Third European Conference on Management of Technology
EuroMOT 2008

Contribution to the Formalization of Individualized Processes during Product Innovation

EVARISTO CASTRO
FREDERIQUE MAYER
PASCAL LHOSTE

Nancy-Université, Equipe de Recherche sur les Processus Innovatifs (Research Team on Innovative Processes) 8, rue Bastien Lepage
Nancy, 54010, France
Evaristo.Castro@ensgsi.inpl-nancy.fr

Abstract. Over the years, innovation has become the central element of enterprise strategy. It is only thanks to innovation that a product or a service differentiates from its competitors and creates the conditions for growth. Nowadays, innovative enterprises have to consider two aspects about new product development: (1) that it generates large product variability and (2) that it increasingly tends to regard the needs of particular customers. These two angles related to product innovation activities cause an enhancement of informational complexity. This situation complicates the management of the product's information, especially during the first phases of its lifecycle. The Product Lifecycle Management (PLM) aims to integrate product information within an information system all through its life cycle. Product traceability throughout the PLM has been improved thanks to PLM Software and the RFID (Radio Frequency Identification) technology. In particular, this technology allows collecting information individually on a given product. Our work takes part in the scope of "Product-Driven Systems". Since the product carries information, it has been given an active role during its production. It interacts individually with some parts of its production system. However, this notion of individualization still hasn't been well identified. Indeed, up to date, there is no characterization that could permit a better comprehension of what product and process individualization are about. Our aim is to make a contribution to this formalization and characterization of an individualized product. Principally, our objective is to propose an intelligible and coherent representation of an individualized product and of its relationship with its production system during the product innovation phases in the context of Systems Engineering (SE). In this paper we present the state of the art of the concepts, theories, approaches, modeling languages and modeling tools to achieve our objective. We present as well the first models that have resulted from our research. These models are still to be developed. Finally, we give a conclusion and perspectives of our future work.

Keywords: individualization, innovation, modeling, complexity, Systems Engineering.

I. Introduction

Innovation is a widely used term to designate new things. Galanakis (2006) makes a review of several authors, where many definitions of innovation are found. A comparison and discussion conducted by Mitterdorfer-Schaad (2001) concludes, that innovation generally includes radical or incremental steps of development in technology, products design, and market reach. Thus it can be defined as *"a company's first successful commercial application of something new"* (Bannert, 2004). During the last decades, Innovation has been identified by several nations or intra-nation organizations as the major factor of economic growth and wealth (EU, 1995) (OECD, 1997a, b).

Innovation driven industries use the development of new products or services as its main source of value creation to attract customers. These kinds of companies have to consider the two following aspects concerning new product development:

(i) It generates **large product variability**. By reducing the cost of making smaller quantities of products, flexible manufacturing has enabled companies to make a much wider variety of goods –all with the goal of giving customers exactly what they want (Fisher *et al.*, 1994, p. 86).

(ii) It increasingly tends to regard the **needs of particular customers**. Recent technological advances have reduced the cost of designing and producing products for "markets of one" (Franke and von Hippel, 2003).

These two angles related to product innovation activities searching **individual** satisfaction cause an enhancement of **informational complexity**. This is partly because information systems integration and networked innovation processes (Rothwell, 1994) allow downstream phases in the product lifecycle to contribute in the product innovation processes in a more direct and reactive way. This is a consequence of increasing pressure for more rapid product development and decreased time to market (Chapman and Hyland, 2004).

Creating products or services implies the flow of information, materials and energy. The manner in which all of this is handled results in networks of a high degree of complexity. In many product innovation processes the interface between the enterprise and customers is becoming increasingly complex, as it requires detailed communications across what Scott (2000) sees cross-organizational boundaries. Across these boundaries there is an exchange of information on a wide range of issues such as quality, process knowledge, outcomes of experiments, and financial and costs structures. It is not only the complexity of the customer interface that adds to the overall complexity in product innovation. As Scott (2000) argues, product innovation requires a diversity of expertise, as it is increasingly technically complex and capital intensive. The process of product innovation can involve a single innovator or an integrated cooperative venture involving customers, suppliers, and a cross functional group or team from within an organization. The complexity of individual products often requires a large group of designers, engineers, and project managers working across an extended manufacturing enterprise in geographically distributed locations. This situation complicates the management of the product's information, especially during the first phases of its lifecycle.

PLM (Product Lifecycle Management) is the activity of managing a company's products all the way across their lifecycles in the most effective way (Stark, 2004). In doing so, it enables the company to take control of its products. Keeping control isn't easy. During the development of a product, it doesn't physically exist, so during that phase of its life it's difficult to control. Once a product does exist, it should be used at a customer location, where again, it's difficult for a company to keep control of it.

Product traceability is the way enterprises have found to keep control of their products. The term traceability related to the product or manufacturing has been defined since the 90ies (Cheng and Simmons, 1994), when a series of industrial needs had been highlighted into the establishment of ISO 9000 procedures. Generally, product traceability is the ability of a user (manufacturer, supplier, vendor…) to trace a product through its processing procedures, in a forward and/or backward direction (Jansen-Vullers *et al.*, 2003). Physically, product traceability deals with maintaining records of all materials and parts along a defined lifecycle (e.g. from raw material purchasing to finished goods selling) using a coding identification. Traceability systems are adopted, according to laws, in the food sector, in manufacturing, in the pharmaceutical sector, in distribution, in constructions. Traceability systems can be useful to increase quality and safety of products, for brand protection, and in order to increase efficiency in production and distribution (Terzi *et al*, 2006).

Product traceability throughout the PLM has been improved thanks to PLM Software[1] and the RFID technology (Radio Frequency Identification) (McFarlane *et al*, 2003). In particular, this technology allows collecting information individually on a given product. In this context, our work takes part in the scope of "Product-Driven Systems" (Gouyon *et al.*, 2007) (Wong *et al.*, 2002). Let's suppose that each product could be provided with an advanced "product information store system" (e.g. RFID based), in order to be (1) from one side tracked into a system (e.g. a plant) and, from another side, (2) to be able to provide itself the needed information. This hypothetical capability gives the product an active role during its own production, in which it seems to "control" the manufacturing system. The product interacts **individually** with some parts of its production system. In such a vision, the product itself becomes the medium of the data set, instantiating a kind of "intelligent product" (Kärkkäinen *et al.*, 2003) (Morel *et al.*, 2004) able to interoperate with the environment, exchanging information (which is within the product itself) in real-time with different resources (e.g. machines and transporters in a plant, or trucks and inventory databases in a warehouse, or with refrigerators and dishwashers at home…).

