



**HAL**  
open science

## A reference framework following a proactive approach for Product Lifecycle Management

Martin Marchetta, Frédérique Mayer, Raymundo Forradellas

► **To cite this version:**

Martin Marchetta, Frédérique Mayer, Raymundo Forradellas. A reference framework following a proactive approach for Product Lifecycle Management. *Computers in Industry*, 2011, 62 (7), pp.672-683. 10.1016/j.compind.2011.04.004 . hal-00598311

**HAL Id: hal-00598311**

**<https://hal.science/hal-00598311v1>**

Submitted on 7 Oct 2011

**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.

Martín G. Marchetta, Frédérique Mayer, Raymundo Q. Forradellas, A reference framework following a proactive-product approach for Product Lifecycle Management, *Computers in Industry* 62 (2011) 672–683

# **A reference framework following a proactive-product approach for Product Lifecycle Management**

Martín G. Marchetta<sup>a,b</sup> Frédérique Mayer<sup>c</sup> Raymundo Q. Forradellas<sup>a</sup>

<sup>a</sup>Logistics Studies and Applications Centre, School of Engineering, National University of Cuyo, Centro Universitario, CC405 (M5500AAT) Mendoza, Argentina

<sup>b</sup>CONICET (Argentina)

<sup>c</sup>Equipe de Recherche sur les Processus Innovatifs, ENSGSI, Institut National Polytechnique de Lorraine, 8 rue Bastien Lepage, BP 90647 (54010) Nancy Cedex, France

## **Abstract**

Product Lifecycle Management (PLM) has been identified as a key concept within manufacturing industries for improving product quality, time-to-market and costs. Previous works on this field are focused on processes, functions and information models, and those aimed at putting more intelligence on products are related to specific parts of the product lifecycle (e.g. supply chain management, shop floor control). Therefore, there is a lack of a holistic approach to PLM, putting more intelligence on products through the complete lifecycle. In this paper, a PLM framework supported by a proactive-product approach based on intelligent agents is proposed. The developed model aims at being a first step toward a reference framework for PLM, and complements past works on both product information and business process models (BPM), by putting proactivity on product's behavior. An example of an instantiation of the reference framework is presented as a case study.

**Keywords:** PLM, Proactive Product, Intelligent Agent, Virtual Enterprise, Concurrent Engineering

## **1 Introduction**

The need to compete at a global level has pushed enterprises to improve their operations and business practices and even to collaborate [1], breaking down physical barriers and creating what are nowadays called virtual enterprises [2, 3]. In this context, Product Lifecycle

Management (PLM) has been considered a key concept in order to achieve efficiency and maintain consistency along the complete product lifecycle, from early stages of the development process of new products to their disposal [4, 5].

PLM involves the management of product information and the integration of business processes [6]. Initially, integration and collaboration were implemented mostly along the supply chain (with suppliers and customers [3]), but currently peer enterprises are also partners [7], as shown in figure 1.

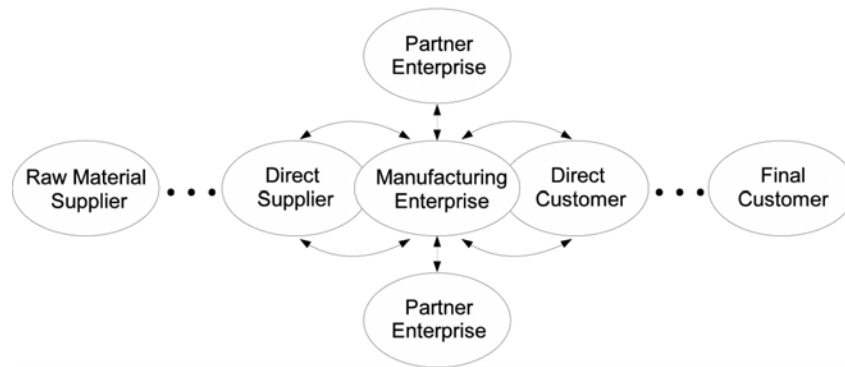


Figure 1: Current peer-to-peer and supply chain enterprise integration schema

Partner enterprises share much information concurrently. Additionally, product information now needs to be shared across heterogeneous systems. Since business processes go beyond a single enterprise, they involve people, information and other resources with different ownership and cultures.

Previous research works have focused on information models, business processes integration and interoperability issues. Only recently, the need for putting more intelligence on products making them proactive [8, 9] during their own development and management, to better support business processes, has been recognized. Additionally, there is lack of a holistic and generic (i.e. independent of the industry/activity) treatment of PLM aspects, including business processes definition, product information models and information exploitation approaches [5].

In this paper, a framework based on a proactive product approach to PLM is proposed. The model aims at being a first step toward a reference PLM framework, rather than a model for a specific application or industry. The framework includes a Business Process

Model (BPM), a Product Information Model (PIM) and an architecture of applications based on intelligent agents for information exploitation.

## **2 Related research works**

At least three main topics have been researched in PLM: Business Process Models (BPM), Product Information Models (PIM) and product information exploitation for supporting business processes.

### **2.1 PLM business process modeling**

Integration of product information, project management, processes and procedures has been addressed through a product digital mock-up in [10]. Collaborative frameworks and technological requirements for their implementation have also been discussed [11]. In this context, usefulness of enterprise modeling and reference models to achieve enterprise interoperability and collaboration has been highlighted [12].

Both technical (i.e. low-level communication) and information/knowledge interoperability issues have also been studied [13]. Centralized approaches concentrating PLM information, accessible through communication infrastructures for supporting collaboration in the supply chain, have also been proposed [14].

The efficiency vs. innovation dilemma in PLM has been addressed [15], by proposing intensive use of Knowledge Management (KM) to involve buyers and suppliers in innovation processes. There is a need for collaborative innovation management within PLM, which can be achieved by integrating collaboration, product development and innovation by means of a technology framework [6].

### **2.2 Product information modeling**

PLM involves management of product information along its complete lifecycle. Several aspects of product models have been researched, including handling of evolution of documents in a Product Data Management system [16], a product data framework for logistics planning activities [17], and an ontology based on the Core Product Model (CPM) of the National Institute of Standards and Technology (NIST) [18, 19].

In [20], the information flows and the relevant existing standards within PLM were evaluated, highlighting current issues regarding interoperability and information exchange. Model-driven approaches for achieving interoperability have also been proposed [9]. This kind of interoperability, called “product-oriented interoperability”, is based on

a meta-model of the product representation, and a mechanism for make a mapping between it and specific models.

### **2.3 Product information exploitation for business process integration**

In previous works, exploitation of information has been applied to different activities throughout the product lifecycle. Support to individual activities (e.g. CAPP) within a single enterprise has been proposed [21]. Extension of this schema for supporting collaboration among customers, suppliers and partner enterprises was also suggested [2]. Change propagation within a single company [22], as well as for maintaining consistency of design documents between partners [23], were also proposed.

