



### **Science Arts & Métiers (SAM)**

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>  
Handle ID: [.http://hdl.handle.net/10985/16756](http://hdl.handle.net/10985/16756)

#### **To cite this version :**

Benoit EYNARD, Thomas GALLET, Pierre NOVAK, Lionel ROUCOULES - UML Based Specifications of PDM Product Structure and Workflow - Computer In Industry - Vol. 55, n°53, p.301-316 - 2004

Any correspondence concerning this service should be sent to the repository

Administrator : [scienceouverte@ensam.eu](mailto:scienceouverte@ensam.eu)



# UML based specifications of PDM product structure and workflow

Benoît Eynard<sup>a,\*</sup>, Thomas Gallet<sup>a,b</sup>, Pierre Nowak<sup>a</sup>, Lionel Roucoules<sup>a</sup>

<sup>a</sup>Laboratory of Mechanical Systems and Concurrent Engineering, Troyes University of Technology,  
12 rue Marie Curie, BP 2060, F.10010 Troyes Cedex, France

<sup>b</sup>Snecma Moteurs, Moissy Cramayel, France

---

## Abstract

The paper deals with the use of a UML approach for the specifications of a PDM system (product data management) implementation. A PDM system enables the management of the whole product data and related information about its entire lifecycle.

The main goal of this paper is to highlight the added value of using an object-oriented approach for modelling, specifying and implementing a PDM system on a business case study. The chosen object-oriented approach and the used UML diagrams for the modelling and integration of product, process, and resource data is detailed for a turbo-prop aircraft project.

*Keywords:* PDM; Object-oriented modelling; Product structure; Workflow; Aeronautics

---

## 1. Introduction

The turbo-prop engine market is increasingly competitive. “Time to market” is becoming one of the most important business and strategic key elements for product development and manufacturing. At the same time, customers want a more customized product with a higher security level. Companies therefore have to reduce design lead-time and provide a better-configured product with an enhanced level of quality and safety [1,2]. These points represent

especially high stakes in turbo-prop manufacturing as it is a niche market; in 2001, the sector total sales were equivalent to the sales of General Motors alone.

### 1.1. Brief PDM overview

A PDM (product data management) system manages and stores product design, manufacturing and support data [3,4]. Two main functionalities of such a system are:

- a product structure manager, which organizes and stores the whole product data. It manages the bills of

---

\* Corresponding author. Tel.: +33 3 25 71 58 28;  
fax: +33 3 25 71 56 75.

*E-mail address:* benoit.eynard@utt.fr (B. Eynard).

material and the various product configurations. It provides functionalities for versioning and linking the various parts and documents in the product structure. It also ensures the management of data status (available, frozen or obsolete);

- a workflow engine which, according to the product structure, sends the right available data at the right time to the right user. This functionality allows the implementation of ISO oriented processes in product development: from the preliminary design phase to the production engineering phase.

Many others functionalities are also currently available, such as integration with a mail server that notifies an event occurring on a given document (i.e. a check-in, check-out, promoting, etc.).

### 1.2. VPM-Chains project

At the end of the 1990s, Snecma Moteurs, one of the world leaders in turbo-prop manufacturing, has chosen to implement a new PDM system called VPM-Chains, mainly based on ENOVIA VPM from Dassault Systèmes. Snecma Moteurs wants to have a huge customization of VPM functionalities in order to integrate the digital mock-up manager with a workflow functionality. Earlier, Snecma Moteurs was using a legacy PDM which did not fulfill the current and future needs of aircraft engine development. The company requires a single and secured storage of product data enabling an early access of production engineers to the design choices or allowing to share product data with its main contractors and suppliers.

As specified in [5], the main benefits of VPM-Chains implementation are:

- Reducing development lead time and non-quality according to ISO oriented and predefined processes. Nowadays in aeronautics, this kind of ISO oriented process is required by the Federal Aviation Authority and Joint Airworthiness Regulatory (JAR) before product certification [6].
- Sharing the right updated data between the relevant users group regarding the product development process, e.g. between mechanical designers and computational fluid dynamic (CFD) designers to minimize their iterations; between engineering designers and production engineers to validate as

soon as possible the decision-making involving manufacturing process specifications, etc.

- Enhancing reactivity and customer supports with a high level of traceability and an efficient product configuration management during the whole development cycle.

### 1.3. Research objectives

This paper aims at highlighting the added value of a UML approach [7,8] for modelling and specifying the product structure [9,10], the workflow [11,12] and the interoperability with CAE and scientific software [13]. Considering its applications to enterprise modelling and business process reengineering [14], UML has been chosen for several reasons. First, it provides a fairly complete modelling notation for specifying the product breakdown structure (parts) with related product data (documents). Second, it provides an efficient language for modelling generic workflow with activities and transition criteria before their implementation in a workflow engine. Finally, UML enables to detail the overall class diagram of the VPM-Chains system and a framework for collaborative design with suppliers in an extended enterprise approach [15,16].

Basically, the tasks of each involved person have been identified in a use case diagram (Fig. 1). The various kinds of functional user needs regarding VPM-Chains system have been clarified. This work has been carried out based on several interviews of design team members: mechanical designers, CFD designers, production engineers, project leader, design manager, etc. All these interviews have been analysed and provide a fairly good overview of the user needs and design project running based on data and process modelling. The method used for this work was based on a classical BPR approach [17].

One of our challenges was to explain and to communicate about this complex project and also to ascertain the various users' level of understanding. A good way to explain the functionalities and the running of a PDM system like this has been found in the use of UML [14,18]. Its various views and diagrams enable the choice of a relevant way of explanation. These explanations are heavily dependent on what kind of information users want to share.