In this context, our main objective is to **formalize and characteriz**e this notion of product individualization. We begin by asking basic questions like "What exactly is an individualized product?" "What does product individualization implies?" "To what extent does individualization affects the environment where it is situated?" By answering these and other questions we will propose a characterization that could permit a better comprehension of what product and process individualization are about. Our aim is to make a modest contribution to the formalization and characterization of an individualized product. In this way we aim to propose an intelligible and coherent representation of an individualized product and of its relationship with its production system during the product innovation phases in the context of Systems Engineering (SE).

Systems Engineering (SE) is an interdisciplinary methodological approach to apprehend and master the conception and development of complex systems and products. We consider SE appropriate because it allows us to make the hypothesis that the product can be an individual system. Still in the vision of "Product-Driven Systems", the product's journey throughout its manufacturing phases demands the flexibility and adaptability of the production system, because the information that every product carries for its fabrication can differ. Yet, we are focused only on the innovation and conception phases of the product's life cycle. By giving the product an active capability, our objective is to add the model of an individualized product to the conception phase considering that innovation processes can be triggered.

This paper is organized as follows. Initially, we present the scope, approach and limits of our research. Then, we gather and present the state of the art of the concepts, theories, approaches, modeling languages and modeling tools that we believe are useful to achieve our objective. Next, we present the followed methodology to develop the models. Afterwards, we show the models that we have developed until now. Then, we present a brief discussion of these results. Finally, we make a conclusion about the work done and we expose the perspectives of our future work.

---

[1] http://www.plm.automation.siemens.com/en_us/

2. Scope, approach and limits

This section establishes the scope, approach, and limits of our research. The scope's definition will allow identifying two important aspects: (1) what concerns this investigation and (2) what doesn't. In order to achieve to make this distinction, it is important to establish clear and understandable limits according to our objectives. In addition, the way in which the problem is addressed (the approach) has to be clear. This description is done in the next paragraphs.

In this work, a review of selected topics related to New Product Development (NPD) and information representation is done. On the side of NPD, there are themes such as information and communication technologies, manufacturing approaches, innovation, theoretical approaches, and Systems Engineering. On the side of information representation we will principally be dealing with modeling languages and modeling tools.

The approach to carry out this research is from the point of view of Systemics or Systems thinking. A brief description of these topics is done in the next section of the paper. What is important now, is to make clear that what we are trying to do is to make evident the lack of characterization of individualized products and processes. They are being developed out there, and there is proof of it. But it is being done without any formalized knowledge of what they are, what they imply, what are their advantages and disadvantages, when they are more likely to be used and when they are not, etc. Therefore, although the collection of subjects may seem too wide at first glance, is just to show the extent that product and process individualization can reach.

Hence, it won't be proposed a specific way to treat and solve the problem of individualization, but we are going to try to collect the information that we believe has more to do with it, to put it in order and to propose a perspective to address the problem. We think that it will help to deal with the high level of complexity that individualization is generating.

Now, as mentioned in the previous section, all products have a lifecycle (Norm ISO 15288, 2008). The most critical phases of a product's lifecycle are the first ones, where it is defined. Indeed, our research sets on these first phases, where innovation processes for the product or for the production system to make it are more likely to be triggered. These first phases have the highest level of complexity (Weigt, 2007). It is in the innovation processes that can take place during these phases that a different modeling perspective to better understand individualized products and processes will be proposed. In doing so we hope it'll help to better handle the complexity.

3. State of the art

This section of the paper describes in a succinct manner the topics that we believe can lead us to build a coherent formalization of individualization. Firstly, in order to avoid ambiguity, the elements that will be used along the article will be described. We take back the summary of these terms from (Lhoste, 1994).

- **Concept**: *general and abstract idea*, or even, *general idea that exists only at the occasion of experience, not proceeding from experience but from the mind (according to Conceptualism, philosophical doctrine originally from Abélard)*
- **Theory**: *Intellectual, methodical, organized and systematized construction of hypothetical and synthetic character. Collection of rules or laws on which a science is based and from which many phenomena can be explained*. This notion of theory will be used to precise a modeling framework.
- **Model**: *Approximation, more or less abstract partial view of a part of the real world according to a given point of view, in order to understand it easily. It can be expressed in terms of mathematics, symbols, words, etc*. This notion of model is understood as a partial view of a system apprehended by means of a theory, eventually associated to a method and expressed by a language.
- **Method**: *Approach of the mind to discover and demonstrate the Truth. Reasoned approach to achieve a goal*. The notion of method is used to define the process leading to model creation.
- **Methodology**: *Coherent collection of methods carried out in a coordinated way in order to achieve a complex goal*. This definition doesn't correspond to the usual definition of "methodology". Nevertheless, we allow ourselves this language abuse to express the existence of methods that organize other methods.
- **Language**: *All means to communicate thoughts on the base of a signs system*. The language is considered as the means to express a model in order to allow its exploitation or the communication of its message.
- **Tool**: *Means to carry out a task*. The notion of "tool" is used to design a means (often software means) that allows supporting the application of methods, model manipulation and its associated languages.

What is presented next is a brief review of the topics that we consider are more related with product and process individualization. For reasons of space-time, just a general overview about them is given. However, we try to be as concise as possible to show the relationship among them and individualization.

3.1 General overview

In order to get a better insight of the topics, we propose firstly to take a look at Figure 1. In this figure, the subjects are enclosed in such a way to facilitate their description within the next pages and the relationship among them. The aim of this section is to present the state of the art as a construct to support the proposed models of section 4.

3.1.1 Systemics

The approach used to develop this formalization proposal is Systemics, also called Systems Science, Systems thinking or Systemism (Bunge, 2000), and it concerns research related to General Systems Theory (Bertalanffy, 1976). According to Systemism, everything is either a system or a component of a system, and every system has peculiar (emergent) properties that its components lack. The term system refers to entities, which are not to be considered merely sum of their parts, but require a holistic approach for reasons of interrelations between these parts (Weigt, 2007). A system is defined as a set of elements and their relations. Inputs and outputs connect a system to its environment, from which a system boundary separates it (Daenzer, 2002).