The idea of intelligent products is to connect the physical products with their counterparts within information systems. In this context, classification dimensions for product intelligence, application domains and obtainable benefits have been proposed [24]. The application of the “holon” concept was suggested in [25]. Intelligent agents, RFID and other data acquisition technologies were applied to intelligent products in post-sales phases in [26].

The use of different kinds of agent technologies (holons, multi-agent systems, etc.), to support different aspects of PLM (control, production management, etc.) has been explored [27]. In [28] composition of services through semantic interoperability was suggested.

Multi-agent systems have been applied to distributed planning along the supply chain [29], project coordination and monitoring [30] and coordination of engineering activities in new product development projects [31]. Creation of instant virtual enterprises and their optimization have also been supported by the use of agents [32, 33].

## **3 Business process model for PLM and SCM**

Traditionally, within PLM the product has concentrated all the data required by different activities, and the approaches considering products as active entities have been restricted in different ways (e.g. focusing on specific parts of the product’s lifecycle). Thus, these systems are able to answer users’ questions regarding the product reactively, and they can implement limited proactive behaviors. Moreover, despite the variety of works on information exploitation, there is a lack of an application-independent framework. Additionally,

PLM business processes have also been researched, but there has been a separation between BPM and information exploitation.

In this section, a business process model for PLM and SCM is presented. Special emphasis is put on the new product development business process, through a detailed model of its activities. Additionally, a taxonomy of interactions is described, which highlights the different kinds of information exchange and assistance opportunities that exist in the described scenario.

### 3.1 Business process model

Several processes must be carried out during a product's lifecycle. Figure 2 depicts an integrated business process model, including development of new products, production and logistics processes<sup>1</sup>.

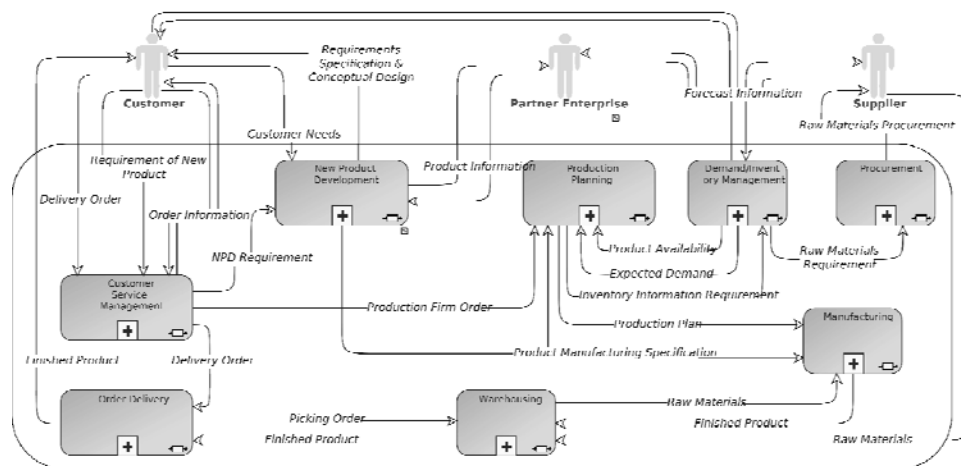


Figure 2: Overview of Business Process Model for PLM including SCM and Product Development

The model is based on two complementary views of business processes in logistics [3, 34]. Processes of a product development/manufacturing company are shown inside the rectangle. The supplier, the customer and the partner enterprises are external actors. Partners are enterprises which design other components, or that manufacture products designed by the company. Two types of customer orders may be received: for development new products and for existing products. Additionally, at any moment the customer may ask for information about its order's

<sup>1</sup> The models in this article were created with the MEGA suite

state. All these information flows are handled by the Customer Service Management process (called Customer Response in [34]).

The Order Delivery process takes care of coordinating all activities to put products in the location accorded with the customer. Warehousing activities includes picking, put away, storage, etc. The New Product Development (NPD) process is concerned with turning customer needs into requirement specifications, designing the product, creating manufacturing specifications, etc. This process handles consistency and coordination both inside the company and with the external actors.

The Production Planning process accommodates the demand to the available manufacturing resources. The result is a production plan which is provided to Manufacturing. Demand/Inventory Management forecasts demand in order to support procurement and production planning. Procurement handles the buying orders. Finally, the Manufacturing process is in charge of products creation using manufacturing resources.

### 3.2 New Product Development process

The New Product Development process involves a great number of interactions of different types (along the supply chain, with partner enterprises, and also inside the process itself). Figure 3 presents a functional diagram for this process.

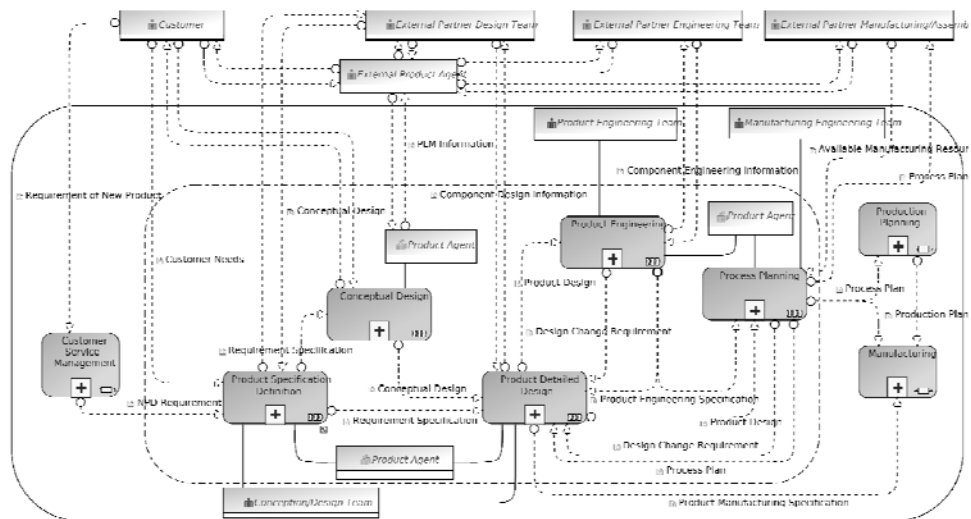


Figure 3: Functional diagram of the Individualized Product Definition

business process

The process is composed of 5 main activities: Product Specification Definition, Conceptual Design, Product Detailed Design, Product Engineering and Process Planning. In a concurrent engineering environment, these activities are carried out in an overlapped way, rather than sequentially.

### **3.2.1 Activities within New Product Development**

The objective of the Product Specification Definition activity is to turn customer needs into a product specification. Customer needs represent the “problems” the product must solve, e.g. functional features, durability, etc, while a product’s requirement specification is a formalized specification of customers’ requirements, considering also performance, cost constraints. The product requirement specification is created by the Conception/Design Team, in collaboration with the partner enterprise’s design team and the customer.