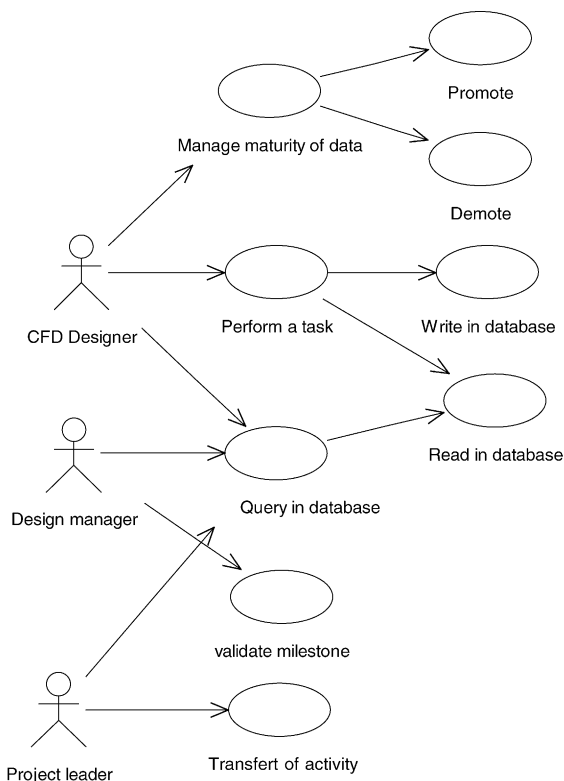


Fig. 1. Use case diagram of PDM functionalities.

## 2. Product structure modelling and management

The product structure manages and stores all the instantiated data. It uses metadata for the data classification and configurable links between the metadata [19]. The product breakdown structure is based on a parts list (including: product, sub-assemblies, and basic components). Files are related to these parts via metadata links according to their types: CAD or MS-Office files. Regarding the instantiation level in VPM-Chains system, the product breakdown structure (Fig. 2) allows the management of bills of material and product configurations [20].

### 2.1. Product breakdown structure

Two types of breakdown viewpoints have been defined:

- a design breakdown,
- an assembly breakdown.

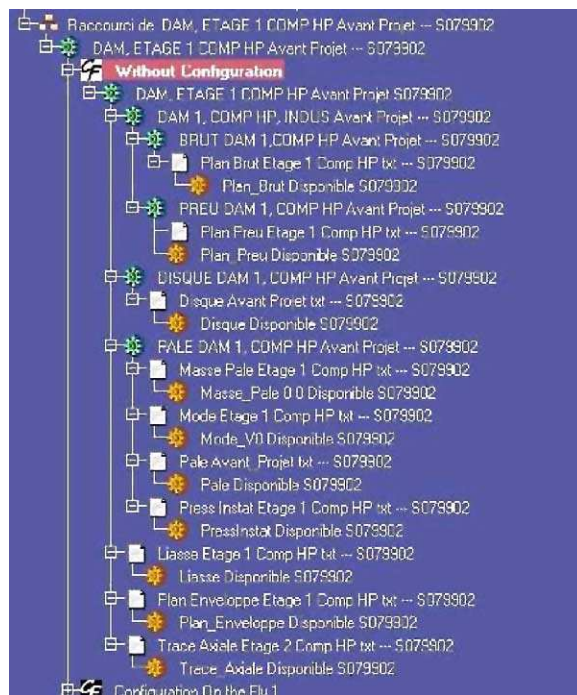


Fig. 2. Bill of material and product configuration manager of VPM-Chains system.

The main issue was that these two viewpoints of the product structure only partially provided a clear understanding of the full product breakdown structure. For example, the mechanical analysis and design of a high pressure compressor (HPC) is based on one s-specific sub-assembly. But a HPC is built on a spool that is a disk sub-assembly. Blades are also assembled on this spool. Regarding the assembly breakdown, the right disk has to be matched with the right blade. Then, no consistent links could be found between them in the product breakdown structure.

A joint product breakdown structure has been instantiated for an HPC made of three stages (Fig. 3). But new problems occurred: multi-instantiation of data shared between viewpoints (design, manufacturing, or assembly) could occur during implementation. As shown on Fig. 3, managing at the same time a viewpoint for design and one for manufacturing leads some parts to be instantiated twice, e.g. disks (1, 2 and 3) or blades (1, 2 and 3). It can be difficult to ensure consistency between data throughout the entire product lifecycle, from design to in-service phases of the turboprop. If a metadata is versioned in an

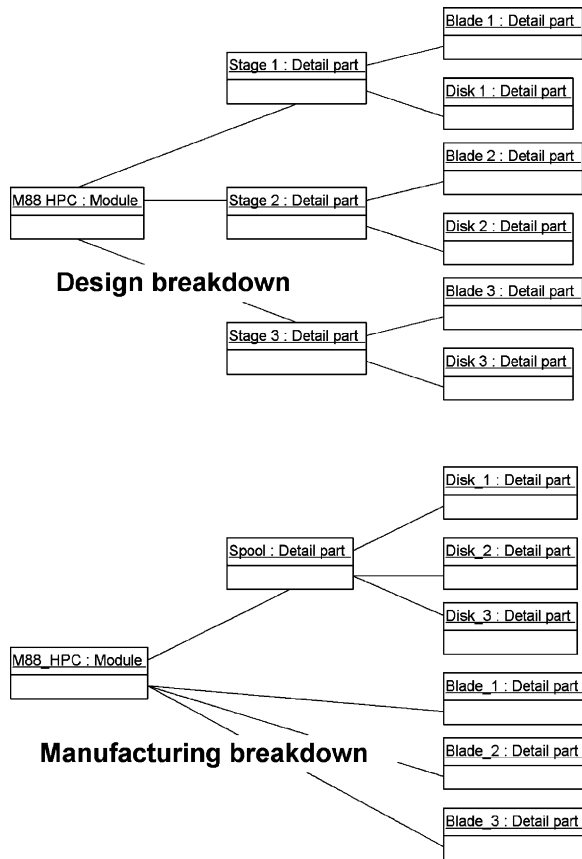


Fig. 3. Object diagram of the joint product breakdown.

instantiated part of the product breakdown, should this version update be notified to every user, even if they are working with the previous version? Do all the works in-progress have to be stopped or do the engineers are free to finish their structural analysis with the version previously in use?

Finally, a mixed modelling solution was chosen for the same instantiated HPC (Fig. 4): the top of the product breakdown (spool and stages) is common to every viewpoint (design, manufacturing, assembly), but each leaf (basic component: blade or disk) used in a lot of specific activities is dedicated to a single viewpoint. Then the whole stages of the HPC (generally, between four and twelve) with their corresponding disk and blade could be clearly modelled. This modelling solution avoids duplicating data: the shared data are integrated into the common part of the product breakdown. The instantiated

product structure does not even have to be browsed when searching specific document related to a specific part used in a given activity. The query will be based on given part linking the document and the activity. The main goal is to allow the workflow engine to automatically retrieve the right data without ambiguity.