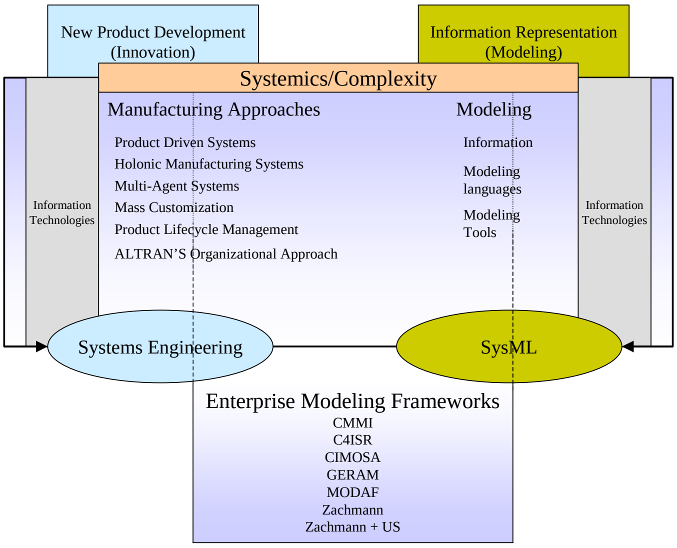


Figure 1. Overview of the concerned topics to formalize individualized products and processes

3.1.2 Systems Engineering

Systems Engineering (SE) is a practical application of system theory, and it is often justified by its providing of effective means to handle complexity (Daenzer, 2002) (Doerner, 1989). Complexity in a narrower sense (i.n.s.) refers to a property of a system model, resulting from type and number of elements and relations each. Complexity in a broader sense (i.b.s.) refers to a property of a situation in which complexity i.n.s and further complication situational factors (such as dynamics, intransparency, ignorance, and false hypotheses) confront an agent. Complexity i.n.s and i.b.s. is therefore subjective: It requires the existence of an observer, and depends on the purpose of system modeling and on knowledge and experience of the agent in the context of similar system models and situations (Doerner, 1989) (Gomez, 1981).

Several factors cause complexity i.b.s. in early stages of innovation processes, i.e. product planning and development. Important reasons are functional plurality and product **individualization** required for competitive reasons (Weigt, 2007). Requisite model and option diversity calls for measures like size ranges, modular products, or package systems, increasing complexity i.n.s. of technical products (Ehrlenspiel, 2003) (Pahl, 1996). Also related to the competitive environment is increasing interdisciplinarity of technical products. Furthermore, engineering design objectives can be undetermined, generic, unclear, implicit and conflicting (Doerner, 1989) (Pahl, 1996).

SE allows the effective organization of resources, equipment, actors, stakeholders, and information flow to better deal with complexity. In SE, activities are regrouped according to the relationship they have with two main concepts: activities concerning the "product system" and activities concerning the "program system". The first one is a technological system that comprises the collection of materials, software, human competencies, and processes organized to fulfill a given need. The second one is an organizational system that comprises the collection of equipment, methods, processes and means organized to fulfill the need of conception, evolution and verification of the "product system" (Meinadier, 2002). Using SE, the elements concerned in the new product development have something to do with at least one of them.

One of the biggest concerns of the new product developer is to be sure to build the right product and to build it correctly. SE uses a methodology known as the V cycle. During the descending part of the V, requirements are defined iteratively, beginning with the clients needs, and refining them until getting the specification of every component of the product system, at the bottom of the V. Then the components are built. The second half of the V is called the IVVQ process (Integration, Verification, Validation, and Qualification). *Integration* is when the components are assembled to compose the system. Then, at each step it is *verified* according to the technical requirements, to show that it is properly conceived and constructed. Then, the product system is *validated* according to the client's requirements, to show that it fulfils its needs. Finally, it is *qualified* in its operational environment, showing its capability to respond to its operational purpose. The base of the correct performance of the product-system is the good definition of its requirements, from the point of view of all stakeholders.

The notion of requirement is an important issue for the development of this work, because an individualized product is also created following requirements. A **requirement** is a sentence that translates or expresses a specific need in terms of constraints (technical, costs, deadlines…). This sentence is presented using a modeling language. Within the next sub-section the notion of modeling language is described.

3.1.3 Modeling

At the beginning of this section the concept of model was presented. Models are used to simplify a part of the real world in order to simplify its comprehension. Modeling is the activity of building models, and it isn't a simple task because it implies knowing how to represent information in the best way in order to better communicate it. Models are always determined by someone. A system is always a model made by an observer of an original, but not the original itself (Gomez, 1981). System determination depends on the observer's frame of reference. Observers often do not discriminate between systems and objects. A broad concept of model includes every purposeful, material or immaterial abstraction of a material or immaterial original. System models in a stricter sense are mental interpretations, and as such can only be immaterial abstractions of material or immaterial originals.

A modeling language is a formalism to represent and communicate information in a standardized way. There are three kinds of modeling languages: graphical, mathematical, and discursive (Durand, 1979). The first ones concern successions and arrangements of lines, figures, points, and colors that have a meaning in a certain context. The second one is the way in which mathematics is expressed, using universally defined symbols and in which most scientific work is required to be presented. The third one consists of arranged lines in which a spoken language is represented on a background, in other words: text.

In this work, three modeling languages are used. The first one is the Natural language Information Analysis Method (NIAM). NIAM is a modeling language that permits to formalize knowledge expressed in natural language in order to describe the behavior of real objects using stable rules (Habrias, 1988). The second one is the Unified Modeling Language (UML) (OMG, 2004). UML is a standardized visual specification language for object modeling that includes a graphical notation used to create an abstract model of a system. The Unified Modeling Language is an international standard: ISO/IEC 19501:2005 Information technologies — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2. Finally, the third language is the Systems Modeling Language (SysML). It is a recent formalism whose first Available Specification was released in September 2007. It was conceived by the Object Management Group (OMG)[2] in collaboration with the International Council of Systems Engineering (INCOSE)[3]. It is aimed to be a means to model a broad range of systems. In fact, SysML is defined as an extension of a subset of UML, using the same profile mechanism. SysML is aimed to be used by systems engineers; its main improvement is the use of the requirement diagram and the parametric diagram. SysML is able to model a wide range of systems, which may include hardware, software, information, processes, personnel, and facilities. SysML furnishes flexible allocation tables that will support requirements allocation, functional allocation, and structural allocation. This capability facilitates automated verification and validation (V&V) and gap analysis. For more information about the differences and similarities between UML and SysML, see (Willard, 2007).