The Conceptual Design activity defines the first product’s features. It also involves participation of customers, in order to assure that the product matches the needs. Product Detailed Design defines the main features of the product regarding its form, geometry and proposed materials. Detailed design creates a mapping from design features to requirements, including the rationale behind design decisions. Exchanging information relevant to all the partners is a good practice in order to avoid later problems during downstream activities. The resulting design undergoes engineering evaluations (Product Engineering), and it is also used by Process Planning. After the final design is obtained, it contains information that allows Manufacturing to produce the product.

The Product Engineering activity assures the accomplishment of engineering constraints by the product design (i.e. determining if geometry, structure, materials are able to cope with stress, durability, efficiency, etc.). Process Planning is the definition of manufacturing and assembly operations needed to produce one instance of each part, their sequence, along with the machines, tools and fixturing required for these tasks [35].

If design changes are required as a result of these activities, these modifications are sent to the internal design team, which in turn makes decisions and articulates change impacts with corresponding internal and external affected stakeholders.

In figure 3, the product is shown as an actor (Product Agent entity),



because it interacts with other products (in partner companies), as well as with people, in a proactive way, detecting inconsistencies, propagating changes and giving automated assistance when relevant events arise.

### 3.3 Taxonomy of product-automated support along product lifecycle

PLM in a concurrent engineering/virtual enterprise paradigm involves interactions between several types of entities and at different abstraction levels. Since each interaction type entails different issues (e.g. information ownership, concurrency, etc.), a structured categorization of these interactions is needed.

Both the “business processes” and the “virtual enterprise” dimensions are considered here for the classification. As a result, a taxonomy containing four interaction types, depicted in figure 4, is proposed here as follows:

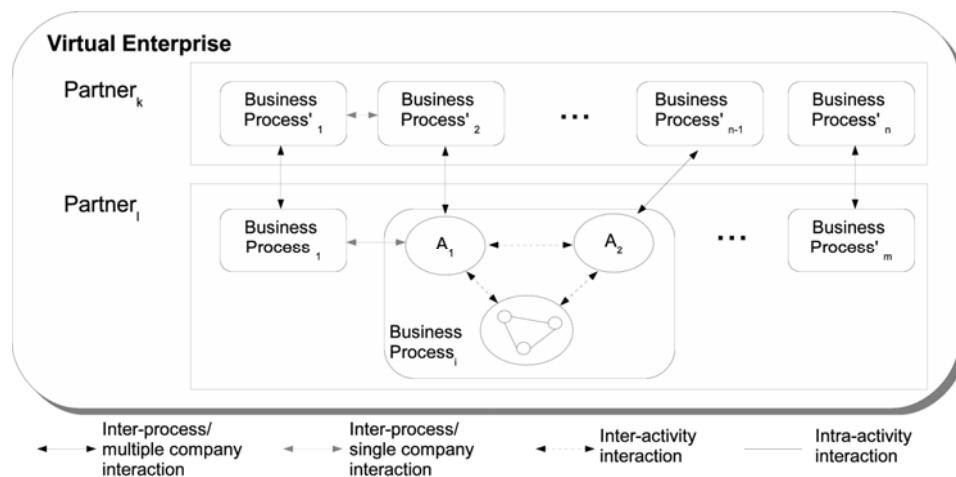


Figure 4: Hierarchy of active-product's interventions along its lifecycle

- 1 Inter-process/multiple company interactions: It involves information exchanges, coordination and information exploitation between processes in different companies within the virtual enterprise. Information ownership as well as high concurrency must be taken into account, since interactions cross a single enterprise barriers.
- 2 Inter-process/single company interactions: It is related to interactions among business processes within a single company.

- 3 Inter-activity interactions: Inside a single business process, there exists the need for coordinating activities which contribute to its value chain. This is especially true when activities whose results impact on the others are executed concurrently.
- 4 Intra-activity interactions: Within a single activity the number of people involved may be very small, and so the information exchanged; thus, the most important thing an active product may contribute with, is exploitation of information in order to optimize results of that activity.

Within our framework, the active product is similar to an expert advisor who actively integrates all the information across the product's lifecycle, resulting in a PLM system of PLM inter and intra systems [36]. For inter-process situations (types 1 and 2), the most important benefits include change impact analysis, propagation/notification of changes to (and only to) relevant people, global optimization, risk assessment, feedback on project evolution, etc.

Considering inter-activity interactions (type 3), similar benefits can be obtained, but restricted to activities within a single business process. Finally, inside a single activity (type 4), local multi-objective optimization can be supported by the product, as well as other aids such as know-how acquisition through machine learning [37], plan recognition for identifying user's intentions, task completion, automated generation of alternative solutions [38], etc.

## **4 Applications architecture**

Software applications that support product development have emerged independently from the needs of individual activities (CAD, CAM, scheduling, etc). This has generated issues related to interoperability, data formats, redundant information exchanges, much rework for interconnecting systems and lack of coordination and collaboration support. Figure 5 shows a typical applications architecture within virtual enterprises.

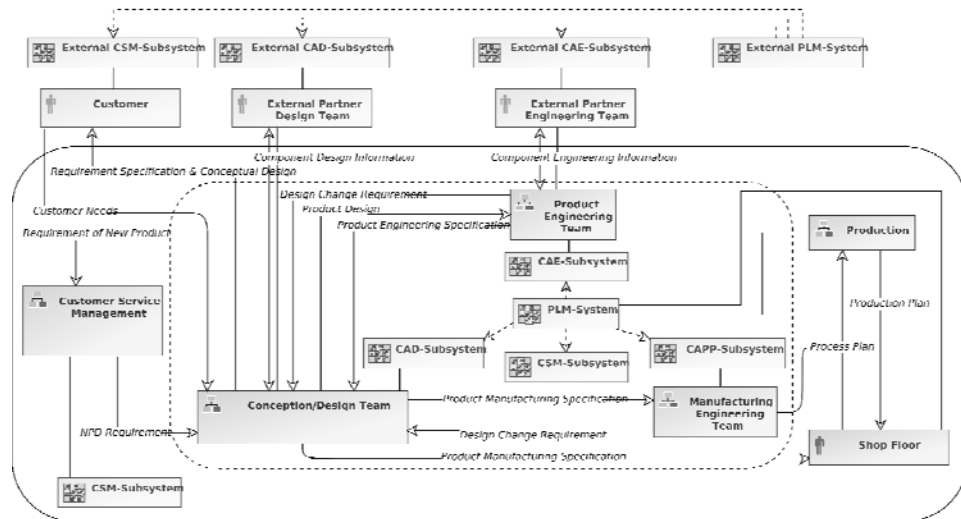


Figure 5: IT applications architecture typically used within virtual enterprises for product development

Interoperability between existing applications has turned to be a key issue in PLM research [39], because companies have made investments in solutions to individual problems, and partners are likely to use different software solutions. A way in which PLM systems provide interoperability is by being an information backbone linking applications. The problem is that much information must be converted to other formats in order to pass it from one system to another, which produces some semantic information loss. Additionally, a “pull strategy” is used for product information exploitation (reactive system), and coordination is made manually.

