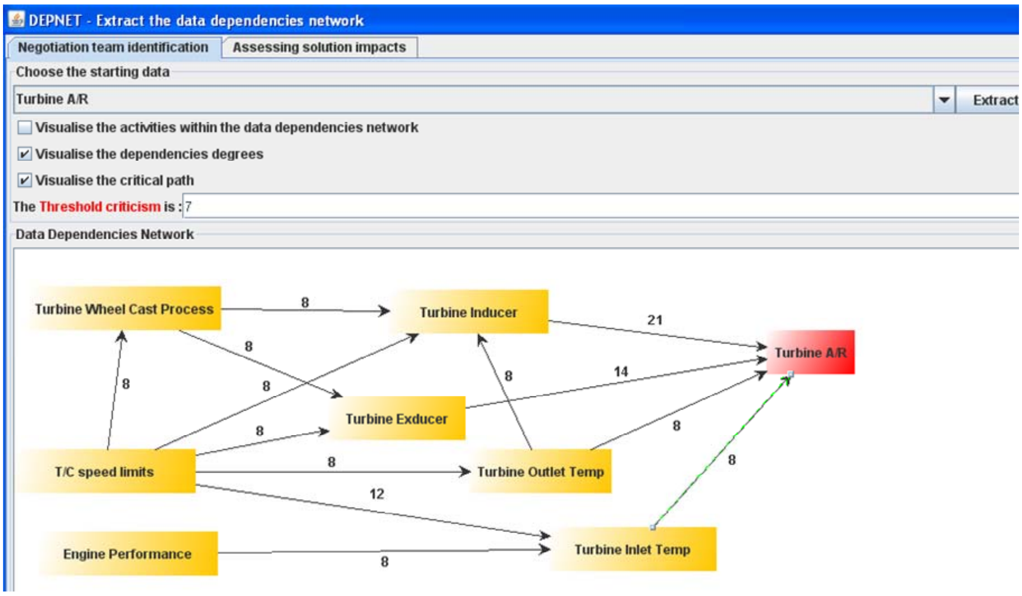


Figure 9. Partial dependency network for the turbine design phase progression

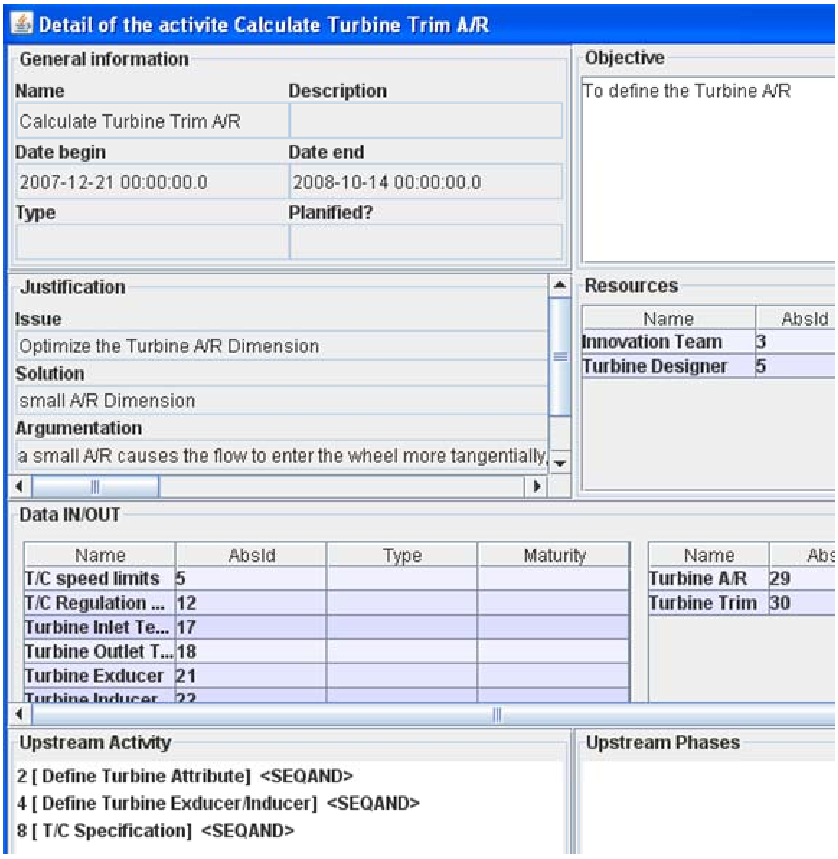


Figure 10. Product knowledge traceability: Turbine A/R definition