The software used to carry out the models in the different languages is described next. For NIAM, *Microsoft Visio* is used. This software was originally developed by the Axon Corporation whose principal product was a diagramming application software for Object-Role Modeling (ORM)[4], which shares the same principles as NIAM. Afterwards, the Axon Corporation became the Visio Corporation, which was finally acquired by Microsoft in 2000. *MS Visio* uses vector graphics to create diagrams, and so it facilitates the construction of NIAM models. For UML, the software *MEGA International*[5] is used. *MEGA* is a powerful software solution for enterprise modeling. It allows process definition, operations control and risks government in order to optimize enterprise processes. It is based in UML and the *Zachman Framework*, which will be described in a further subsection. Finally, for SysML the used software is *TOPCASED* [6] (Toolkit in OPen-source for Critical Application & SystEms Development). The TOPCASED Project was conceived at Airbus France. Its main objective is to make available to the Open Source community a set of systems engineering tools that cover the descending half of the V cycle along with its transversal activities. These tools have to be adapted to critical systems development constraints (such as requirements), mature and open enough to respond to the modeling needs.

3.1.4 Manufacturing approaches

Manufacturing systems have significantly evolved since the Industrial Revolution. Nowadays, there are several factors that demand different manufacturing approaches to remain competitive face to global competition. Some of these factors are (HMS, 1994) (Shen and Norrie, 1999):

---

[2] http://www.omg.org/

[3] http://www.incose.org/

[4] http://www.objectrolemodeling.com/AboutORM/History/tabid/48/Default.aspx

[5] http://www.mega.com/

[6] www.topcased.org

(i) *Enterprise integration*: in order to support global competitiveness and rapid market responsiveness.
(ii) *Distributed organization*: for effective enterprise integration across distributed organizations.
(iii) *Heterogeneous environments*: such manufacturing systems will need to accommodate heterogeneous software and hardware in both the manufacturing and information environment.
(iv) *Interoperability*: a heterogeneous information environment may use different programming languages, represent data with different representation languages and models, and operate on different computing platforms.
(v) *Open and dynamic structure*: It must be possible to rapidly integrate new subsystems into or remove existing subsystems from the system without stopping or reinitializing the working environment.
(vi) *Cooperation*: manufacturing enterprises will have to fully cooperate with their suppliers, partners, and customers for materials, supplies, parts fabrication, final product commercialization, and so on.
(vii) *Integration of humans with software and hardware*: people and computers need to be integrated to work collectively at various stages of the product development and even the whole product lifecycle.
(viii) *Agility*: agile manufacturing is the ability to adapt quickly in a manufacturing environment of continuous and unanticipated change and thus is a key component in manufacturing strategies for global competition.
(ix) *Scalability*: Expansion of resources should be possible without the disruption of previously established organizational links.
(x) *Fault tolerant*: the system should be fault tolerant both at the system level and at the subsystem level so as to detect and recover from system failures at any level and minimize their impacts on the working environment.

These requirements for manufacturing systems have triggered research to deal with the challenge. Many manufacturing paradigms promise to meet these challenges. Two of these paradigms, distributed intelligent manufacturing systems (or agent-based manufacturing systems), and holonic manufacturing systems (HMS) have received a lot of attention in academia and industry (Giret and Botty, 2004).

Nowadays, the competitive situation of companies is characterized by a very strong orientation towards **product individualization** (Blecker *et al.*, 2005). Mass customization is a business strategy that aims at fulfilling individual customer needs with near mass production efficiency (Pine, 1993). The challenge that manufacturing companies have to face is to provide individualized products and services by maintaining a high costs' efficiency. In order to do that, the factors mentioned above could be used to achieve such goal.

Another approach that has derived from these needs is the "Product-Driven Systems". This approach is based on the intensive deployment of Information Technologies in manufacturing systems. It takes into account the make-to-order business models and the mass customization of products. The concept of product driven control considers the product as pivotal to the automation rationale. This approach consists in providing the product with information, decision and communication capabilities in order to make it active in the scheduling and the execution of its manufacturing operations (Gouyon *et al.*, 2007) (Wong *et al.*, 2002).

RFID (Radio Frequency Identification) is the Information Technology in which product-driven control is based (McFarlane *et al.*, 2003). This technology is described in the following section.

3.1.5 Information technologies

Coordination among the elements of an enterprise is achieved through effective communication. Communication is the process of transferring information from a sender to a receiver with the use of a medium in which the communicated information is understood by both sender and receiver (Severin and Tankard, 2000). In an enterprise, the New Information and Communication Technologies (NICT) have allowed a continuously increasing coordination for product manufacturing. However, complex systems (both products and enterprises) still make it difficult to achieve a better improvement of manufacturing processes. One of the issues that have to do with coordination among the elements of an enterprise is product control. Product control has been improved thanks to PLM. This strategy aims to integrate product information within an information system all through its life cycle. This allows a permanent state of communication among all (or nearly all) the participants in the lifecycle of a given product. Several vendors have developed different kinds of software to support PLM (Terzi *et al.*, 2006):

(i) Vendors coming from the digital engineering world (UGS PLM Solutions[7], Tecnomatix, IBM-Dassault), which start from PD (Product Development) and MSE (Manufacturing System Engineering) processes and are trying to connect Enterprise Engineering and Management processes.
(ii) Vendors coming from the ERP (Enterprise Resource Planning) world (Baan, SAP, Oracle), which, at the contrary, start from Enterprise Management processes for turning to connect PD/MSE tools and platforms.
(iii) Vendors coming from the ICT world, dealing with architecture and platforms, which aim to establish such collaborative environments for PLM integration (Microsoft, MatrixOne, Agile), basically using web technologies.

Information management is the core of product and production management, that’s why the related emerging market is ICT characterized. Nevertheless, PLM is not primary an ICT problem, but at first, is a strategic business orientation of the enterprise (Gareti and Terzi, 2004).

Another way to keep control of the product is to give it the control, and verify its performance. This is the approach of “Product-Driven Systems”, which is based on RFID Technology. At its most simple, a radio frequency identification (RFID) system consists of two components, namely a tag (also called a transponder) and a reader (also called an interrogator). The tag is designed to be small and cheap – perhaps the size of a credit card or smaller – while the reader is more expensive and larger. The RFID tag contains a small amount of memory for holding data, and whenever that tag comes into proximity with the RFID reader, the reader will detect the tag’s presence and can read the data from the tag.

---

[7] Now is Siemens PLM Software http://www.plm.automation.siemens.com/en_us/

There are many different types of RFID systems that vary in their exact mode of operation and operating performance. With 'active' RFID systems, the tag contains a small battery that enables it to control communication with the reader. A completely 'passive' RFID tag, however, has no battery but instead harvests power for its operation from the reader's radio communication signal. This means that the reader has to drive communication, but it makes the tag much cheaper.