Figure 6 depicts the global IT architecture of our framework. For simplicity, only entities relevant to the NPD process are shown. The architecture aims at providing a structured integration infrastructure to:

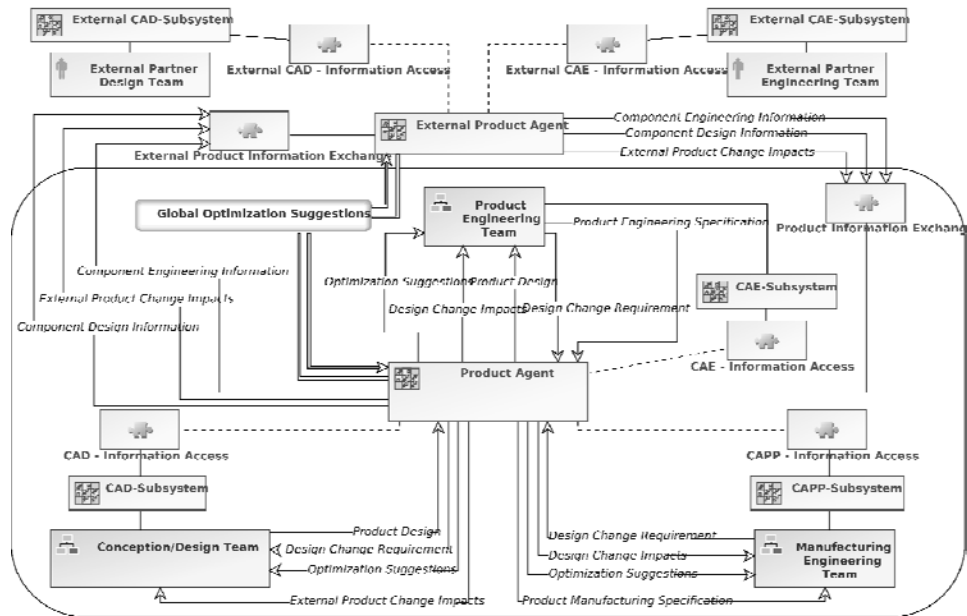


Figure 6: Proposed IT applications architecture for supporting active-products

- 1 Support development of applications with proactive behaviors.
- 2 Support enhanced interoperability between heterogeneous systems.
- 3 Reduce information exchanged in non-structured formats.
- 4 Improve collaboration and coordination.

The model is intended to support automated reason on design and manufacturing information, make improvement suggestions, identify impact of distributed changes and support global optimization. Systems (CAD, CAPP, ERP, etc.) can be physically separated and have different owners. Within our model coordination, collaboration and information exchanges are supported through interfaces exposed as services.

In a distributed environment, there is no single system having all the relevant product information. Thus, global optimization can be achieved by means of collaborative optimization techniques. In our framework, the product is at the core of its lifecycle management, having both reactive and proactive behaviors. Therefore, it is capable of identifying opportunities to be exploited and to suggest how capitalize them. The Product Agent of figure 6 is an Intelligent Agent, which plays the role of an active product. Its environment is composed

of both reactive (e.g. other applications) and proactive entities (e.g. human users or other artificial agents) in PLM.

The Product Agent acts as an automated expert connected to all the applications supporting PLM activities, which is capable of identifying events that take place on the environment and to act as a consequence. It has also the ability to communicate with other agents when not enough information is available to make (or suggest) decisions. This also allows managing information property issues, since there is no single agent having all the information but a set of agents, each one having access to a part of it, which communicate with each other to exchange data.

Thus, the whole PLM system moves from a set of isolated automation islands toward a set of systems which interact with each other. In our framework, the notion of system of systems is considered as a potential artifact [40] in order to design a PLM system having proactive behaviors, supported by the underlying applications, information models and processes. Such a system must be able to autonomously exploit information scattered across the virtual enterprise and to communicate with other autonomous and non-autonomous systems representing other's interests and intentions. Therefore, intelligent agents are a suitable technology for implementing the framework, since they entail 4 properties [41]: autonomy, social ability, reactivity and proactivity.

## **5 Product information model to support PLM integration**

A suitable product information model is the structure on which automated and semi-automated support to business processes is based. It contains product metadata defining the semantics of each product aspect to be registered within PLM databases. A product ontology is useful for identifying relevant data to be recorded during the product's lifecycle, it allows automated processing of product information and enables the interoperability with other systems.

It is very difficult to include every single piece of information concerning a product in all cases, due to the huge range of existing product types and industries. In this work, an extension of the model proposed in [18, 19] is presented. Relationships added to the original model are shown in figures with thick black lines. The model reported in [18, 19] includes information about product's structure, but it does not consider other aspects of PLM, such as logistics, process planning or manufacturing, aspects covered by two additional models, namely

the Process Planning and Logistics models.

## 5.1 Extensions to the Core Product Model

Figure 7 shows a UML class diagram representing an ontology for the core part of the information model. The main class is **Artifact**, which represents a product or component. An artifact has **Form** and **Function**, which define some of its physical (**Material** and **Geometry**) and behavioral properties. The artifact is also composed of a number of **Features**, which represent engineering characteristics associated to portions of the product, and that may include function, form and other aspects (e.g. design rationale through the mapping between features and requirements). Mapping to requirements is made through the relationship between the **CoreProperty** and the **Specification** classes.

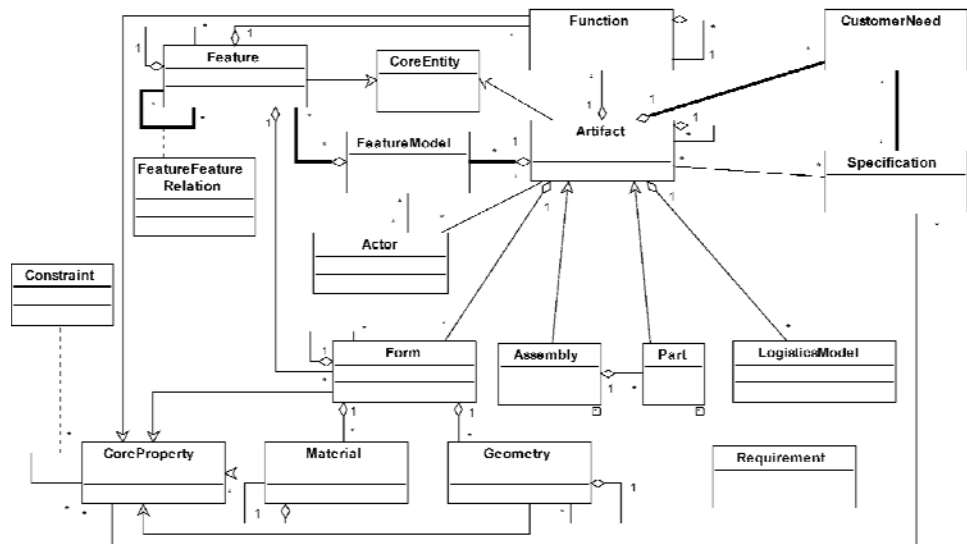


Figure 7: Core Product Model and proposed modifications

The most important change made to this portion of the model is the inclusion of a **FeatureModel**. The original model associated **Feature** classes directly to the product. However, usually different activities have different views of the same product information, and each one of these points of view is represented by one or more feature models, so features are grouped according to the particular view they belong to. The consideration of feature models as explicit entities allows providing the most suitable model to each user or team, supports the assessment of change impacts and it enables automated reasoning for

several purposes (e.g. improvement suggestions, identification of cross-activity optimization opportunities, etc).

