Making Products Active with Intelligent Agents for Supporting Product Lifecycle Management

Martin G. Marchetta[1], Frédérique Mayer[2], Raymundo Forradellas[1]

1CEAL, Logistics Studies and Applications Centre, School of Engineering, National University of Cuyo, Centro Universitario, CC405 (M5500AAT) Mendoza, Argentina
mmarchetta@fing.uncu.edu.ar
kike@uncu.edu.ar

2ERPI – Equipe de Recherche sur les Processus Innovatifs, ENSGSI, Institut National Polytechnique de Lorraine, 8 rue Bastien Lepage, BP 90647 (54010) Nancy Cedex, France
frederique.mayer@ensgsi.inpl-nancy.fr

Abstract. Modern organization paradigms within manufacturing enterprises have arose in last years, like Agile Manufacturing and collaboration, in order for enterprises to increase their productivity and be more competitive in front of shorter due dates and increasing product qualities required by customers. Most previous works on PLM and currently available systems are usually focused on the use of additional information to support business processes, and integrate limited information of lower-level applications (CAD, CAPP, etc). However, little emphasis has been put on making products more intelligent during their complete lifecycle, in order to exploit PLM information for improving their development and management. In this paper, a framework based on intelligent agents is proposed, for giving products active behaviors, in order to assist people involved in PLM to reduce lead times and costs, and improving product quality. Application of the proposed framework to a product definition example is presented as a case study.

Keywords: PLM, Active Product, Intelligent Agent, Virtual Enterprise, Concurrent Engineering.

1 Introduction

Current international context has forced enterprises to optimize their operations and to collaborate, in order to be competitive. This has yielded the need of reducing lead times and costs, and increasing final product quality. In order to achieve these objectives, collaboration among peer enterprises has been more and more accepted, creating the so called virtual enterprises [1, 2].

Product Lifecycle Management (PLM) has been considered a key concept in order to maintain consistency, efficiency and quality as products are created, from conception to disposal [3, 4]. PLM involves the management of product information

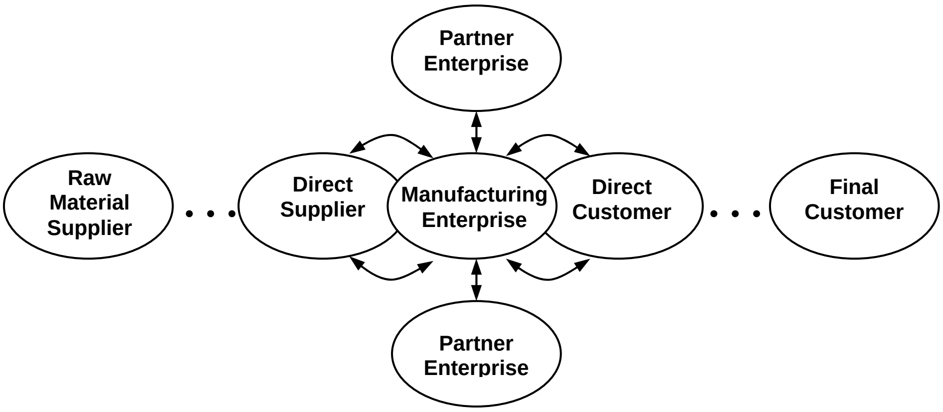

Fig. 1. Horizontally and vertically integrated environment of current manufacturing enterprises

and the integration of business processes from birth to obsolescence of a product [5], and it is crucial for effective management of corporate intellectual capital [6].

Until some years ago, integration and collaboration was implemented mostly along the supply chain. However, currently this integration has been extended vertically, including not only supplier and customers as partners, but also peer enterprises, as shown in figure 1. These partner enterprises share much information concurrently, more than customers and suppliers, so PLM now turns to be a fundamental support to maintain consistency and to capitalize concurrent engineering and collaboration benefits in these scenarios.

Current mixed vertical [7] and horizontal relationships [2] create special issues not present before. Because business processes now cross enterprises' physical barriers, products information must be shared across heterogeneous systems (with different data formats and semantics), and coordination of activities needs to consider people, information and other resources with different ownership, policies and cultures.

Previous works have focused on several aspects of PLM, particularly on information models and business processes integration. However, even the same product development, deployment and disposal activities are much more difficult to be carried out because of the complexity of current schemas. Little efforts have been done in order to put more intelligence on products during their own development and management process, in order to reduce this complexity and make it manageable. Currently, the evolution of information technologies allows giving active behaviors to (both materialized and under development) products, a characteristic from which companies can take advantage.

Giving such autonomy to products can help to handle complexity, reducing inconsistencies, and optimizing product's definition and manufacturing activities. In this paper, a framework for making products actively involved in their own development and management in PLM is proposed, based on intelligent agents. The proposed model considers the exploitation of product information by means of a new actor in the development process of new products, an active-product, which gives advice to users of a PLM system, improve product quality and reduce costs and time to market.