For example, on Fig. 5 the class diagram of the product structure provides not only static information. This diagram helps to understand the evolution rules based on methods and the quite complex links existing between metadata. The class diagram is also highlighting how the product is structured in the system and how it is used. The product breakdown will be composed of Part class, its sub-classes and Document class. The performer will be able to use these classes for carrying out his activity. The relevant data will be characterised with feature class and its sub-class.

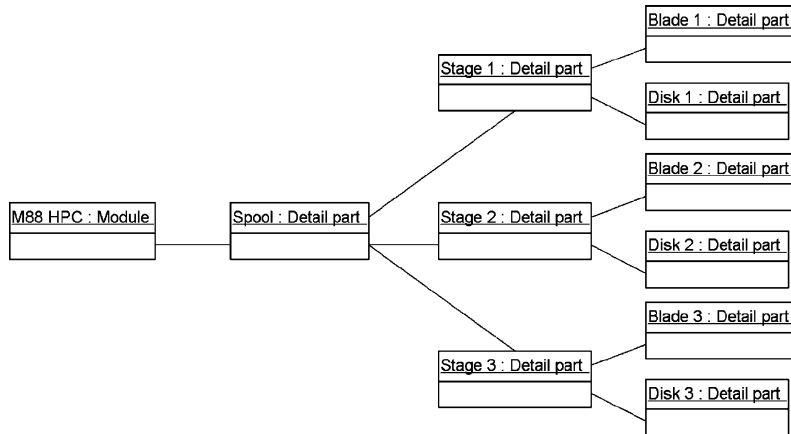


Fig. 4. Object diagram of the mixed product breakdown.

## 2.2. Change of data status during their lifecycle

The status represents the current issue of a state change of a data or a document during its lifecycle. It provides maturity criteria for data transfer and persistency.

There are four possible statuses for a document:

- “In progress” means that the data is currently being modified by an activity (or that nothing has been done yet).
- “Shared” means that the data is mature enough to be used by the process as input for other activity.
- “Released” means that this data is frozen and will not be further modified in this configuration.
- “Obsolete” means that the data cannot be used as input for an activity. A comment usually explains why data is not usable.

The owner of the data is in charge of it and promotes or demotes the maturity level.

The state diagram of a document (Fig. 6) clearly shows the various levels and states of a document and the rules of evolution. The obsolete status means that the data is no longer used. It corresponds to a “trashed” status.

## 3. Workflow modelling and management

As above mentioned, the VPM-Chains system includes workflow functionality [21]. The main goal is

to build and run the same generic process with controlled data for each new product development cycle. This process ensures customers of ISO oriented quality and information traceability. It also allows the project team member to find every relevant input data of a given activity. For example, he could find the meshing and the boundary conditions required for a blade structural analysis.

### 3.1. Capture of workflow data and characteristics

During the first phase of the VPM-Chains project, interviews were carried out with experts, and the process was broken down into elementary and basic activities [22,23] following two criteria:

- an activity is under the responsibility of one person,
- an activity lasts only a few days.

Then all the business processes have been redefined. The relevant data (input or output process data) were identified for the whole process [24,25]. The most difficult job was to highlight which constraints were limiting activities. Some activities are “push” activities. This means that the end of such an activity launches the start of the following one. But this process-oriented control does not correspond to a real process behavior. The need to pull data in order to launch activity is required, mainly due to the concurrent engineering approach [26,27]: rather than waiting the end of an activity before starting the

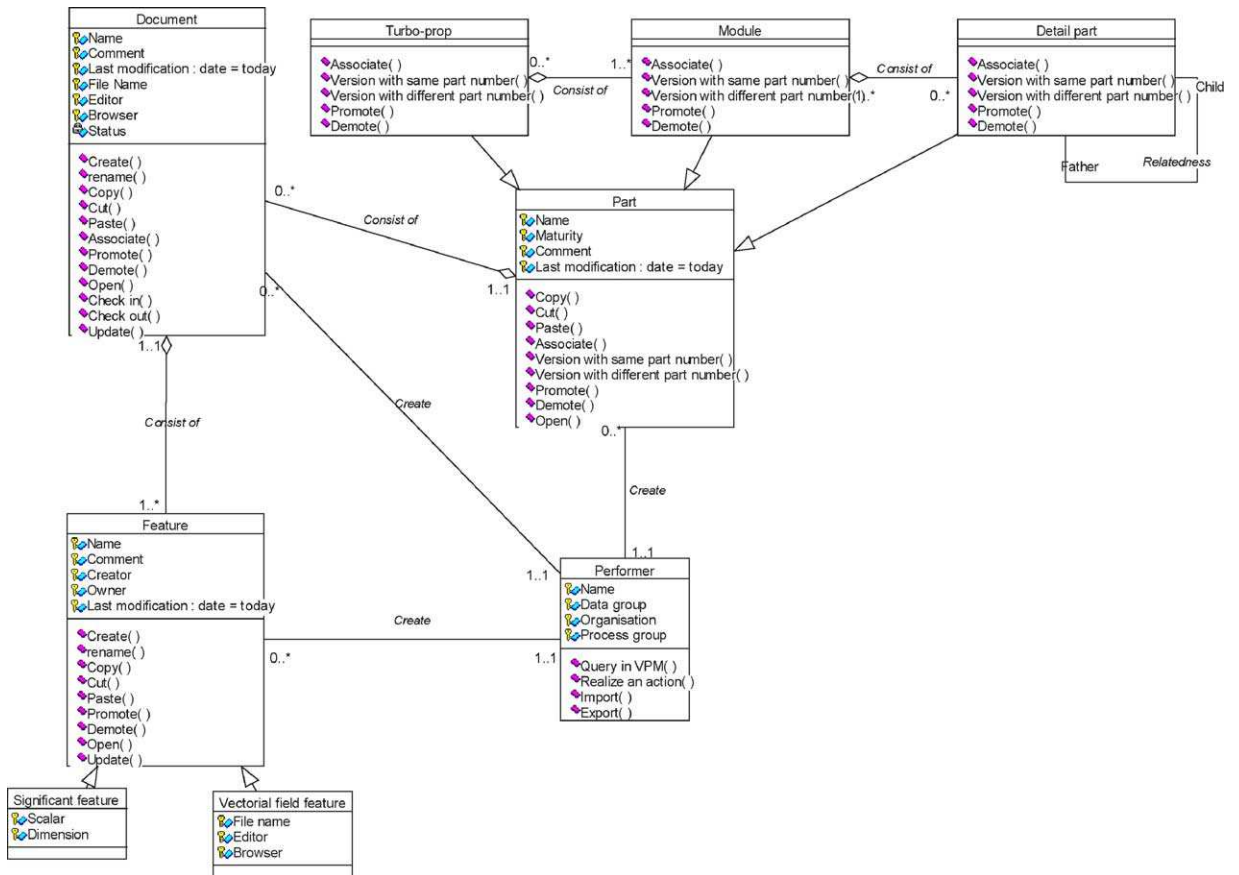


Fig. 5. Class diagram of the product structure.