This capability of the product to become an actor, and having decision and communication abilities in the scheduling and the execution of its manufacturing operations is one of the main issues of this work. This will allow discussing, along with other related topics, the proposition of a different perspective to address the problem of individualization on the early stages of product innovation. Our main concern is to characterize product individualization from both aspects: product customization and product "control" of the manufacturing system. We expect this to have an impact in enterprise modeling.

3.1.6 Enterprise modeling

Any organization has an Information System (IS) supporting the business. There is no doubt that IS supports decision-making, coordination, and control and may also help managers and workers analyze problems, visualize complex subjects, and create new products (Laudon *et al.*, 2000). According to a study conducted in the 1990's (Niederman *et al.*, 1991), Information Architecture was referred to as one of the most important issues to address in IS management and as stated by Zachman, "with increasing size and complexity of the implementation of information systems, it is necessary to use some logical construct (or architecture) for defining and controlling the interfaces and the integration of all of the components of the system" (Zachman, 1987).

Taking this into consideration, it is necessary to define an Enterprise Architecture in an organization to gain the associated advantages of that architecture, therefore, it is a better option to simplify the IS management. Some architecture's advantages are (Marques and Sousa, 2004):

(i) It acts as a way to pass from chaos and disagreement to order and structure.
(ii) It enables an integrated vision and a global perspective of informational resources.
(iii) It enables the discovery and elimination of redundancy in the business processes reducing information systems complexity.
(iv) It contributes to having information systems that reflect common goals and performance measures for all managers, to encourage cooperation rather than conflict and competition within organizations.
(v) It becomes the bridge between the business and technical domains.

There are several Architecture Frameworks for organizations. It is not in the scope of this paper to consider them all in this. Some of them are:

(i) The US Department of Defense (DoD) Command, Control, Communication, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework (AF) (C4ISR, 1997)

(ii) The Generalized Enterprise Reference Architecture and Methodology (GERAM) Framework (ISO/TC184, 2000)
(iii) The Zachman Framework for Enterprise Architecture (Zachman, 1987)

The Zachman Framework[8] has been chosen to support the proposed models of section 4. Its aim was described as an architecture that represents the information systems' artifacts, providing a means of ensuring that standards for creating the information environment exist and they are appropriately integrated. It proposes a logical structure for classifying and organizing the descriptive representations of an enterprise, in different dimensions, and each dimension can be perceived in different perspectives. The Zachman Framework helps govern the architectural process with the dependency, coherence, and traceability needed for an enterprise to manage change, and to ensure that the alignment is achieved. In this framework, the architecture is described across two independent aspects, the rows represent the different perspectives which may be used to view a business, a situation, an opportunity, or a system and, the columns represent the different dimensions which apply to each perspective of the business, situation, opportunity, or system.

4. Methodology, models and discussion

The reviewed topics have one thing in common: product creation. Using the systemic approach, Systems Engineering turns out to be one of the best tools to manage complexity in product development. It can be used within the proposed manufacturing approaches, modeling languages and information technologies. In addition, it uses a primordial concept to develop the right product in the correct way: requirements. To create an individualized product that will satisfy a specific client, requirements are also the mainline. If we make the hypothesis that a product is able to "control" the system that is going to produce it, since the first phases of its lifecycle, what is the standpoint with which we should address a modeling framework? First of all, it should be defined what an individualized product is. A precise description that is somewhat wrong is better than a description so vague that no one can tell if it is wrong. In the next paragraphs, a preliminary formalization for an individualized product is proposed.

4.1 Characterization of an individualized product

In (Blecker *et al.*, 2005) some interesting concepts were found concerning what the client wants. These concepts are: *primary needs* and *secondary needs* as well as *objective needs* and *subjective needs*. "*The customers' subjective needs are defined as the individually realized and articulated requirements whereas the objective needs as the real ones perceived by a fictive neutral perspective. Using knowledge management terms, we argue that the subjective needs are explicit, while the objective needs are implicit*" (Blecker *et al.*, 2003)

Muther (2000) specifies the need of differentiating between *primary needs* and *secondary needs*. The customer is generally motivated by a primary need, which triggers a secondary need for certain products or services.

---

[8] www.zifa.com

Table 1. Relationship between the customer's needs and an individualized product's aspects.

Customer's needs		Product aspect	
Objective	Primary	Must-modules	Essential
Subjective	Secondary	Can-modules	Inessential
Blecker et al, 2003	Muther, 2001	Rosenberg, 1996	Simondon, 1958

These described needs allowed to define the aspect of the individualized product's structure. In the philosophical domain, it was found that there are two aspects that comprise every product: the essential aspect and the inessential aspect (Simondon, 1958). The first one includes all of the product's characteristics that give it its identity as the object that it is. In other words, within the essential aspect we find the characteristics that define its main function. The second aspect contains the characteristics that "*can be tailor-made, because they are incidental*" (Simondon, 1958, p.24). A first formalization of this structure has been carried out using NIAM (Figure 2)

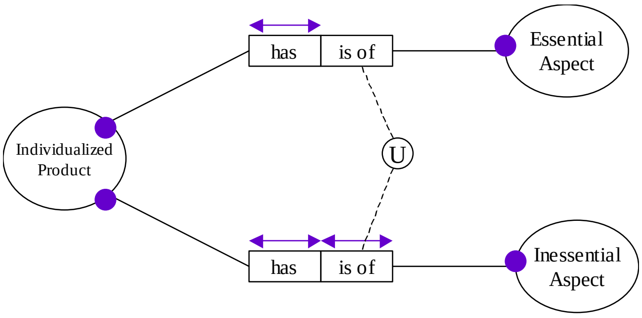


Figure 2. Individualized product model structure in NIAM.

Researches in Mass Customization (Blecker *et al.*, 2005) allowed finding a link between the essential and inessential aspects of the product structure, the product requirements and the clients' needs. At this time, we have focused on developing models about the product. By following the methodological approach of Systems Engineering (SE), it has been possible to build a preliminary product requirement model. In a similar way to the aspects of the product structure, product requirements can be classified in "must" requirements and "can" requirements (Rosenberg, 1996) (Figure 3).

The first ones must be accomplished in order to give the product its basic functionality. These requirements have a relationship with the essential aspect of the product structure. On the other hand, "can" requirements are the ones that could be accomplished, but not necessarily. They give the product other functionalities besides its main one, but their absence does not twist its main functionality. These requirements have a relationship with the inessential aspect of the product structure. In order to formalize this requirements model, we used UML. This modeling language has more appropriate modeling tools. Yet, NIAM will keep being used to identify the "right" objects to be modeled.