2 Previous works on information exploitation within PLM

As a new concept, PLM has evolved in recent years along different research directions. These research and development lines are focused on the creation, storage and use of product information. Among these aspects, the most relevant one from the point of view of this work is the product information exploitation along its lifecycle. This aspect raises some issues, such as interoperability due to different information formats and semantics, and the difficult to guarantee completeness and consistency of information across departments and enterprises involved in the PLM business processes [3].

Some works propose limited supporting and coordination mechanisms both within a single activity and between activities and partner enterprises. The main focus in this kind of proposals is to give assistance on a specific activity, like CAPP [8], without considering coordination and integration with other activities. Another proposal is to include support for collaboration among customers and suppliers, and at the same time between partner enterprises, but focused on some activities (CAPP/CAM), not the complete product lifecycle [9]. In this case, assistance given by the system is limited, including only simple coordination and concurrency management mechanisms. However, a more complete support is needed in order to realize PLM benefits, as will be described in later sections.

The other idea for information exploitation is related to change propagation. Ma et. al. [10] address the problem of maintaining consistency of product information when modifications are done during iterations along lifecycle activities within a single company. They propose the use of a unified feature modeling scheme, on which different kind of product features (functional, geometric, etc) are related to each other by means of a dependency network, in order to determine changes impacts. Shiau and Wee [11] propose an algorithm for maintaining consistency of design documents in a configuration management environment, both within a company and with respect to its partners. The algorithm tries to minimize the chain of changes needed when a design modification is required.

As can be seen from the works mentioned above, the kind of applications taking advantage of information along the product lifecycle is restricted to a few activities or to specific problems, and the focus is on specific algorithms and methods. This yields the need of a more general approach of information exploitation within PLM, including business processes, information models and information exploitation models.

3 Business process model for PLM and SCM

In previous works, the product role within PLM has been to be an information hub, concentrating all data required by different activities along the PLM. This limited view allows people from companies to get answers to questions regarding product information, but only people take the initiative for exploiting this information. PLM business processes have also been studied, but there has been a separation between

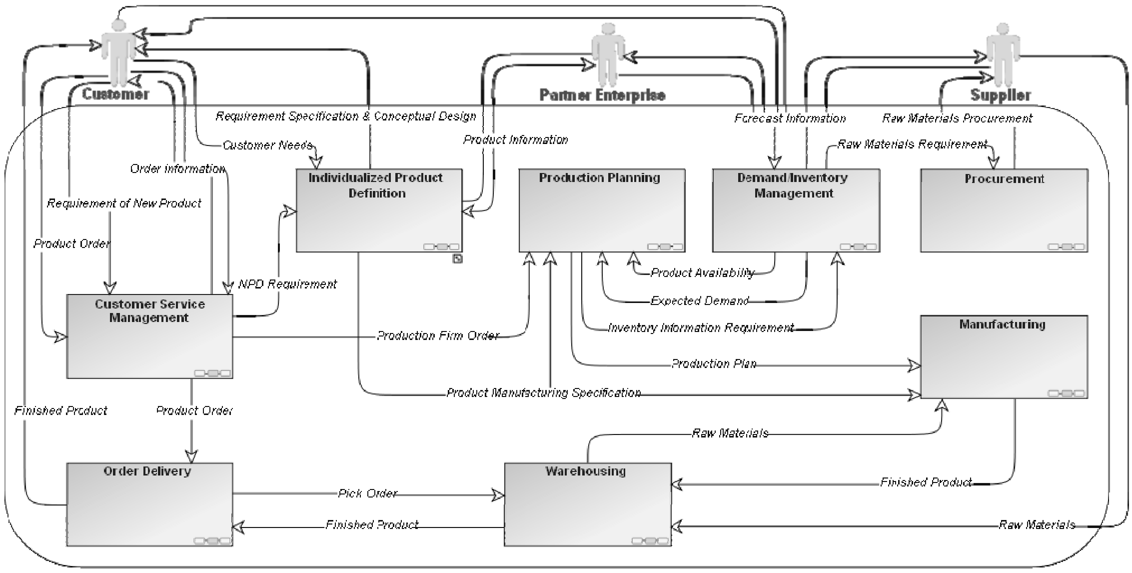

Fig. 2. Overview of Business Process Model for joint PLM and SCM

these two dimensions (business processes and information to support them), what reduces the benefits obtained from the implementation of PLM.

Current technologies allow putting active behaviors inside information systems, what turns them into an active actor during PLM activities. This gives the possibility of having additional benefits as product information is generated and stored, like identifying optimization opportunities or detecting potential risks, rather than only passively answer questions when people need it.