following one, activities may start as soon as the input data are available. This data-oriented control is what the users needed, whereas today, only process-oriented control is provided in VPM-Chains.

### 3.2. Generic process specification

The build-time is the VPM-Chains functionality for modelling design processes called templates. This means that UML-defined processes should be as generic as possible in order to ensure a huge level of reuse [28]. It includes generic sub-processes as specified in the class diagram of the process structure (Fig. 7). The process and the product structures contain the generic names of the relevant data (in the product structure, data is also identified by the name of the father's part). For example, a high pressure compressor consists of several stages of disks and

blades. The design process of a turbo-prop stage is the same whatever the position of this stage in the compressor. Thus the same process can be instantiated for every stage of a turbo-prop (Fig. 8). If the process requires a data attribute called "mass of the blade" and the process associated to "Stage X" of the part is running, then the mass of the blade at stage X will be received.

The query on the right data is ensured by the link between a product and the considered process. This product is a sub-set of the product breakdown structure from which the query algorithm tries to retrieve the relevant data. The only way to associate this product with the process is to instantiate an empty activity in process. Thanks to this empty activity an engineering designer links the relevant data of product to be used in the process. Browsing the process the query algorithm is able to retrieve the associated product and relevant

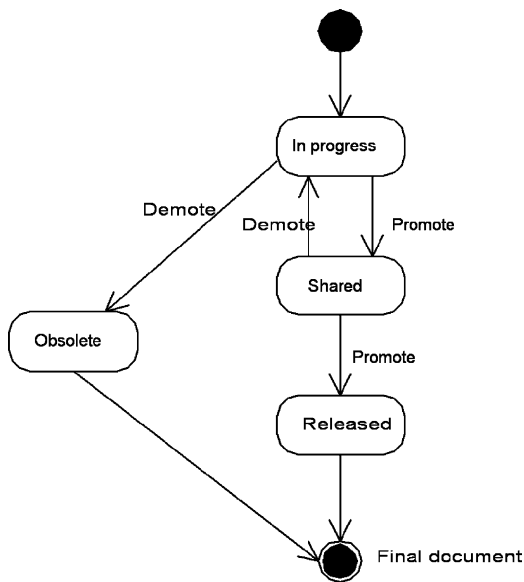


Fig. 6. State diagram of a document.

data. Then they could be taken into account for the next activity.

### 3.3. Classification of process and evolutions of the build time

Regarding design case, the process is strongly non-linear. It corresponds to a collaborative workflow according to the basic classification of Leyman and Roller [29]. This means that it is not a predictable and structured workflow: choosing one activity over another is often an expert decision-making.

Currently, only basic sequencing functions between activities are supported by the build time, such as iteration, “and” diverging and creation of sub-processes. The writing of new specifications is currently underway:

- The process has to control and monitor the product structure evolution. Users operations in the instantiated product structure are repetitive, complicated and boring, and therefore must be minimized.
- The data pull for controlling the start of activity should be enabled. This means to be able to launch an activity when all the required data are available. This allows a more complete concurrent engineer-

ing organisation thanks to the activity start as soon as the data is available and not when the previous activity is completed.

- Because the sequence logic during the design process could not be predefined and predictable, optional tasks are required. Based on these optional tasks the engineering designers will be able to choose to right activities for carrying out their job. Then thanks to this mechanism of optional task a little bit of flexibility may be introduced in the workflow.

A good overview of these requirements can be found on Fig. 9 in the activity diagram of a process sample. This process describes a mechanical design of a blade using structural analysis with an optional task of CFD analysis in order to validate structural analysis output.

## 4. Integration of the product with the process and the resources based on a framework for collaborative work

In order to understand how the product is integrated with the process and the resources, the architecture of the VPM-Chains system is briefly summarized bellow. The user has three available applications:

- Portal: the web client of the VPM server that supports the access to product data [30–32].
- BuildTime: the module for modelling generic processes.
- RunTime: the workflow engine for running the instantiated processes.

### 4.1. Principles of the briefcase

When running an instantiated workflow in Run-Time, the system creates, at the expected time, an action in VPM for each workflow activity (Fig. 10). This action retrieves all the relevant data in the vault. After promoting his action, the engineering designer can download and read a package called “the briefcase”. It contains the files corresponding to the relevant data and a procedure, which is an XML file that lists all the relevant data and their corresponding software release [33,34]. A briefcase includes:



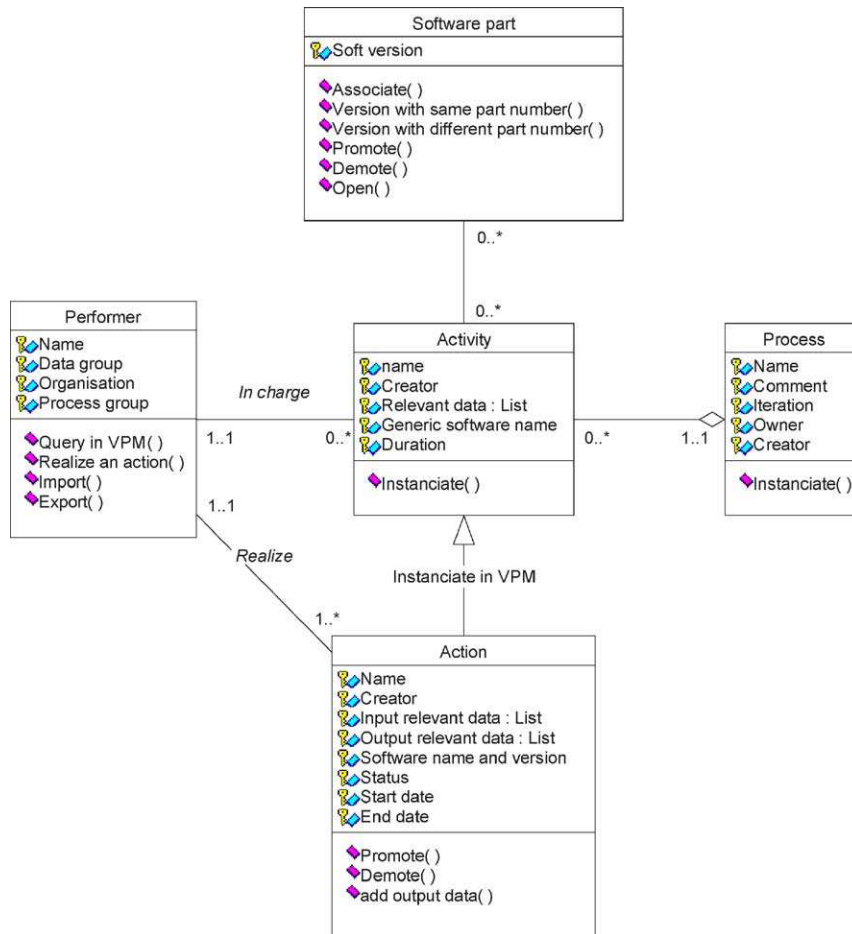


Fig. 7. Class diagram of a process structure.

- elements of the process because actions are created according to a workflow,
- documents and files available in the product structure,
- metadata specifying the current implemented release of the CAD and CAE software to be used for issuing output data.

An action and its local instantiated briefcase provide the integration of the various dimensions of VPM-Chains system: product, process, and resource [5]. These dimensions are summarized in the class diagram of the system (Fig. 11). Product dimension is mainly composed of Part, Document, Turbo-prop, Module, and Detail part classes. Process dimension links Activity, Process, Action, and Performer classes.

Resource dimension is mainly composed of Part and Software part classes.

In Portal, a query can be done on the particular briefcase of the user computer. Portal reads the procedure in XML file and shows which documents should be available in the activity input and output. It also provides access to the attribute of the relevant data.

After carrying out his task, the engineering designer refreshes the relevant data attribute in his briefcase. He saves the modifications and checks the briefcase into the vault. Finally, he promotes his action and validates it.

#### 4.2. A framework for collaborative work

This briefcase provides a framework enabling an easy transfer to a supplier. The current problem is how



Fig. 8. Instantiated workflow in VPM-Chains RunTime.

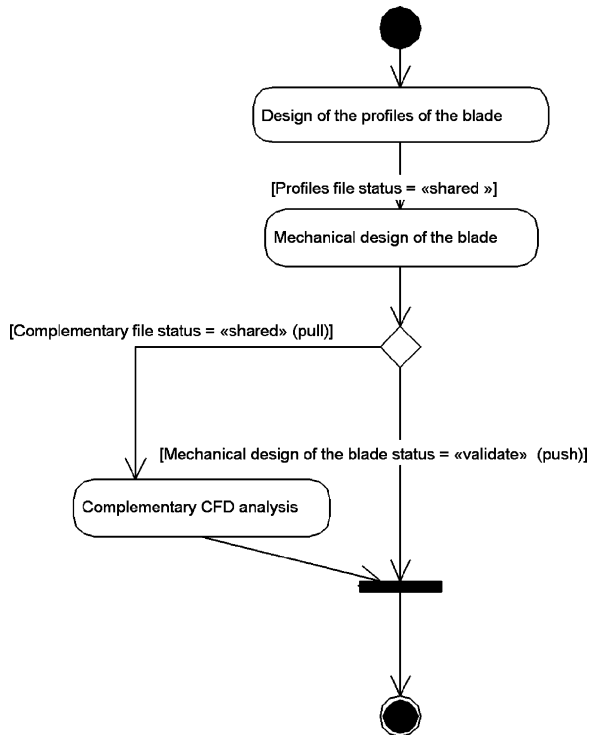


Fig. 9. Activity diagram of a blade mechanical design.

to maintain association with the context when the briefcase is out of the system [35]. Once the briefcase is downloaded, no more links to the VPM server are left. This problem is critical for CAD files. The mechanical engineer designs a part taking into account the constraints of the sub-assembly integrating the part. This means he needs to download a lot of data for his design task. It also means that a tool is required for synchronizing the changed and affected data. If a CAD model is modified, then the model has to be refreshed whether it is in the briefcase or not. All these questions remain unresolved for the time being.

#### 4.3. Software interoperability

An XML based solution has been chosen for the interoperability of software with VPM when data are checked-out of the vault and downloaded on local computer. Linked software can read the XML procedure file and check the release of the referenced software and its own release. If the software identifies a difference, an alert message is sent to the user, but the work is not stopped.

Regarding specific software, the information are automatically retrieved by the procedure. For example, Patran, a finite element pre-processor, can find the

**Action DEM21\_ITE\_AEROMECA\_AEROPALE\_3\_MACT486\_1**

Rafraichir      Création      Porte Documents

Action: DEM21\_ITE\_AEROMECA\_AEROPALE\_3\_MACT486\_1      Statut Valide

**Description** (255 caracteres maxi.)  
AEROPALE\_3

Priorité *Low*      Mémoriser      Historique  
Type Bon de Travail      Produire      Actions filles  
Propriétaire S071553      Méthodologie      Action mère

**Dates**

Date de début      Date de fin  
Fevrier - 19 - 2003      Fevrier - 19 - 2003  
13:49:51      14:27:33

**Documents**

Nom	Description	Entrée/Sortie	Type
<input type="checkbox"/> Pale Ite_1 --2 Disponible A transferer		0	Document
<input type="checkbox"/> Pale Disponible A transferer	Pale	0	Feature
<input type="checkbox"/> Pale Disponible A transferer		0	Feature
<input type="checkbox"/> PALE DAM 1 COMP HP Ite_2 Track AeroMeca Consigne2 MACT405 --0 En cours		0	Part
<input type="checkbox"/> PALE DAM 1 COMP HP Ite_1 Track AeroMeca Consigne MACT421 --A En cours		0	Part
<input type="checkbox"/> Pale Ite_1 --1 Disponible A transferer		0	Document
<input type="checkbox"/> Veine Disponible A transferer	Veine	1	Feature
<input type="checkbox"/> Application GOUX v7.1 Operationnel	GOUX	1	Part
<input type="checkbox"/> COMP HP --A ActuaMerid consigne MACT450 -- En cours		1	Part
<input type="checkbox"/> Veine Avant Projet --1 Disponible A transferer		1	Document

Fig. 10. Action created by VPM server for each workflow activity.

value of the temperature field to be used and loads it in the current analysis session. It can also read other information such as the name of the project, etc.