It is important to remark that the notion of *essential requirements* of an object has been identified in other areas, see (Chapman and Hyland, 2004). The subjectivity of *essential* and *inessential* notions makes it hard to determine a stable definition to say when a given feature can be essential or inessential. This means that it will change according to the point of view of each stakeholder. What is constant is their presence in every object.

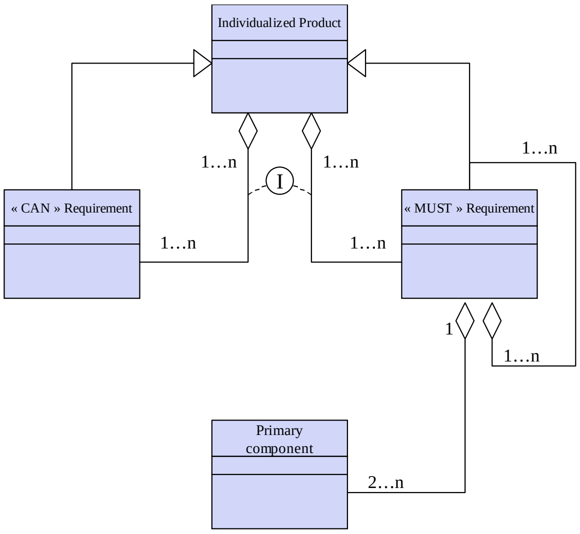


Figure 3. Individualized product's requirements model

SysML offers an interesting perspective to characterize the individualized product. As mentioned in section 3.1.3, SysML retrieves some of the diagrams of UML and adds two more: the requirements diagram and the parametrics diagram. These two diagrams, together with the behavior and structure diagrams, conform the four pillars of SysML. Utilizing this modeling language and the essential and inessential introduced aspects for an individualized product, the following approach in SysML is proposed (Figure 4):

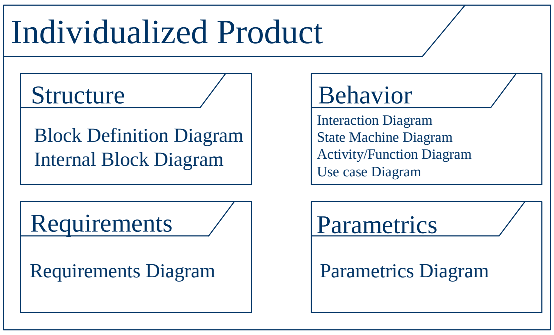


Figure 4. Modeling approach of the individualized product in SysML.

Definition of "must" and "can" requirements will have an impact in the structure, behavior and parametrics of the individualized product. This means that the product's structure will also have its "must" and "can" blocks to satisfy any given "must"/"can" requirements. In the same way, the product will have a "must" and a "can" behavior, referring to the functionalities that it must absolutely have, and the ones that it can or cannot have, respectively. This depends on both the product's structure and the requirements. Finally, in the parametrics diagram, which is used to represent usage with given values, there will also be "must" and "can" values, according to the wanted functionality of the product. This is why SysML is adequate to describe the individualized product, because it helps to link the relationships between the kinds of customers' needs (objective/subjective, primary/secondary) with the product's structure, product's requirements, product's behavior and its parametrics (the "must" and "can" features).

From the point of view of systemics and the "Product-Driven Systems" approach, the product has capabilities that allow considering it as a system. It has within a decisional system, an informational system, and an operational system. In this context, during production processes there will be two interacting systems. Calling back the concepts from Systems Engineering, those are the "product system" and the "program system". This interaction brings the individualized product characterization to a higher level. The fact of it communicating, getting informed, making decisions and operations individually requires a different approach to consider it. For this reason, an interaction matrix between the "product system" and the "program system" is presented. It will allow identifying and characterizing the kinds of interactions taking into account the decisional, informational and operational systems.

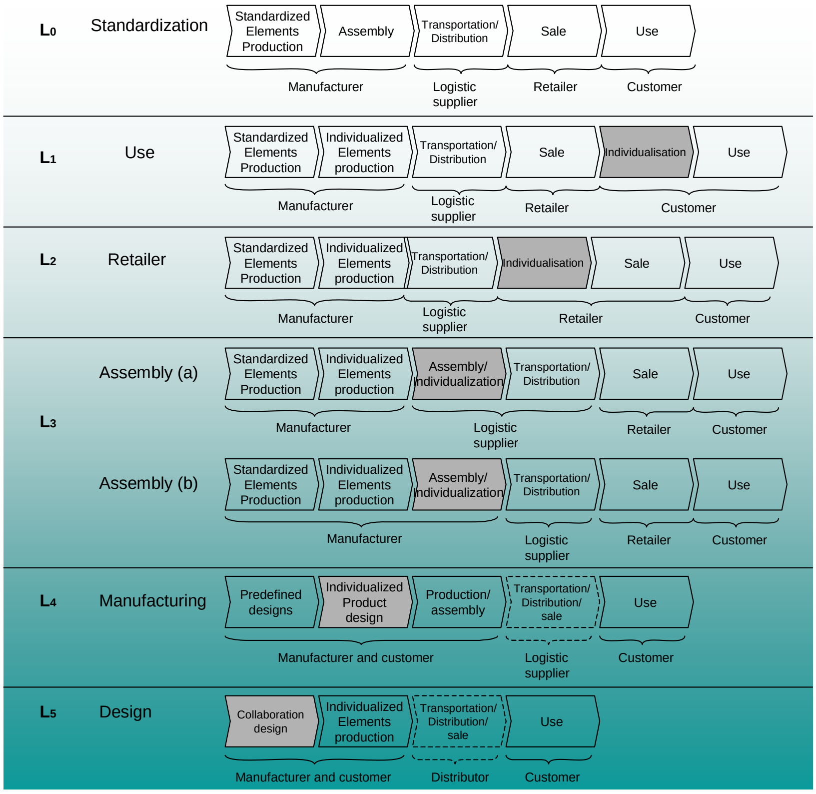
Program System Product System	D	I	Φ
D	D-D	I-D	Φ-D
I	D-I	I-I	Φ-I
Φ	D-Φ	I-Φ	Φ-Φ

Figure 5. Proposed matrix to identify "product system" - "program system" interactions.