Another addition is that features may also be related to each other (e.g. dependency, mapping between different feature models, etc), through the **FeatureFeatureRelation** class. This class enables many information exploitation opportunities, since it allows linking features belonging to different feature models. Thus, different teams in charge of different activities may be coordinated, notified (of changes) and advised (about optimization opportunities), as the product development advances. The **CustomerNeed** entity was also added, in order to model needs as expressed by the customer.

**Assembly**, **Part** and **LogisticsModel** classes are further presented in following sections. The CPM presented here contains only the portion related to the changes we introduced (for further information see [18]).

## 5.2 Proposed extended Open Assembly Model

The assembly model extends the core product model by adding more information about how components and parts are assembled into subassemblies and final products. Figure 8 shows part of the NIST Open Assembly Model [18], and some modifications we introduced.

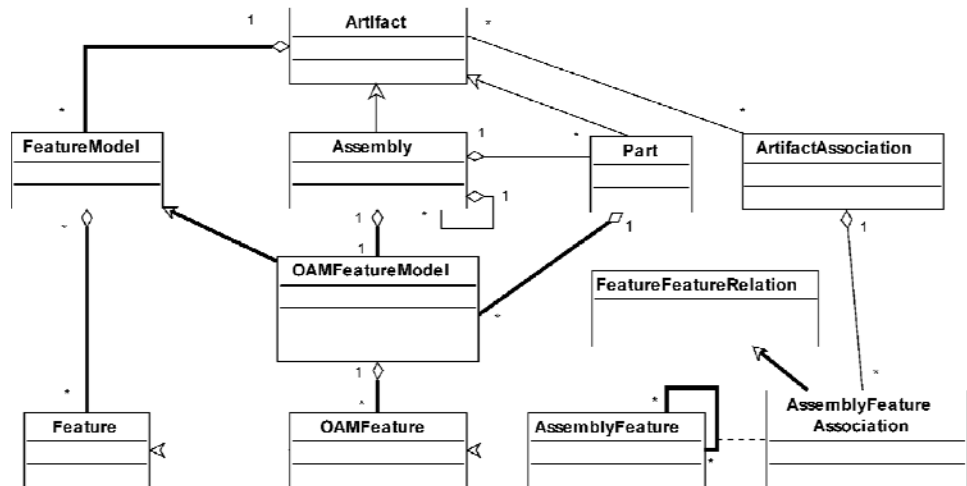


Figure 8: Assembly Model and proposed extensions

As in the product model presented in the previous section, the most important modification is the addition of the **OAMFeatureModel**

class, a subtype of **FeatureModel**, that groups features. **OAMFeature** is a subclass of **Feature**, which shows how a component’s feature model represents the particular point of view of assembly.

In the original model, **AssemblyFeature** classes were not directly associated with each other, but as a transitive relation with **AssemblyFeatureAssociation**. In our model, **AssemblyFeatureAssociation** is a subtype of **FeatureFeatureRelation**. This small modification has the advantage of preserving the same general structure defined in the core product model, customizing it across the different specific models.

### 5.3 Manufacturing Process Planning Model

Figure 9 shows the Manufacturing Process Planning Model. For making a product, non-decomposable parts are manufactured separately and then (if needed) they are assembled. Hence only the **Part** class (a “monolithic” product) is directly “manufacturable”.

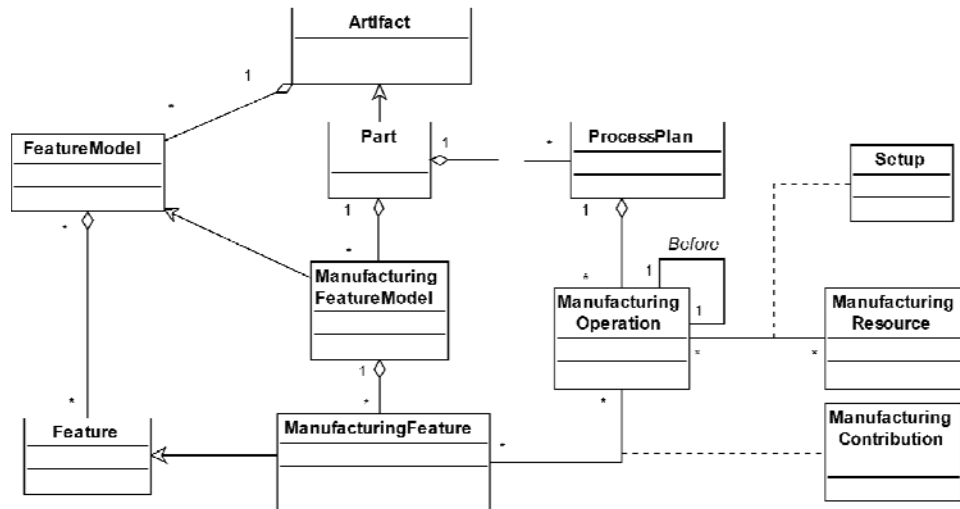


Figure 9: Manufacturing Process Planning information model

A part has one or more **ManufacturingFeatureModel**, which are specializations of the generic **FeatureModel**. Each manufacturing feature model is composed of **ManufacturingFeature**, a subtype of **Feature** which conveys manufacturing aspects (e.g. tolerances, finishing qualities, etc). Each part has one or more associated **ProcessPlan**, including manufacturing processes, their sequence, tools and machines to be used, etc. **ManufacturingOperation** represents



individual manufacturing operations, related to each other by restrictions (e.g. precedence).

Each operation uses a set of resources (tools, machines, etc.), represented by **ManufacturingResource**. Manufacturing resources, in turn, have configuration parameters represented by **Setup**. A single process may not completely produce a feature, either its geometry or its finishing characteristics. Thus, each manufacturing operation is associated to a feature by means of a **ManufacturingContribution**, which represents the effect of applying the process to produce the feature.