Several processes must be carried out by an industrial enterprise having relationships both along the supply chain (with suppliers and customers), and with partners during product development and/or production. Considering this scenario, figure 2 depicts a logistics and production oriented business process model.

The model is based on two different (and somehow complementary) views of business processes in logistics [2, 12]. Processes of a product development and manufacturing company are shown inside the rectangle. The supplier, the customer and the partner enterprises are external actors. Partners are enterprises which design other product's components, or that manufacture products designed by the company.

Two types of customer orders may be received by the enterprise: new product development orders and existing product orders. In addition to this, at any moment the customer may ask for information about its order's state and/or the state of the development process of the new product. All these information flows are handled by the Customer Service Management process (called Customer Response by Frazelle [12]).

The Order Delivery process takes care of coordinating all activities to put products in the location accorded with the customer. The Warehousing process includes all warehouse management activities, such as picking, put away, storage, etc. Individualized Product Definition is a global process which groups all new product

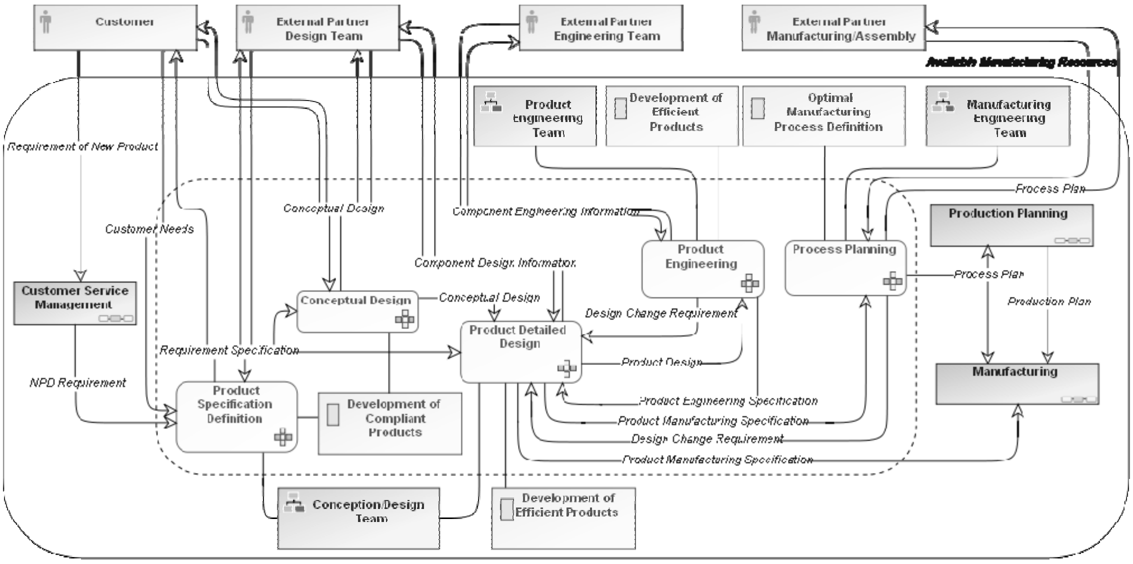

Fig. 3. Individualized Product Definition's functional diagram

development (NPD) activities, including turning customer needs into requirement specifications, designing the product, creating manufacturing specifications, etc. This process exchanges information with customers and it also involves interactions with partner enterprises which are in charge of designing other components or who manufacture/assemble them.

The Production Planning process aims at accommodating demand to manufacturing resources considering firm orders, forecasts and manufacturing process plans indicating routings, bills of materials, time and other resources. Demand/Inventory Management forecasts demand in order to support procurement and production planning. Procurement is concerned with handling of buying orders. It is part of the Supply process as defined in [12], and part of the Supplier Relationship Management as proposed in [2]. Finally, the Manufacturing process is in charge of products creation using manufacturing resources. It mainly uses product specifications from Individualized Product Definition and the amount of units to be produced, when they should be produced and with which resources from Production Planning.

Vertical integration with partner enterprises is mostly carried out by means of the Individualized Product Definition process, yielding a great number of interactions of all types, including interactions along the supply chain, interactions with partner enterprises, and also within several activities inside the process itself.

Figure 3 presents the Individualized Product Definition functional diagram. The process is composed of 5 main activities: Product Specification Definition, Conceptual Design, Product Design, Product Engineering and Process Planning. In a concurrent engineering environment, these activities are carried out in an overlapped way, rather than sequentially. Thus, there is a complex (and usually asynchronous) information exchange among them.