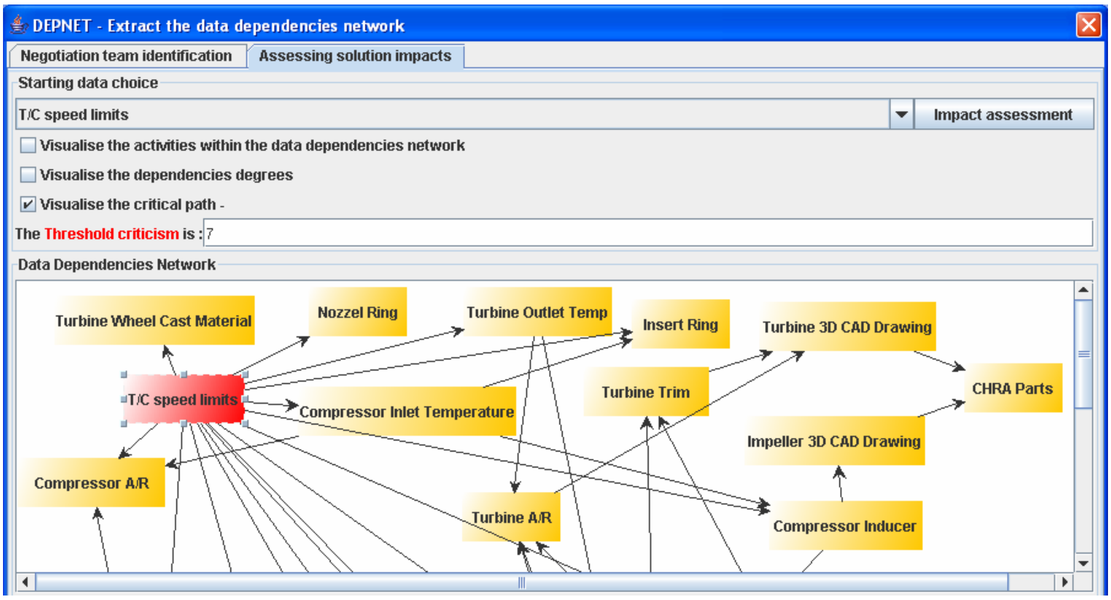


Figure 11. Impact of engineering change propagation

Figure 12. Starting point of products tracking.

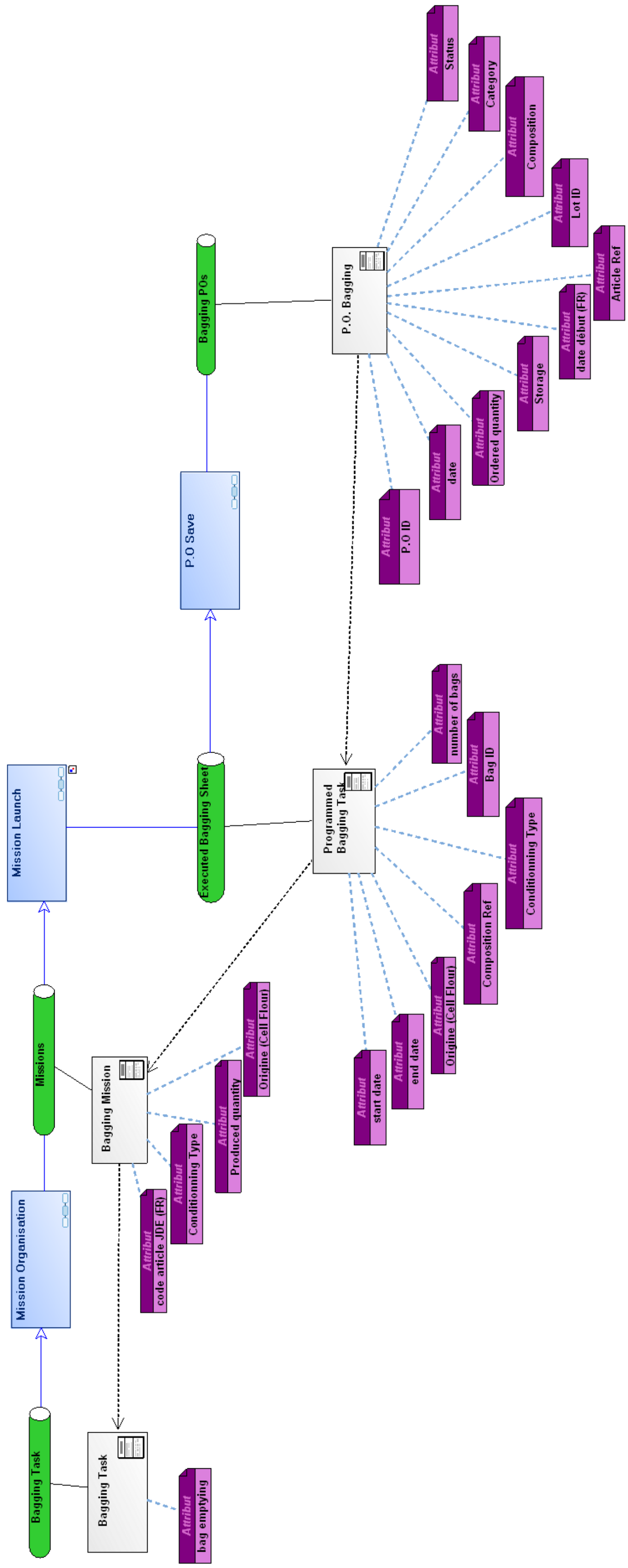


Figure 13. Example of conceptual map for the product oriented approach.