## 5.4 Logistics model

Figure 10 shows a basic logistics model, part of our framework. This model allows the product agent to support logistics activities, such as coordinating supply chain activities, determining lead times and costs, computing raw materials needed for production, etc. An artifact has one or more **LogisticsModel**, which conveys all this information. The product may have more than one model since it may have several assembly alternatives, or it can be produced in different ways (e.g. buying some components instead of making them).

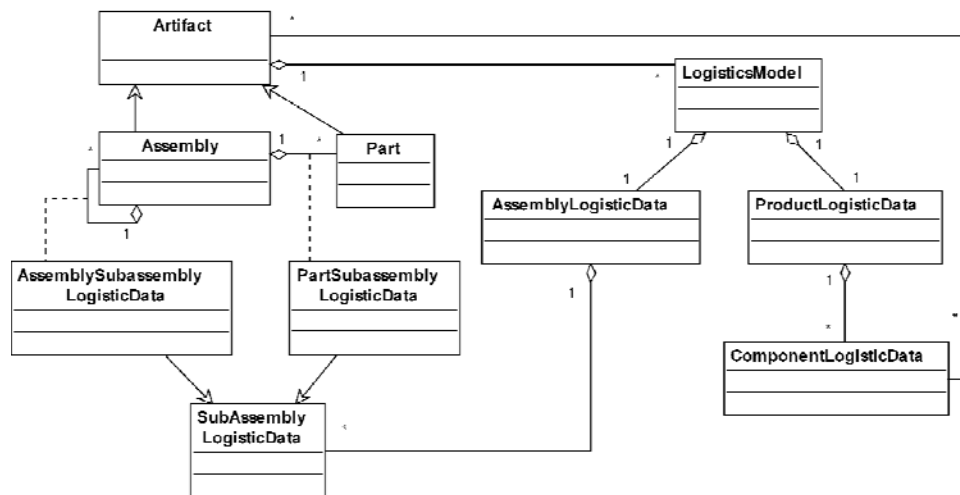


Figure 10: Product Logistics information model

Each logistics model is composed of **AssemblyLogisticData** and **ProductLogisticData**. The former aggregates data related to assemblies and subassemblies and the latter is composed of logistics

data of a single artifact (either a part or a subassembly). **AssemblySubassemblyLogisticData** represents information for the relationship of an assembled product with its subassemblies, and **PartSubassemblyLogisticData** for the relationship between a monolithic part and a subassembly. These data include the Bill of Materials (BOM), costs of assembly operations, lead times, etc.

**ProductLogisticData** and **ComponentLogisticData** include logistics information about a single product, assembly or part, as opposed to the previously mentioned classes which apply to relationships between components. Information considered in this case includes component's total cost, total lead time, warehousing information (component's volume, weight and security stocks), quantities available in the warehouse, production state, etc. The model is intended to be generic, so any relevant data may be added according to the needs of a specific application through subtypes of **ProductLogisticData** and **ComponentLogisticData**.

## 6 Case study

In order to illustrate how the described framework may be applied to specific applications and industries, a particular instantiation for supporting change propagation in the automotive industry is presented as a case study. This case aims at exemplifying the use of the general framework described in this paper, but the framework may be used in other applications and industries.

Consider the product shown in figure 11.a, which is complex in itself, and also because it interacts with other components (e.g. a clutch, an engine, etc). In this example, we assume that some components (shifter and cover, figure 11.b) and the internal mechanism of the product (figure 11.c) are designed by teams belonging to different (partner) companies. In addition to designers, another actor involved in the development of the product is the process planner, who is in charge of creating the manufacturing plan for each part.

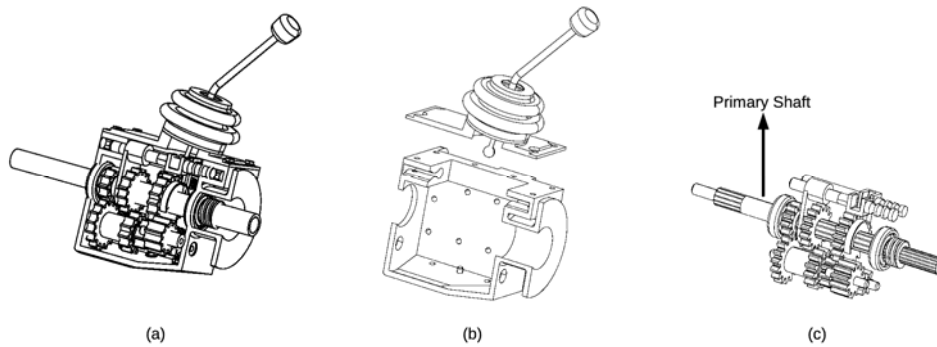


Figure 11: Solid model of a complex product (gearbox), and its components

Considering a concurrent engineering environment, both design teams and the process planner would work in parallel. In this schema, changes introduced by either actor are likely to have impacts on the artifacts handled by the others. Figure 12 shows a collaboration diagram of typical design/process planning interactions within a virtual enterprise and a concurrent engineering setting.

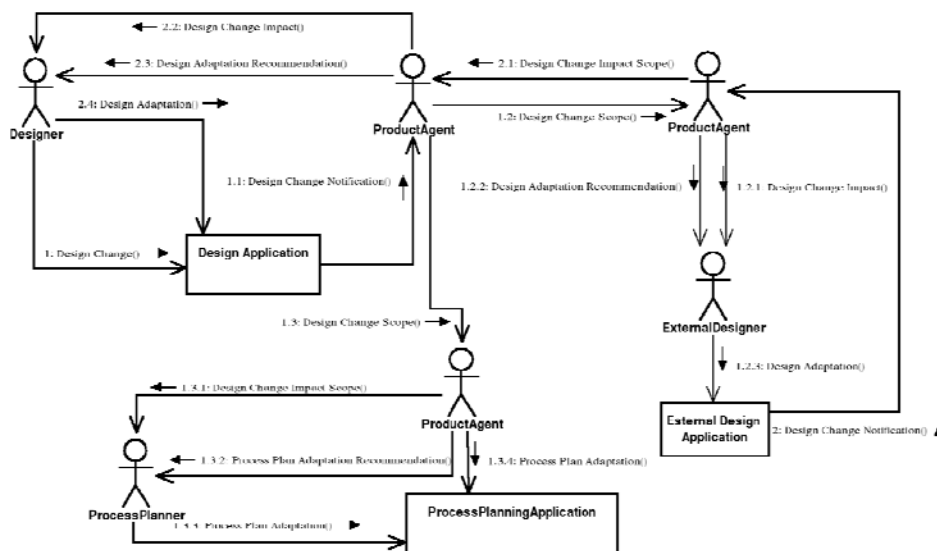


Figure 12: Change propagation collaboration within a virtual enterprise/concurrent engineering environment

Each (human) actor or team is assisted by an instance of the product agent, which is also modeled as an actor. Product agents may reside in

different companies, and may communicate with each other through network links between partners. In such a setting, these agents have proactive capabilities which reduce the complexity and number of tasks that must be performed by humans in different situations (e.g. when a design change is introduced). For example, if a designer changes the diameter of the primary shaft, changes on the cover will be needed (that must be handled by another designer), as well as on the gears attached to it (which will affect the process plans of these components).