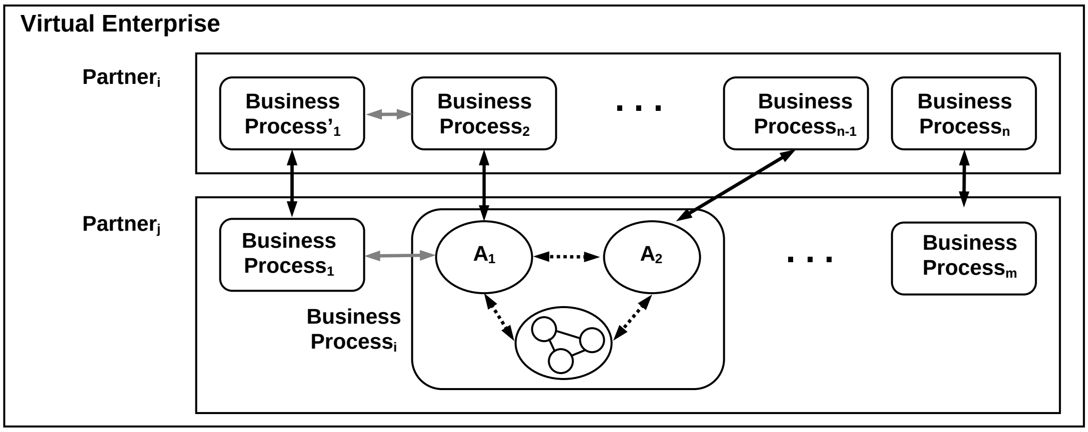

Fig. 4. Hierarchy of active-product's interventions along its lifecycle

3.1 Hierarchy of product-automated support along product lifecycle

Integrated PLM within a concurrent engineering/virtual enterprise paradigm involves interactions at several abstraction levels. These interactions are points for potential contribution of automated support by an active product. Within specific applications areas, such as change propagation, interactions at some of these abstractions levels have been considered, namely inter-company and intra-company scenarios [11] and inter-activity interactions within a single company [10].

In addition to this, two additional abstraction levels are considered here. First, the product development process, as defined in this article, includes several internal activities, but it also has relations with other business processes within a single organization. Second, individual activities themselves present opportunities for improving the development process by means of an active product approach. Thus a 4-level interactions hierarchy, depicted in figure 4, is proposed in this work. The hierarchy considers both the business processes and the virtual enterprise dimensions, through the following interaction types:

1. *Inter-process/multiple company interactions*: it is represented in the figure by thick continuous arrows. It involves information exchanges between processes in different companies within the virtual enterprise. Information ownership must be taken into account, since interactions cross a single enterprise barriers.
2. *Inter-process/single company interactions*: this case is depicted by gray arrows in the figure. 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 of coordinating activities which contribute to its value chain. This is especially true when activities whose results impact on the other ones are executed concurrently.

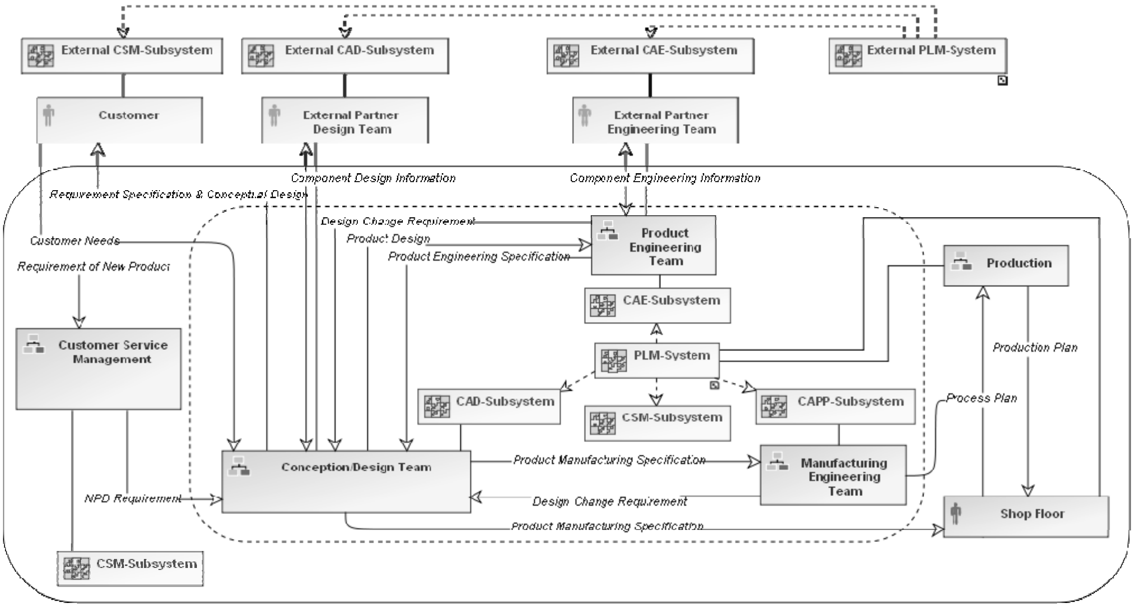

Fig. 5. IT applications architecture typically used within virtual enterprises for product development

This interaction type is shown in the figure with dotted arrows, and activities are represented by ellipses.