4.2 Individualization degrees of a product

Along this article, it has been shown the generalized tendency that exists in the market to regard the needs of individual costumers during product development. In this perspective, one can say that an individualized product is also the one that has an intervention of the customer before being used. Depending on the technological resources and on the nature of the product, such an intervention could happen at different stages of the value chain: a few moments before using the product, when it is bought, when it is assembled, when it is fabricated, or when it is conceived. The more the customer takes part in the first phases of the value chain, the more it can influence the essential aspect of the product. This situation introduces the notion of individualization degrees of a product, depending on the stage where the client gets involved with its development. We believe that such client interventions belong to a set of factors that can trigger innovation processes. However, we haven't found a reference model to link a product's individualization degrees with the capability of the enterprise to create a product with a certain individualization degree. We think that an approach to measure this capability is needed, in analogy to the CMMI (Capability Maturity

Model Integration) (CMMI, 2006). The CMMI is a model to measure the maturity level of sound practices within an enterprise's processes. In a similar way, we propose a reference model to measure the maturity level of an enterprise to create individualized products. This model is one of the main objectives of our work, because it will complete the characterization of an individualized product and of the processes (in Systems Engineering) to be carried out to fabricate it, particularly during the innovation process. This characterization will permit a better understanding of several factors that are directly related with the manufacturing of individualized products, e.g. product comprehension for a higher process optimization, the manufacturing system's requirements, the innovation approach to carry out if needed, complexity management.



No indiviudalization

No direct interaction between the customer and the value chain stages

Indiviudalization is done when the customer has the physical product, manually or automatically, depending on the sophistication level of the product.
the manufacturer takes into account certain needs of the customer

Individualization takes place at the retailer stage. It can Be on the product or on the services with which the product is sold. At this level, individualization is rather A service that doesn't concern the product so much

Level 3 retrieves the notion of individualization during Assembly at the logistic supplier stage or at the manufacturing stage. It is possible that at this level the retailer is less important. This level of indiviudalization can comprise the packaging and distribution of the product. The use of standardized moduls is obvious. The customer can choose between must modules and can modules.

At this level, the approach is similar to the one proposed in [Duray et al, 2001].
There is a high level of customer intervention.
He can choose among predefined designs and he can modify them respecting the constraints of essential and inessential aspects of the product (costs, processes, etc)
There is no need of a retailer

At this level is found the highest level of individualization. The product is coneived 100% to satisfy a specific customer's need. This level of individualization can even conncern the product's components.
Projects such as « Solar Impulse » are found at this level of individualization, were innovations to better respond to the needs of each component can be developed.

Figure 4. Proposition of six levels of product individualization

5. Conclusions and perspectives

In this article we highlighted two important aspects that innovation has to take into account: large product variability and product customization. This generates an enhancement of complexity along the product's lifecycle. There are several approaches to handle it. The proposed angle of this article considers addressing the complexity problem from the point of view of Systemics. In this context, several approaches have been reviewed. Systems Engineering and product-driven control are the chosen solutions that, supported by Information Technologies, have allowed to introduce a perspective to treat the individualized product information management problem at the early stages of its lifecycle. We postulate that the individualized product can be regarded as a system. This perspective requires firstly formalizing and characterizing it. To do this, we proposed five preliminary models. The first one is a model concerning the structural aspects of the product. This model aims to suggest that two kinds of aspects compose an individualized product: essential and inessential aspects. The second model concerns the product requirements. This model's goal is to propose that, in a similar way to the first model, there can be two types of requirements: "must" and "can" requirements. We postulate that there is a relationship between the structural aspects and the kinds of requirements. The third model is a proposition to use SysML's four main diagrams to start a more formal characterization of the individualized product. Such diagrams that SysML uses to describe a system are: requirements, behavior, structure and parametric.

We used the concepts of Systems Engineering, "product system" and "program system", to refer to the product and its production system, respectively. Considering the hypothesis that a product can be regarded as a system, an interaction matrix between them has been proposed. Its aim is to identify and characterize the kinds of interactions between both systems.

The fourth model aims to measure the individualization degree of the product. Six individualization levels were proposed, starting at level 0, where no individualization is identified, until level 6, where the highest degree of individualization is found. We believe that the fact of considering such degrees belong to a set of factors that can trigger innovation processes.

We contemplate refine these models in future work. Also, we consider proposing an innovation reference model taking into account these presented models. We will also work on an approach to link the product and the manufacturing system, basing us on the Zachmann Framework. We contemplate to use the "must" and "can" requirement aspects and the modalities of a production system ("have to", "want to", know-how to", "able to") (Mayer, 1995) to propose, within a coherent model, a formalization of individualized processes in Systems Engineering. Finally, we'll search to prototype the proposed models within an enterprise that has close contact with its customers. An RFID Toolkit will be used in order to instantiate the individualized product. In doing so, it is expected to validate the proposed models.

6. References

Bannert, V (2004). Managing Corporate Acquisitions in Innovation Driven Industries, PhD Thesis, Swiss Federal Institute of Technology

Bertalanffy, L von (eds.) (1976). *General System theory: Foundations, Development, Applications,* New York: George Braziller

Blecker, T, Abdelkafi, N, Kaluza, B, Friedrich, G (2003). Variety Steering Concept for Mass Customization, Working Paper No. 2003/04, University of Klagenfurt.

Blecker, T, Abdelkafi, N, Kaluza, B, Friedrich, G, Kreutler, G (eds.) (2005). *Information and Management Systems for Product Customization*, Boston: Springer Science

Bunge, M (2000). Systemism: the alternative to individualism and holism. *Journal of Socio-Economics*, 29, 147–157

C4ISR (1997). Architectures Working Group, Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance—C4ISR (v. 2.0) US DoD

Chapman, R and Hyland, P (2004). Complexity and behaviors in product innovation. *Technovation*, 24, 553–561

Cheng, ML and Simmons, JEL (1994). Traceability in manufacturing systems. *International Journal of Operations and Production Management*, 14, 4–16 (in Terzi et al, 2006)

CMMI Product Team (2006). *CMMI for Development version 1.2*, Carnegie Mellon, Software Engineering Institute. Available at: http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html (vu le 22/01/08)

Daenzer, WF, Huber, F (eds.) (2002). *Systems Engineering*, Industrielle Organisation, Zurich (in Weigt, 2007)

Doerner, D (eds.) (1989). *Die Logik des Misslingens*. Rowohlt, Reinbek (in Weigt, 2007)

Durand, D (eds.) (1979). *La Systémique*, Presses Universitaires de France, 10ème édition, octobre 2006

Ehrlenspiel, K (eds.) (2003). *Integrierte Producktentwicklung*. Hanser, Munich (in Weigt, 2007)

European Union (1995), Green Paper on Innovation. European Commission, Brussels (in Galanakis, 2006)