In this situation, the group of messages number 1 starts when the designer changes the primary shaft. This is detected by the corresponding product agent (message 1.1), who can then assess the impact of changes, going from the modified entities to the affected ones (gears, cover, etc), and determining the actors involved (external designer, process planner, etc).

In this way, these actors may be notified of the changes through their corresponding agent instances (messages 1.2, 1.2.1, 1.3 and 1.3.1). These latter agents, in turn, may carry out further assistances exploiting additional information of the specific task. For example, the product agent of the external designer may provide not only the impact of changes, but also some advice on how to overcome the inconsistencies they may have caused. Similarly, the agent assisting the process planner may propose an alternative process plan considering the new state of the product's design, using patterns learned from past experiences. According to user preferences, these change proposals may be automatically executed (message 1.3.4), or just suggested (message 1.3.2). Finally, design adaptations introduced by the external design team may have the same effect on the other one, yielding more interactions in the opposite sense (group of messages number 2).

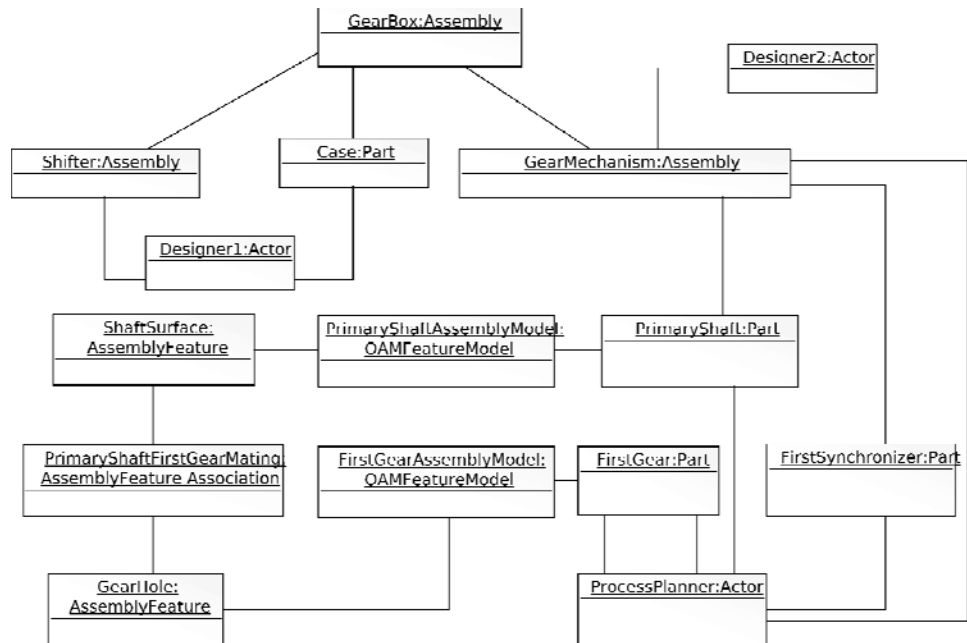


Figure 13: Partial instance of the Product Information Model for the gearbox

Figure 13 shows a partial instantiation of the core product and assembly models of the PIM. This model allows product agent instances to assess the impact of changes, and determine the people and other agents affected by these changes. The gearbox is modeled as an assembly, composed of several parts and subassemblies. The cover is a Part since it is manufactured as a single piece, while the shifter and the internal mechanism are composed of several parts. For simplicity reasons only a few entities and relations are shown in the figure. Actors may be associated to parts or assemblies, which allows identifying people responsible for each component. Three actors are shown in the PIM instance model, namely Designer1, Designer2 and ProcessPlanner.

As shown in figure 13, the PrimaryShaft (a Part) has an assembly feature model (OAMFeatureModel), which in turn contains an AssemblyFeature (ShaftSurface). Similarly, the FirstGear part has its own assembly feature model (FirstGearAssemblyModel) containing the GearHole feature. These assembly features (ShaftSurface and GearHole) are in turn related by a mating relation, called PrimaryShaftFirstGearMating. When the primary shaft is changed, the

product agent can navigate the model through assembly relations in order to find other components (the FirstGear) and people (the ProcessPlanner actor) affected. In a complete PIM there would be many of these relations between entities (figure **Erreur ! Source du renvoi introuvable.** is a small partial example). Thus, information may be propagated to the relevant people and actions may be carried out by both humans and intelligent agents for handling these changes.

## **7 Conclusions and Future work**

In this paper, a first step towards a reference framework for PLM, including a business process model, a product information model and an architecture of applications based on modern technologies, particularly intelligent agents, was presented. The framework aims at providing a suitable model to expand the participation of the product along its complete lifecycle.

One contribution of this work is the integrated treatment of the virtual enterprise setting seen from the product development stages, and the Supply Chain Management integration. Another contribution is the holistic treatment of the three PLM aspects mentioned above. Additionally, the framework includes a generic applications architecture for PLM based on an intelligent product agent, which may be instantiated for specific applications. In addition to that, a hierarchy of interactions that an active product agent must face was also identified. As far as we know, such a structured hierarchy has not been previously defined. An extension of the product information model presented in past works was also presented, including two additional models, namely process planning and logistics.

Future work will include a refinement on the business process model. Additionally, application of the same active-product concept to previous and later stages along manufacturing and logistics administration will be explored, including coordination of production in the shop floor, storage, transportation, procurement, etc. In this approach a product would have an immaterial existence (product knowledge), and a material one (physical product). Finally, a refinement on the applications architecture is needed for assuring the interoperability and product-independence properties of the framework.

## **Acknowledgements**

This research was jointly developed by the School of Engineering (UNCuyo, Argentina), National Council on Scientific and Technical

Research (CONICET, Argentina) and École Nationale Supérieure en Génie de Systèmes Industriels (INPL, France), and partially financed by the PREMIER F-599, ARFITEC ARF-08-04 programs, and CONICET PhD fellowship.