4. *Intra-activity interactions*: Within a single activity the number of people involved may be very small, and so the information exchanged, so the most important thing an active product may contribute with is exploitation of information in order to optimize results of that activity.

An active product is similar to an expert advisor who actively integrates all the information across the product's lifecycle. The interactions hierarchy is useful since at each level there exist different potential opportunities for realizing an active product's benefits. For inter-process situations (types 1 and 2), most important benefits include change impact analysis, propagation/notification of changes to (and only to) relevant people, global optimization, risk assessment, feedback project on 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 [13,14], plan recognition for identifying user's intentions, retrieval of patterns used in the past in a mixed-initiative approach [15], automated task completion [16], etc.

3.2 Applications architecture

Software applications that support product development have emerged independently from the need for automated support of individual activities (CAD, CAM, scheduling,

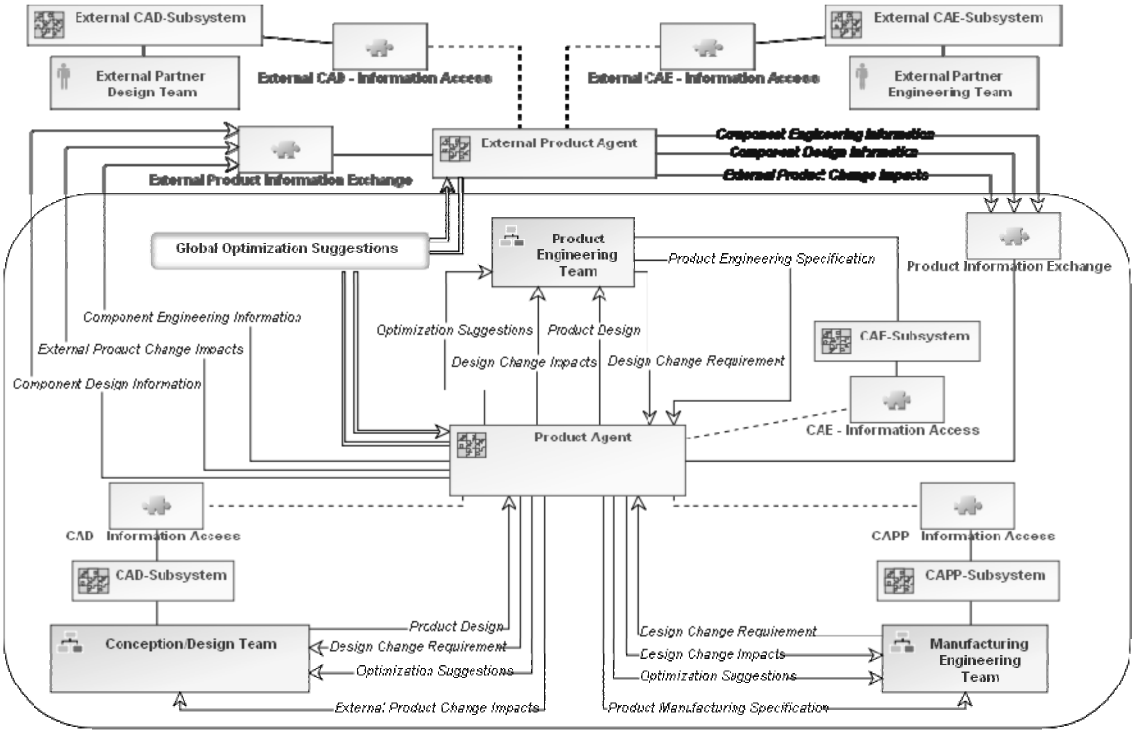

Fig. 6. Proposed IT applications architecture for supporting active-products

CRM, etc), which has generated issues related to interoperability, data formats, lack of coordination and collaboration support, etc. Figure 5 shows a typical applications architecture used within product development companies within virtual enterprises.

Because of the great diversity in the nature of activities covered along the PLM, it is difficult to create a single software application for supporting all of them. Besides, many companies have made great investments in solutions to individual problems. Moreover, within virtual enterprises it is likely that different software solutions be used to solve the same problems (e.g. different CAD/CAPP/CAM systems). Because of these reasons, interoperability between existing applications has turned to be a recurrent issue.

Usually PLM systems provide interoperability by being an information backbone [17]. Standards are very important in order to achieve interoperability [18], and those related to information exchange, such as the ISO 10303 [19], have supported this trend.