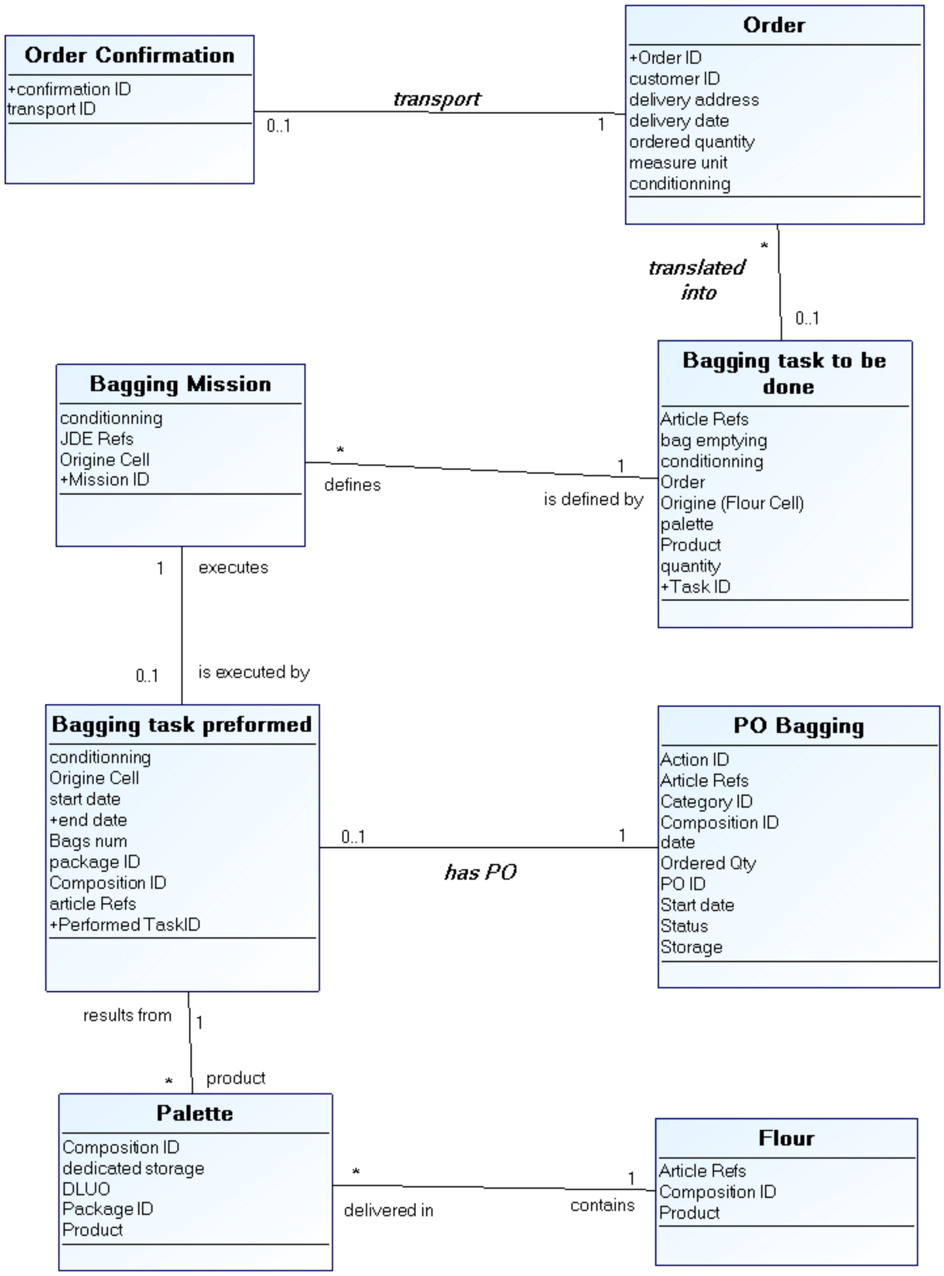


Figure 14. Automatically generated class diagram.

Table 1. Transformation for Holon Models to UML Class Diagrams

Holon Model concepts		**UML Class Diagram concepts**
Holons		UML Classes
Properties		Attributes of classes
Attributes		Attributes of classes
Dependencies between Holons		Associations between classes