Fisher, ML, Hammond, JH, Obermeyer, WR, Raman, A (1994). Making Supply Meet Demand in an Uncertain World, *Harvard Business Review*, May-June

Franke, N, von Hippel, E, (2003). Satisfying Heterogeneous User Needs via Innovation Toolkits: The Case of Apache Security Software, *Research Policy*, 32, 1199–1215

Galanakis, K, (2006). Innovation process. Making sense using systems thinking. *Technovation*, 26, 1222–1232

Garetti M, Terzi S, (2004). Product Lifecycle Management: definition, trends and open issues. In *Proc. 3rd Int. Conf. On Advances In Production Engineering*, 17 - 19 June, Warsaw, Poland (in Terzi et al, 2006)

Giret, A, and Botti, V (2004). Holons and Agents. *Journal of Intelligent Manufacturing*, 15, 645–659

Gomez, P (1981). *Modelle und Methoden des systemorientierten Managements*. Haup: Bern (in Weigt, 2007)

Gouyon, D, Pétin, JF, Morel, G (2007). A product driven reconfigurable control for shop floor systems. *Studies in Informatics and Control*, 16

Habrias H, (eds.) (1988). *Le Modèle Relationnel Binaire*. Méthode I.A. (NIAM), Editions Eyrolles

HMS, PR (1994). *HMS Requirements*. http://hms.ifw.uni-hannover.de/:HMS Server (in Giret and Botty, 2004)

ISO/TC184 (2000). ‘‘Annex A: GERAM’’, ISO/DIS 15704: industrial automation systems requirements for enterprise-reference architectures and methodologies

Jansen-Vullers, J, van Dorp A, Beulens B (2003). Managing traceability information in manufacture. *International Journal Of Information Management* 23, 395–413 (in Terzi et al, 2006)

Kärkkäinen J, Holmström G, Främling J, Artto G (2003). Intelligent products – A step towards a more effective project delivery chain. *Computers in Industry*, 50, 141–151 (in Terzi et al, 2006)

Laudon, KC, and Laudon, JP (2000). *Management Information Systems: Organization and Technology in the Networked Enterprise*, Prentice-Hall, 2000 (in Marques and Sousa, 2004)

Lhoste, P (1994). *Contribution au génie automatique: concepts, modèles, méthodes et outils*, HDR, Université Henri Poincaré.

Marques, C, Sousa, P (2004). A method to define an Enterprise Architecture using the Zachman Framework, Symposium on Applied Computing, In *Proc of the 2004 ACM symposium on Applied Computing*, 1366–1371

Mayer, (1995). *Contribution to manufacturing engineering: Application to pedagogical engineering within a CIME centre*. Ph.D. thesis, CRAN-UHP (in French).

McFarlane D, Sarma, J, Chirn, G, Wong, J, Ashton, A (2003). Auto-ID systems and intelligent manufacturing control, *Journal of Engineering Applications of Artificial Intelligence*, 16, 365–376 (in Terzi et al, 2006)

Meinadier, JP (eds.) (2002). *Le métier d'intégration des systèmes*. Paris : Hermes Science Publication

Mitterdorfer-Schaad, DD (2001). *Modellierung unternehmensspezifischer Innovations-Prozessmodelle*. ETH, Zürich (in Bannert, 2004)

Morel, G, Panetto, H, Zaremba, A, Mayer, F (2004). Manufacturing enterprise control and management system engineering rationales and open issues, *IFAC Annual Reviews in Control*, 27, 199–209

Muther, A (eds.) (2000). *Electronic Customer Care: Die Anbieter-Kunden-Beziehung im Informationszeitalter*, Berlin: Springer (in Blecker et al, 2005)

Norm ISO/IEC 15288 (2008), *International Standardization Organization*

Niederman, F, Brancheau, JC, and Wetherbe, JC (1991). Information Systemns Management Issues for the 1990s. *MIS Quartely*, 14 (4) (in Marques and Sousa, 2004)

Organization for Economic Co-operation and Development (1997a). *National Innovation Systems*. Paris: OECD (in Galanakis, 2006)

Organization for Economic Co-operation and Development (1997b). *Proposed Guidelines for Collecting and Interpreting Technological Innovation Data*. Oslo Manual, OECD, Paris (in Galanakis, 2006)

Object Management Group (2004), *UML 2.0 Superstructure, Available Specification*, OMG document ptc/04-10-02, 2004. Available at: http://www.ibm.com/developerworks/rational/library/05/321_uml/ (vu le 14/04/08)

Pahl, G, Beitz, W (eds.) (1996). *Engineering Design*. Berlin: Springer (in Weigt, 2007)

Pine II, BJ (eds) (1993). *Mass Customization: The New Frontier in Business Competition*, Boston: Harvard Business School Press

Rosenberg, O (1996). Variantenfertigung, in *Handwoerterbuch der Produktionswirtschaf*t, (ed.), Stuttgart: Schaeffer-Poeschel, 2119-2129.

Rothwell, R (1994). Towards the fifth-generation innovation process. *International Marketing Review*, 11 (1), 7–31 (in Galanakis, 2006)

Scott, JE (2000). Facilitating Interorganisational Learning with Information Technology, *Journal of Management Information Systems*, 17 (2), 81–113 (in Chapman and Hyland, 2004)

Severin, WJ, and Tankard, JW (2000). *Communication Theories: Origins, Methods and Uses in the Mass Media*, Allyn & Bacon

Simondon, G (eds.) (1958). *Du mode d'existence des objets techniques*, Paris: Aubier

Shen, W and Norrie, D (1999). Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems, an International Journal*, 1(2), 129–156 (in Giret and Botty, 2004)

Stark, J (eds.) (2004). *Product Lifecycle Managment: 21st Century Paradigm for Product Realisation*. Springer – Verlag London Limited

Terzi, S, Cassina, J, Panetto, H (2006). Development of a Metamodel to Foster Interoperability along the Product Lifecycle Traceability. *Interoperability of Enterprise Software and Applications*. Springer – London, 1–11

Weigt, M (2007). Implications of Complexity in Early Stages of Innovation Processes for the Definition of Heuristic Engineering Methods, in *The Future of Product Development*, (ed.) Springer-Berlin Heidelberg

Willard, B (2007). UML for systems engineering, *Computer Standards & Interfaces*, 29, 69–81

Wong CY, McFarlane D, Ahmad Zaharudin AA, Agarwal V (2002). The intelligent product-driven supply chain, *IEEE International Conference on Systems, Man and Cybernetics*

Zachman, JA (1987). A Framework for Information Systems Architecture, *IBM Systems Journal*, 26 (3)