The problem with this approach is that much information must be converted to other formats in order to pass it from one system to another, which usually produces some semantic information loss. Besides, these information exchanges are usually made by means of manual coordination mechanisms, and through external support applications such as e-mail. Product information should be handled within the systems as much as possible, avoiding conversions and reconversions, and non-structured communication mechanisms.

Especially because of the “pull strategy” of product information exploitation, many optimization opportunities as well as early detection of global problems can be missed. These improvement opportunities may be related to any of the interaction levels mentioned in section 3.1.

Figure 6 depicts the proposed global IT architecture. For simplicity, only entities relevant to the individualized product definition process are included in the figure. The architecture aims at providing a structured integration infrastructure to:

1. Support development of applications with proactive behaviors.
2. Support enhanced interoperability among heterogeneous systems (different applications, from different vendors, owned by different organizations, etc).
3. Reduce information exchanged in non-structured formats.
4. Reduce information re-work, by including rationale and semantics behind product development decisions within the PLM system.

This architecture can be accommodated to be independent of the particular software solution or application vendor. In order to realize the proposed objectives, as much semantic information as possible must be put into the system. This will allow not only pointing to product design and manufacturing specification files, but also to automatically reason on them, make improvement suggestions, identify impact of distributed changes made along different activities, and support other kind of improvements such as global optimization.

It is common nowadays to have interface to access information stored within systems through web services and Service Oriented Architectures (SOA) standards. This allows reusing knowledge already available in current applications, such as CAD, CAPP, CAM, ERP, etc, and gives the architecture the capability of taking advantage of information distributed across different locations. As depicted in figure 6, coordination, collaboration and information exchanges are possible between organizations through interfaces exposed as services.

Since there is no system having all the product’s information, global optimization can be achieved by means of collaborative optimization techniques. The main idea is to put the product at the core of its lifecycle management, not only as a reactive system, but also as a proactive entity capable of identifying opportunities to be exploited and to suggest how to do that.

In figure 6, an entity called Product Agent is at the center of the architecture. Product Agent is an Intelligent Agent [20, 21], whose environment is composed of both reactive (e.g. other applications) and proactive (e.g. human users, teams or other artificial agents) entities involved in PLM activities and processes. 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, such as when information from different partners must be put together. 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.

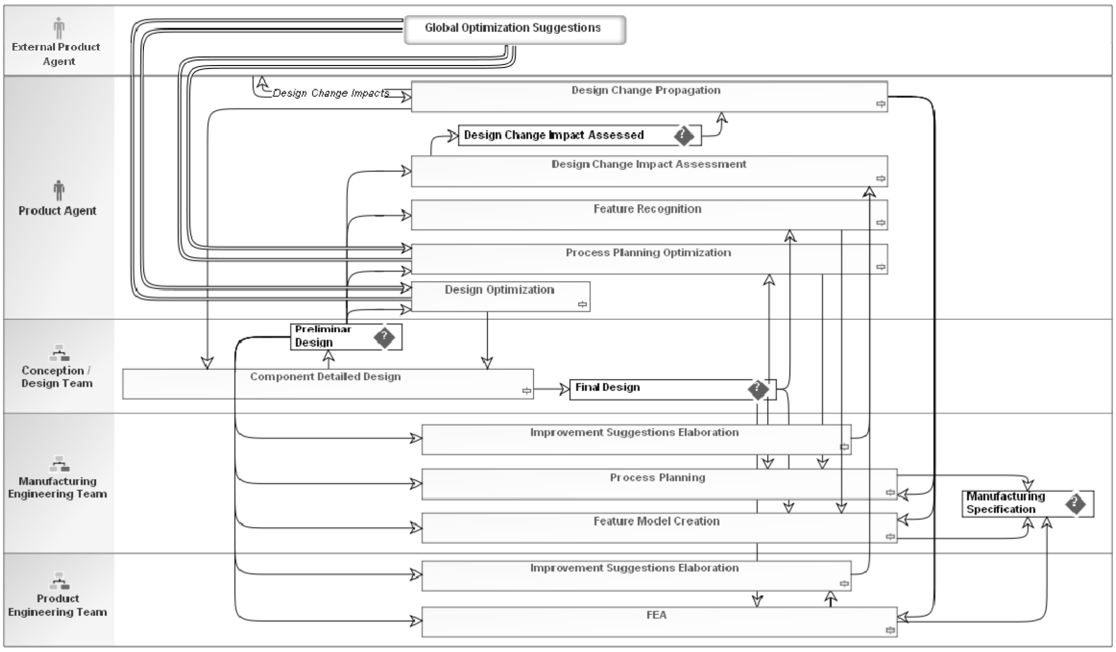

Fig. 7. Flowchart diagram for the Individualized Product Definition supported by Product Agent

Thus, the system moves from isolated automation islands towards the concept of system of systems. This concept is related to adaptable distributed systems which interact with each other, resulting in productivity and functionality that are greater than that provided by the sum of individual systems. Behaviors are associated to the product entity, instead of considering it as a concept represented by the underlying systems, data bases, information models and processes.