## 5. Business case study

The aim of this section is to illustrate the use of the UML approach in an industrial project of PDM system deployment. This project is carried out with the help of an external business partner for specifying, customising, and implementing the requirements of Snecma Moteurs in VPM-Chains system. The several steps of such a project will be detailed and it will be underlined how UML can improve this work [18].

Considering the viewpoint of the external business partner a PDM system deployment is always a new project. In such a project the past experiences and feedback of already achieved projects are difficult to take in account because PDM hugely affects the business processes and also has to integrate a lot of internal best practises. Because such a system deployment strongly influences the company competitiveness, it is quite difficult to get some benchmarks for implementing PDM systems.

### 5.1. Modelling processes and product data

First of all, PDM projects start with the clarification of the process and the data to be used in the system.



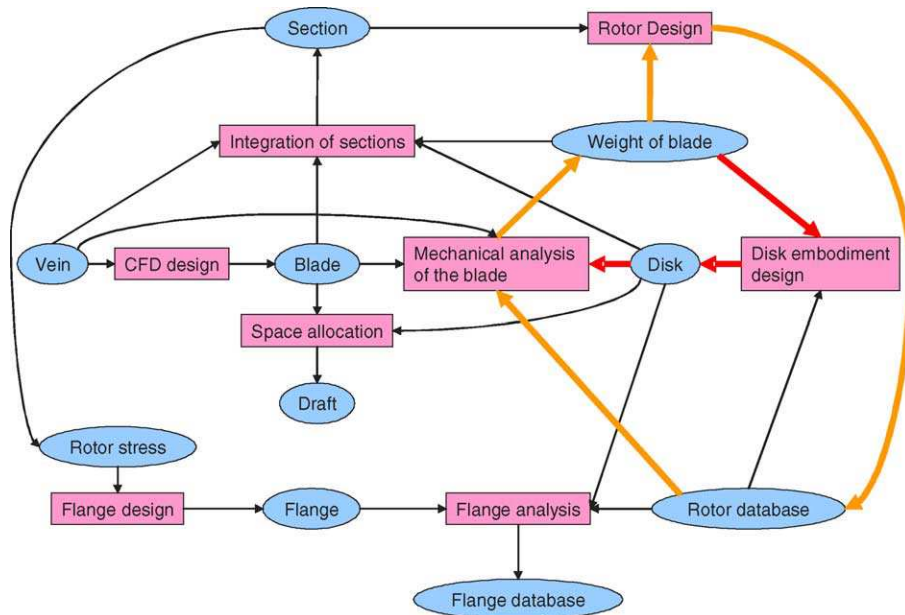


Fig. 13. Data flow as modelled at the project beginning.

Fig. 14 represents the activity diagram detailing the issue in Fig. 12. The validation of the diagrams with all people involved in the activity is strongly suggested and also the clear specification of the transition criteria

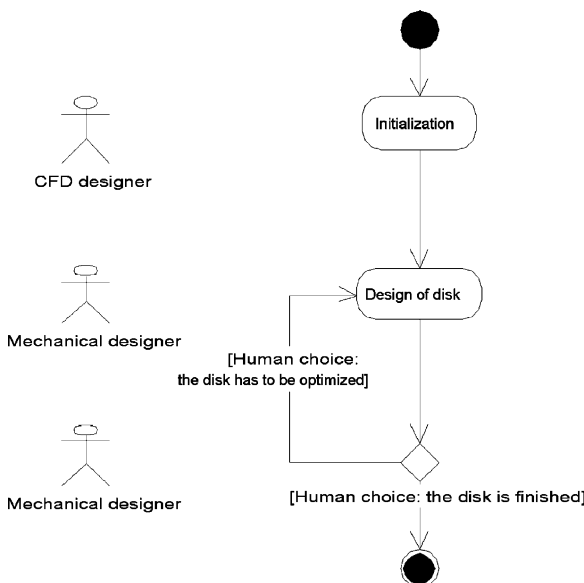


Fig. 14. UML based process modelling.

in order to enable the translation of the process model into the workflow. Fig. 15 shows the life cycle of a blade profile through a state diagram. A blade profile is created by a CFD designer and the vibratory and structural analyses are carried out by a mechanical designer. In an optional task, the CFD designer can verify whether the issued mechanical profile is still compliant with his specifications. Then he freezes (validates and promotes) the profile. Concerning the process modelling, dynamic and time criteria are supported. A clear specification of the transition criteria of activity is enabled. These criteria are difficult to be clarified but are essential for a good implementation of workflow. Concerning the modelling of data flows, two diagrams have been used: state diagram, describing the data life cycle, and class diagram, specifying all the attributes needed for an efficient deployment.