## References

## References

- [1] S. Nof, Collaborative control theory for e-work, e-production, and e-service, *Annual Reviews in Control* 31 (2) (2007) 281–292.
- [2] X. Ming, J. Yan, X. Wang, S. Li, W. Lu, Q. Peng, Y. Ma, Collaborative process planning and manufacturing in product lifecycle management, *Computers in Industry* 59 (2008) 154–166.
- [3] K. Croxton, S. Garcia-Dastugue, D. Lambert, D. Rogers, The supply chain management processes, *The International Journal of Logistics Management* 12 (2) (2001) 13–36.
- [4] A. Saaksvuori, A. Immonen, *Product Lifecycle Management*, 2nd Edition, Springer Berlin, 2005.
- [5] G. Schuh, H. Rozenfeld, D. Assmus, E. Zancul, Process oriented framework to support plm implementation, *Computers in Industry* 59 (2008) 210–218.
- [6] A. Sharma, Collaborative product innovation: integrating elements of cpi via plm framework, *Computer-Aided Design* 37 (2005) 1425–1434.
- [7] M. P. Bhandarkar, R. Nagi, Step-based feature extraction from step geometry for agile manufacturing, *Computers in Industry* 41 (2000) 3–24.
- [8] G. Morel, P. Valckenaers, J. Faure, C. E. Pereira, C. Diedrich, Manufacturing plant control challenges and issues, *Control Engineering Practice* 15 (11) (2007) 1321–1331.
- [9] S. Bañna, H. Panetto, G. Morel, New paradigms for a product oriented modelling: Case study for traceability, *Computers in Industry* 60 (2009) 172–183.
- [10] F. Danesi, N. Gardan, Y. Gardan, M. Reimeringer, P4lm: A methodology for product lifecycle management, *Computers in Industry* 59 (2008) 304–317.
- [11] A. Osório, L. Camarinha-Matos, Distributed process execution in collaborative networks, *Robotics and Computer-Integrated Manufacturing* 24 (2008) 647–655.
- [12] H. Panetto, A. Molina, Enterprise integration and interoperability in manufacturing systems: Trends and issues, *Computers in*

- Industry 59 (2008) 641–646.
- [13] C. Chituc, A. Azevedo, C. Toscano, A framework proposal for seamless interoperability in a collaborative networked environment, *Computers in Industry* 60 (2009) 317–338.
  - [14] A. Trappey, D. Hsiao, Applying collaborative design and modularized assembly for automotive odm supply chain integration, *Computers in Industry* 59 (2008) 277–287.
  - [15] H. Chen, H. Kang, X. Xing, A. Lee, Y. Tong, Developing new products with knowledge management methods and process development management in a network, *Computers in Industry* 59 (2008) 242–253.
  - [16] B. Eynard, T. Gallet, P. Novak, L. Roucoules, Uml based specifications of pdm product structure and workflow, *Computers in Industry* 55 (2004) 301–316.
  - [17] D. Gimenez, M. Vegetti, H. Leone, G. Henning, Product ontology: Defining product-related concepts for logistics planning activities, *Computers in Industry* 59 (2008) 231–241.
  - [18] R. Sudarsan, S. Fenves, R. Sriram, F. Wang, A product information modeling framework for product lifecycle management, *Computer-Aided Design* 37 (2005) 1399–1411.
  - [19] S. Fenves, S. Foufou, C. Bock, R. Sriram, Cpm2: A core model for product data, *Journal of Computing and Information Science in Engineering* 8.
  - [20] S. Rachuri, E. Subrahmanian, A. Bouras, S. Fenves, S. Foufou, R. Sriram, Information sharing and exchange in the context of product lifecycle management: Role of standards, *Computer-Aided Design* 40 (2008) 789–800.
  - [21] Y. Tong, D. Li, M. Yuan, Product lifecycle oriented digitization agile process preparation, *Computers in Industry* 59 (2008) 145–153.
  - [22] Y. Ma, G. Chen, G. Thimm, Change propagation algorithm in a unified feature modeling scheme, *Computers in Industry* 59 (2008) 110–118.
  - [23] J. Shiau, H. Wee, A distributed control workflow for collaborative design network, *Computers in Industry* 59 (2008) 119–127.
  - [24] G. Meyer, K. Fr mmling, J. Holmstr m, Intelligent products: A survey, *Computers in Industry* 60 (2009) 137–148.
  - [25] P. Valckenaers, B. S. Germain, P. Verstraete, J. V. Belle, Hadeli, H. V. Brussel, Intelligent products: Agere versus essere, *Computers in Industry* 60 (2009) 217–228.
  - [26] X. Yang, P. Moore, S. Chong, Intelligent products: From lifecycle data acquisition to enabling product-related services, *Computers in*



- Industry 60 (2009) 184–194.
- [27] V. Marik, J. Lazansky, Industrial applications of agent technologies, *Control Engineering Practice* 15 (2007) 1364–1380.
  - [28] M. Contreras, L. Sheremetov, Industrial application integration using the unification approach to agent-enabled semantic soa, *Robotics and Computer-Integrated Manufacturing* 24 (2008) 680–695.
  - [29] P. Forget, S. D'Amours, J. Frayret, Multi-behavior agent model for planning in supply chains: An application to the lumber industry, *Robotics and Computer-Integrated Manufacturing* 24 (2008) 664–679.
  - [30] A. Trappey, T. Lu, L. Fu, Development of an intelligent agent system for collaborative mold production with rfid technology, *Robotics and Computer-Integrated Manufacturing* 25 (2009) 42–56.
  - [31] S. Huk, H. Kim, J. Lee, S. Han, S. Park, An e-engineering framework based on service-oriented architecture and agent technologies, *Computers in Industry* 59 (2008) 923–935.
  - [32] P. Grefen, N. Mehandjiev, G. Kouvas, G. Weichhart, R. Eshuis, Dynamic business network process management in instant virtual enterprises, *Computers in Industry* 60 (2009) 86–103.
  - [33] T. Kaihara, A multiagent-based complex systems approach for dynamic negotiation mechanism in virtual enterprise, *Robotics and Computer-Integrated Manufacturing* 24 (2008) 656–663.
  - [34] E. Frazelle, *Supply Chain Strategy*, McGraw Hill, 2002.
  - [35] S. Kalpakjian, S. Schmid, *Manufacturing Engineering and Technology*, 4th Edition, Addison-Wesley, 2002.
  - [36] M. Maier, Architecting principles for systems-of-system, *Systems Engineering* 1 (4) (1998) 267–284.
  - [37] M. Marchetta, R. Forradellas, Supporting interleaved plans in learning hierarchical plan libraries for plan recognition, *Inteligencia Artificial: Revista Iberoamericana de IA, (AEPIA, Asociacion Española para la Inteligencia Artificial)* 32 (2006) 47–56.
  - [38] M. Marchetta, R. Forradellas, Artificial intelligence planning for generative computer aided process planning, in: *19th International Conference on Production Research*, 2007.
  - [39] W. Shen, L. Camarinha-Matos, R. Brennan, Special issue on multi-agent and holonic systems in manufacturing, *Robotics and Computer-Integrated Manufacturing* 24 (2008) 595–596.
  - [40] F. Mayer, J. Auzelle, Is system of systems a candidate rationale artifact for enterprise information-intensive system modeling? , in:

9th MITIP International Conference on The Modern Information Technology in the Innovation Processes of the Industrial Enterprise, 2007.

- [41] M. Wooldridge, N. Jennings, Intelligent agents: Theory and practice, *The Knowledge Engineering Review* 10 (2) (1995) 115–152.