Several issues must be solved in order to achieve the enhanced capabilities mentioned above. First, this new system must be able to automatically exploit information created and processed by people scattered across the virtual enterprise. Second, the system must be capable of communicating with other similar systems, which may represent other's interests and intentions. Third, it must also be able to access information from both autonomous and non-autonomous systems, such as CAD and CAPP systems. And finally, it should do all of this autonomously and dynamically, which means that it must detect special situations and act accordingly without having continuous and direct human intervention. Intelligent agents are a suitable technology for supporting this.

3.3 Proactivity and Intelligent Agents

Figure 7 presents a flowchart diagram representing a temporal view of activities within a new product development a project. In order to proactively assist teams and organizations in PLM, a product needs to have the ability to detect relevant events and to make decisions automatically. Wooldridge and Jennings [22] discussed several definitions of intelligent agents including a “weak” one in term of 4 properties an intelligent agent should have: autonomy, social ability, reactivity and proactivity.

Autonomy is related to the level on which the agent can act by itself without human intervention. Since one of the most complex tasks within concurrent engineering in large projects is to detect relevant events (such as collateral effects of changes or optimization opportunities) and coordinate activities, an intelligent product agent should be as autonomous as possible.

Social ability allows an agent to communicate with other agents and also with humans. This is useful since partner enterprises do not need to use the same applications or have the same data formats, as long as the agents speak the same language (both semantically and syntactically). Agents also expose a set of services they can do (such as give information to other agents or human users), and hence this schema is similar to the one proposed by Service Oriented Architectures. This also allows tackling information property problems since agents act as “delegates” which restrict the information other partners have access to.

Reactivity means that agents should receive requirements from human users, and respond to them. On the other hand, proactivity is related to goal-directed behaviors, which means that the product agent may take the initiative for detecting events in the environment (e.g. optimization opportunities, design changes which impact other people’s work, etc), suggest optimizations and help coordinate activities without human intervention.

Since these 4 properties may support a tighter integration between activities and people along PLM, intelligent agents are likely to be a useful technology for achieving the proposed objectives.

6 Case study

In order to illustrate the schemas proposed in this work, a simple case study is commented here. Figure 7 shows a flowchart diagram of activities required by a product development project, starting with the detailed design (for simplicity previous activities are not shown in the diagram). In the figure, development teams within an organization interact with partner organizations and also with the product agent.

After a product requirement specification is available, the first stage is composed of the creation of a preliminary design. As soon as a preliminary design is created, most of other activities can be started. Product engineering and manufacturing engineering team’s activities can be performed in parallel. At the same time, the product agent assists these teams both locally and globally.

In this stage, the product agent gives assistance to PLM by: assessing impact of design changes, propagating changes to affected people and suggesting local and global optimizations. Design changes may be required for several reasons, such as engineering problems, because a partner enterprise changes the design of some component, etc. Assessing the impact of changes includes determining which people are affected, how they are affected, and possibly suggestion for accommodating these changes.

Local optimizations and assistance involve making easier some tasks such as feature identification and process planning, and suggesting alternatives for improving results (e.g. design optimizations, process planning optimizations, etc). These optimizations may yield great benefits in terms of time and money. Global optimizations involve the identification of improvement opportunities which go beyond the scope of a single activity or enterprise. When products are complex and their components specifications are produced in different partner enterprises, these optimizations may allow an optimized product as a whole, instead of local optimizations.

After the final design has been obtained, final engineering analysis and process planning need to be done in order to complete de product specification. Design optimizations and changes are not allowed after this stage, but other assistances continue to be performed (e.g. process planning).

After all activities are finished, the final Manufacturing Specification is obtained, which includes product's materials, manufacturing processes to be applied to raw materials, machines and tool types to be used, etc. This information will later be used for production scheduling and manufacturing in order to create the physical product. The result of the complete process is composed of the final design and the manufacturing specification.

7 Conclusions and Future work

In this paper the use of modern technologies, particularly intelligent agents, was proposed to improve and enhance information exploitation along PLM.

One contribution of this work is the integrated treatment of both the vertical virtual enterprise setting seen from the product development stages, and the horizontal Supply Chain Management integration, represented in the general Business Process Model proposed.

Another contribution is the proposal of an application architecture to support PLM by means of an intelligent product agent, having the properties of autonomy, social ability, reactivity and proactivity. In this work a hierarchy of the kind of assistance an active product agent may give to product development teams is also proposed, which is exemplified with some concrete coordination, local and global optimization opportunities. This allows generalizing the joint enterprise and application architecture to apply these and other automated assistances, instead of concentrating on a single one (such as change propagation).