UML is considered as a structured notation system understandable without too many explanations. For many people, it could be used as a common language providing a shared point of view between users and PDM implementers. Even if there is not a single diagram of the whole system, there is always a set of diagrams which underlines a specific

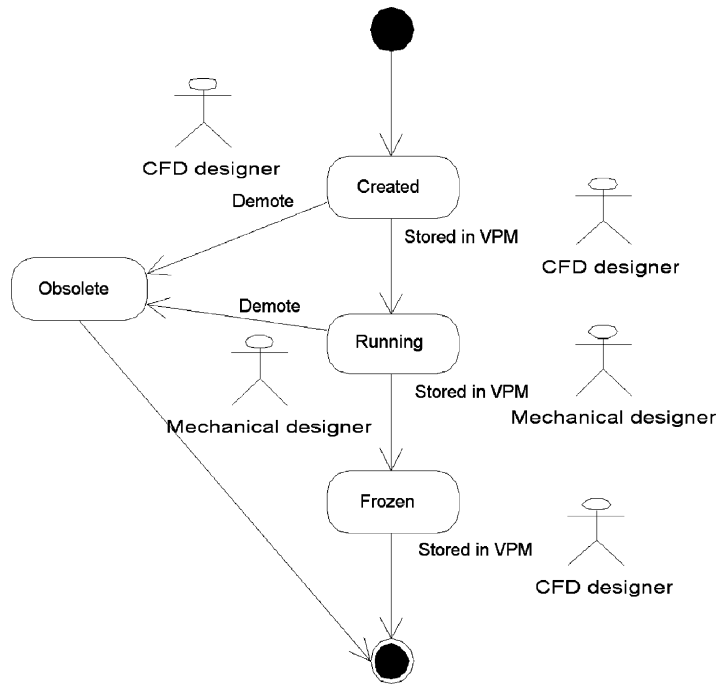


Fig. 15. UML based modelling of data lifecycle.

idea. The introduction of UML in the project has also improved the communication in the team. UML also natively supports main generic and instance concepts based PDM and workflow systems.

After modelling the process and the data, UML diagrams provide relevant tools for describing the link between the object of the system and the data of the company. This step is very useful because, if every links are clearly identified and specified, the implementation step will be easier and faster. It avoids to come back on process modelling and to ask clarifications to the users who are usually very busy with their daily tasks.

### 5.2. Clarifying needs and detailing specifications

The second field of a PDM-project where UML can prove to be useful is the clarification of needs. At this step of the project, the goal is to identify what will be the interactions between the users and the system. In other words, it means that the system implementers have to clearly identify the needs of the users and to share with them a common view (Fig. 1).

Use case diagrams can also be used to detail specifications. Splitting up the clarification of user needs and the system specifications is interesting because the second one has to fulfil the first one. After clarifying the needs, each one can be detailed into one or

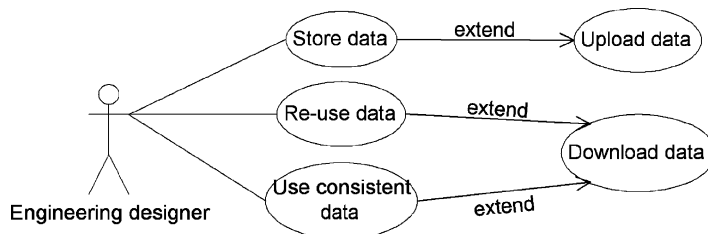


Fig. 16. Use case diagram of specifications.

several specifications. UML also provides interesting tools for this step. Use case diagrams can be completed to show the links with functionality specifications (data upload and download) as described on Fig. 16.

### 5.3. *Synthesis of business case study*

UML has provided during the whole project a clear and efficient solution to communicate between team members who had heterogeneous views and understanding of PDM systems. Technically, the nine UML diagrams always provide useful viewpoints considering the “to be” implemented system. The class and object diagrams efficiently support product structure and workflow modelling. The activity diagrams have seemed to be a good way to specify, discuss, and establish common views.

## 6. Conclusion

Currently in its last phase of implementation, the VPM-Chains system provides a single and secured storage for CAD or CAE data linked with product structure and workflow. Like many PDM systems, the VPM-Chains system and its strong link with CAE software also ensures a high level of traceability of design activities and technical choices. Based on the VPM powerful mechanism, the product structure management is efficient regarding other PDM system. Data redundancy is avoided thanks to instantiable metadata.

The analysis of the process has highlighted the need for more flexible tools for the build time (workflow modelling) and run time (workflow instantiation) modules. Although the process is repetitive and iterative, control criteria are difficult to formulate. One of the system’s main restrictions of use is its inability to support a real concurrent engineering organisation, specifically by controlling processes based on data flow. The last phase of implementation should normally solve these problems.

A PDM is a complex system which could be clearly defined and modelled using a tool such as the Unified Modelling Language. Various abstraction levels with the same kind of diagrams have been described: the needs of business case study, the global architecture, and the links between objects with class diagrams. The user interactions with the

system have been modelled with use case diagrams and generic processes with activity diagrams. The system environment regarding the software links and the co-development work with suppliers have also been taken into account based on UML diagrams. To summarize, UML provides a relevant and efficient modelling notation that might limit misunderstanding problems for the various dimensions for specifying and implementing the product structure and workflow in a PDM system. Enterprise Modelling methodologies (e.g. ARIS, CIMOSA or GRAI) are useful for the global understanding and re-engineering of company and its business processes but regarding the specification and implementation levels of dedicated software such as ERP, PDM, or SCM, UML is considered as an unquestionable supplementary modelling language.

## Acknowledgement

The results presented in this paper have been developed within the VPM-Chains Project of Snecma Moteurs. The authors are most grateful to Snecma Moteurs for its support and technical contribution.

## References

- [1] T. Pardessus, The multi-site extended enterprise concept in the aeronautical industry, *Air Space Europe* 3 (3/4) (2001) 46–48.
- [2] B. Carratt, B. Eynard, J.F. Calm, Concurrent aviation certification, in: *Proceedings of European on Safety and Reliability Conference, ESREL2001*, September 16–20, Torino, Italy, 2001.
- [3] CIMdata, 2001, Collaborative Product Definition management (cPDM): An Overview of a Collaborative Approach to Managing the Product Definition Lifecycle, CIMdata Inc, Ann Arbor, <http://www.cimdata.com>.
- [4] D.T. Liu, X.W. Xu, A review of web-based product data management systems, *Comput. Ind.* 44 (3) (2001) 251–262.
- [5] F. Lafont, VPM-Chain design and industrialisation process software, in: *Proceedings of European Catia Forum*, October 17, Berlin, Germany, 2001.
- [6] B. Carratt, 2002. Ingénierie collaborative et gestion de données techniques en certification aéronautique, DRT Thesis of Troyes University of Technology.
- [7] Object Management Group, 1997. The Unified Modeling Language, Release 1.1, Reference Manual, <http://www.omg.org/>.
- [8] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modelling Language User Guide*, Addison-Wesley, Reading, 1999.
- [9] P.Y. Chao, T.T. Chen, Analysis of assembly through product configuration, *Comput. Ind.* 44 (2) (2001) 189–203.