From the enterprise and application architecture point of view, future work will include the application of the same active-product concept to previous and later stages, along manufacturing and logistics administration, giving the product the ability to coordinate its own production in the shop floor, storage, transportation, raw materials procurement, etc. In this approach a product would have an immaterial existence (the product agent, including its own information and knowledge), and a material one (the product that is gradually constructed, stored, transported, etc). This enables new production management schemas to explore, such as distributes manufacturing and logistics coordination.

From the framework realization point of view, some research has already been done in the local optimization and assistance (the fourth type of the proposed interactions hierarchy), mostly in the CAPP activity [13, 14, 16]. Further work is needed in order to implement assistance along the complete proposed hierarchy. In the near future, the implementation of the CAPP local assistance agent will be finished, and research will continue for realizing agent support to other activities, especially on coordination, global optimization and change propagation.

Acknowledgements

This research was jointly developed by the School of Engineering (UNCuyo, Argentina) and École Nationale Supérieure de Génie de Systèmes Industriels (INPL, France), and partially financed by the PREMER F-599 and ARFITEC ARF-08-04 programs, and CONICET PhD fellowship.

References

1. X.G. Ming, J.Q. Yan, X.H. Wang, S.N. Li, W.F. Lu, Q.J. Peng and Y.S. Ma, Collaborative process planning and manufacturing in product lifecycle management, Computers in Industry 59 (2008) 154-166.
2. K.L. Croxton, S.J. García-Dastugue, D. Lambert and D. Rogers, The Supply Chain Management Processes, The International Journal of Logistics Management, Vol. 12, No 2 (2001) 13-36.
3. A. Saaksvuori and A. Immonen, Product Lifecycle Management, 2nd edition, Springer Berlin, 2005.
4. G. Schuh, H. Rozenfeld, D. Assmus and E. Zancul, Process oriented framework to support PLM implementation, Computers in Industry 59 (2008) 210-218.
5. A. Sharma, Collaborative product innovation: integrating elements, of CPI via PLM framework, Computer-Aided Design 37 (2005) 1425–1434.
6. K. Amann, Product Lifecycle Management: empowering the future of business, CIM Data, Inc.; 2002.
7. M. P. Bhandarkar and R. Nagi, STEP-based feature extraction from STEP geometry for Agile Manufacturing, Computers in Industry 41 (2000) 3-24.
8. Y. Tong, D. Li, M. Yuan, Product lifecycle oriented digitization agile process preparation, Computers in Industry 59 (2008) 145-153.
9. Y. Ma1, G. Chen and G. Thimm, Change propagation algorithm in a unified feature modeling scheme, Computers in Industry 59 (2008) 110-118.
10. J. Shiau, H.M. Wee, A distributed control workflow for collaborative design network, Computers in Industry 59 (2008) 119-127.
11. E. Frazelle, Supply Chain Strategy, McGraw Hill, 2002.
12. M. Marchetta, R. Forradellas, A new model for automatic generation of plan libraries for plan recognition, Brazilian Journal of Operations and Production Management, Vol. 3 No. 2 (2006) 5-19.
13. M. Marchetta and R. Forradellas, Supporting Interleaved Plans in Learning Hierarchical Plan Libraries for Plan Recognition, Inteligencia Artificial: Revista Iberoamericana de IA, (AEPIA, Asociación Española para la Inteligencia Artificial), No. 32 (2006) 47-56.

14. J.E. Allen, C.I. Guinn and E. Horvtz, Mixed-Initiative Interaction, IEEE Intelligent Systems and Their Applications Vol. 14 Issue 5 (1999) 14-23.
15. M. Marchetta and R. Forradellas, Artificial Intelligence Planning for Generative Computer Aided Process Planning, 19º International Conference on Production Research, Valparaiso, Chile, 2007.
16. B. Denkena, M.ShpitaIni, P. Kowalski, G. Molcho and Y. Zipori, Knowledge Management in Process Planning, CIRP Annals – Manufacturing Technology Vol. 56 Issue 1 (2007) 175-180.
17. S. Rachuri, E. Subrahmanian, A. Bouras, S.J. Fenves, S. Foufou and R.D. Sriram, Information sharing and exchange in the context of product lifecycle management: Role of Standards, Computer-Aided Design 40 (2008) 789-800.
18. ISO 10303-1: Industrial automation systems and integration – Product data representation and exchange, Part 1: Overview and fundamental principles, International Standards Organization, 1994.
19. S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Ed., Prentice Hall, 2002.
20. M. Wooldridge, An Introduction to MultiAgent Systems, John Wiley & Sons Ltd, 2002.
21. M. Wooldridge and N.R. Jennings, Intelligent Agents: Theory and Practice, The Knowledge Engineering Review, 10(2) (1995) 115-152.