- [10] Y. Oh, S.H. Han, H. Suh, Mapping product structures between CAD and PDM systems using UML, *Comput. Aid. Des.* 33 (7) (2001) 521–529.
- [11] B. Eynard, C. Merlo, B. Carratt, Aeronautics product development and certification workflow based on process modelling, in: *Proceedings of the Eighth International Conference on Concurrent Enterprising, ICE 2002*, June 17–19, Rome, Italy, 2002.
- [12] I. Choi, M. Song, C. Park, N. Park, An XML-based process definition language for integrated process management, *Comput. Ind.* 50 (1) (2003) 85–102.
- [13] S. Charles, B. Eynard, C. Paleczny, T. Gallet, Echange et gestion de données techniques en ingénierie mécanique chez un motoriste avion, in: *Proceedings of the 22nd International Conference in CAD/CAM, PLM and New Technologies of Design and Manufacturing (MICAD'03)*, April 1–3, Paris, France, 2003.
- [14] C. Marshall, *Enterprise Modeling with UML*, Addison-Wesley, Reading, 2000.
- [15] H.S. Jagdev, J. Browne, The extended enterprise – a context for manufacturing, *Prod. Planning Control* 9 (3) (1998) 216–229.
- [16] Y. Zhang, C. Zhang, H.P. Wang, An Internet based STEP data exchange framework for virtual enterprises, *Comput. Ind.* 41 (1) (2000) 51–63.
- [17] M. Hammer, J. Champy, *Re-engineering the corporation*, HaperCollins, New York, 1993.
- [18] C.H. Kim, R.H. Weston, A. Hodgson, K.H. Lee, The complementary use of IDEF and UML approaches, *Comput. Ind.* 50 (1) (2003) 35–56.
- [19] E. Eloranta, A.P. Hameri, L. Lahti, Improved project management through improved document management, *Comput. Ind.* 45 (3) (2001) 231–243.
- [20] K. Rouibah, K.R. Caskey, Change management in concurrent engineering from a parameter perspective, *Comput. Ind.* 50 (1) (2003) 15–34.
- [21] L. Fischer, *The Workflow Handbook 2001*, Future Strategies Inc., Lighthouse Point, 2000.
- [22] A.W. Scheer, R. Robowsky, S. Kalbunde, A. Taut, Flexible industrial applications through model based workflows, in: *Proceedings of the International conference on Enterprise Integration and Modelling Technology*, Torino, Italy, 1997.
- [23] K. Saikali, 2001, Flexibilité des workflows par l'approche objet: 2Flow: un framework pour workflows flexibles, Ph.D. Thesis of Ecole Centrale de Lyon.
- [24] L. Gzara, 2000, Les patterns pour l'ingénierie des systèmes d'information produit, Ph.D. Thesis, INP Grenoble.
- [25] L. Gzara, D. Rieu, M. Tollenaere, Product information systems engineering: an approach for building product models by Reuse of patterns, *Robot. Comput. Integr. Manufact.* 19 (3) (2003) 239–261.
- [26] G. Sohlenius, Concurrent engineering, *Ann. CIRP* 41 (2) (1992) 645–655.
- [27] B. Prasad, *Concurrent engineering fundamentals*, vol. 1, Prentice-Hall, Englewood Cliffs, 1996.
- [28] D.R. Liu, M. Shen, Workflow modeling for virtual processes: an order-preserving process view-approach, *Info. Syst.* 28 (6) (2003) 27.
- [29] F. Leyman, D. Roller, *Production Workflow*, Prentice-Hall, Englewood Cliffs, 2000.
- [30] M. Rezayat, The Enterprise-Web portal for the lifecycle support, *Comput. Aided Design* 32 (2) (2000) 85–96.
- [31] G.Q. Huang, Web-based support for collaborative product design review, *Comput. Ind.* 48 (1) (2002) 71–88.
- [32] X.W. Xu, T. Liu, A web-enabled PDM system in a collaborative design environment, *Robot. Comput. Integr. Manufact.* 19 (4) (2003) 315–328.
- [33] W.C. Burkett, Product data markup language: a new paradigm for product data exchange and integration, *Comput. Aided Design* 33 (7) (2001) 489–500.
- [34] Y.H. Yao, A.J.C. Trappey, P.S. Ho, XML-based ISO9000 electronic document management system, *Robot. Comput. Integr. Manufact.* 19 (4) (2003) 355–370.
- [35] H.F. Wang, Y.L. Zhang, CAD/CAM integrated system in collaborative development environment, *Robot. Comput. Integr. Manufact.* 18 (2) (2002) 135–145.



**Benoit Eynard** is currently Assistant Professor in Mechanical Engineering in the Department of Mechanical Systems Engineering at the Troyes University of Technology. He is also member of the Laboratory of Mechanical Systems and Concurrent Engineering (LASMIS). He received his Ph.D. degree from the University Bordeaux 1 in 1999. The Ph.D. subject dealt with Engineering Design and Computer Integrated Manufacturing. Currently, his research interests include Concurrent Engineering, Product Data Exchange, Product Lifecycle Management, Engineering Knowledge Modelling and Reuse.



**Thomas Gallet** received the Engineer degree in Mechanical Systems from Troyes University of Technology in 2002. He has defend his MSc in Systems Engineering at UTT in 2003 after twelve months of research engineer position at Snecma Moteurs working on PDM implementation in mechanical and aeronautics engineering. Currently, he is PLM engineer at the Renault-Nissan Information System Department.



**Pierre Nowak** received the Engineer degree in Mechanical Systems from the Compiègne University of Technology in 2001. He is a Ph.D. student in the Laboratory of Mechanical Systems and Concurrent Engineering of the Troyes University of Technology. His research interests include Traceability and Reuse of Engineering Knowledge, Design Alternatives Modelling, and Meta-modelling for Combining Models.





**Lionel Roucoules** currently is Assistant Professor in Mechanical Engineering in the Department of Mechanical Systems Engineering at the Troyes University of Technology. He is also member of the Laboratory of Mechanical Systems and

Concurrent Engineering (LASMIS). He received his Ph.D. from the National Polytechnic Institute of Grenoble in 1999. He did one year of Post-Doc research at the University of Twente. His investigations are related to Product Modelling, Design For Manufacturing, Concurrent and Collaborative Engineering.