PDM system implementation based on UML

Benoît Eynard, Thomas Gallet, Lionel Roucoules, Guillaume Ducellier

Abstract

The paper deals with a Unified Modeling Language (UML)-based approach for implementing Product Data Management (PDM) systems. Such kind of system enables the management of the entire product lifecycle and related information about its design, its manufacturing and its in-service.

The interest of using an UML-based approach for modeling, specifying and implementing PDM systems is detailed. Then an implementation method is proposed in order to clarify the instantiation of UML diagrams into database entities. Regarding a business case study based on the needs of a turboprop aircraft development project, the preliminary implementation of PDM system is presented. The used UML diagrams for the modeling and integration of product, process, and resource data is detailed in order to argue the interest of an object-oriented approach in such kind of PDM implementation.

© 2005 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: PDM, Unified Modeling Language, Product structure, Workflow, Aeronautics development process

1. Introduction

“Time to market” is becoming one of the most important business and strategic key elements for product development and manufacturing. Customers also want a more customized product. In aeronautics, companies have to reduce leadtimes and to develop a better-configured product fulfilling customers requirements including an higher level of quality and safety [24,8]. These topics represent major challenges in turboprop manufacturing as it is a niche market; in 2001, the sector total sales were equivalent to the sales of General Motors alone.

Section 2 presents the key features of Product Data Management (PDM) systems and the basic needs for their implementation. Section 3 proposes the overview of a Unified Modeling Language (UML)-based method for PDM implementation. Section 4 describes the used approach for product structure modeling. Section 5 details the modeling of design process and the specification of the corresponding workflow. Section 6 deals with a business case study of PDM implementation based on UML specifications. Finally, Section 7 concludes this research work and summaries some future developments on UML-based method for PDM implementation.

© 2005 IMACS. Published by Elsevier B.V. All rights reserved.

2. PDM features and implementation requirements

PDM system aims at managing and storing the product data and also information related its entire lifecycle (manufacturing, assembly, maintenance, etc.) [6,17]. In this paper, two main functionalities of such a system will be considered. The first one is the product structure management and the second one is the workflow engine [15,14].

Product structure manager organizes and stores the whole product data including the parts list and the various product configurations. It controls the data versioning and the link between the parts of the bill of material and associated documents (CAD or text files). It also manages of data lifecycle status: available, frozen or obsolete.

According to the product structure definition, workflow engine distributes the right data to the key user at the right time. The workflow functionality allows the implementation of ISO oriented processes in product development in order to be sure that key tasks (such as decision making, product review, etc.) are effectively carried out by the right team member.

Many others functionalities are also currently available in PDM system but they will not be discussed in this paper. Functionalities such as product configuration management, digital mock-up visualization, and collaborative tasks management can be mentioned.

2.1. Industrial context of the VPM-Chains project

Snecma Moteurs, one of the world leaders in turboprop manufacturing, has decided, at the end of the 90s, to specify, implement and deploy a new PDM system called VPM-Chains. This system should be mainly based on ENOVIA VPM from Dassault Systèmes. The main idea was to strongly customize VPM functionalities with the integration of virtual product manager and workflow engine from ENOVIA LCA. Snecma Moteurs specified the need of secured product data management allowing an early access of production engineers to the design decision-making and simplifying the sharing of digital mock-up between original equipment manufacturer and its suppliers.

VPM-Chains implementation should provide improvements on the bellow topics [15]:

- To reduce product development lead time and decrease non-quality according to ISO oriented and predefined processes. For example, the Federal Aviation Authority and Joint Airworthiness Regulatory (JAR) require such kind of ISO oriented process before product certification.
- To share the updated data between the right key persons of project team regarding the product development process, e.g. between mechanical designers and computational fluid dynamic (CFD) designers in order to reduce their iterations on the blade profile specification, between engineering designers and production engineers to agree as soon as possible the product definition and manufacturing process specifications, etc.
- To enhance reactivity and customer supports with a high level of traceability and an efficient product structure management during the whole development cycle.

2.2. Modeling requirements for PDM implementation

The added value of UML [22,1] for modeling and specifying the product structure [3,23] and the workflow [4,18] is considered in the paper. Its applications [20] in Enterprise Modeling and Business Process Reengineering [31] has led to the choice of UML for several reasons. It efficiently supports the modeling and specification of the product breakdown structure (parts) and related product data (documents). It ensures a relevant modeling of generic workflow with activities and transition criteria. Then their implementation in workflow engine is fairly simplified. Last, the overall class diagram of the VPM-Chains system for specifying all kind of data to be managed is efficiently supported by UML.

2.3. Brief survey of PDM implementation methods

Often, PDM implementation methods are based on information system modeling approaches [10,11,30]. Considering this subject, various methods have been proposed for capturing the requirements [19,28], modeling and structuring the data [4,9], specifying the implementation [5,13]. Several of the above mentioned research works are based on object-oriented methods. The MOKA framework can also be mentioned as a good example of UML-based approach.
3. UML-based method for PDM implementation

The implementation process of new chain includes five major phases (Fig. 1):

- Clarification of the design chain:
  In this phase, the subject and scope of the design chain are defined. The team members to be involved are identified and the implementation project planned. The requirements list of design chain is specified including the use of UML use-case diagram.

- Capture and analysis of process and data:
  The capture of design and production engineering processes and the modeling of input and output data of each task are carried out. This phase involves key experts of technical processes and business units working in the chain and designers carrying each basic task. The issue of this phase is deliverable summarizing the required data in order to implement the chain with UML use-case, class and activity diagrams.

- PDM implementation:
  This phase consists in the specification of tree structure including product data and documents on the one hand and in the definition of workflow including all the tasks and data processing on the other hand. This workflow tends to be created as a generic model with the buildtime. Therefore, this phase requires a good understanding the expert tasks of the design chain and a clever use of system. At the end of this phase, the system is ready to be tested at a large scale.

- CAx integration:
  The aim of this phase is to link the numerous CAD and CAE systems used in design chain in order to ensure a complete traceability and to improve the quality of the design. Therefore, design and simulation tasks must be clarified. Each task is identified and named according to generic ID. CAx integration consists then in specifying the
name and type of task where the software should be used including the data list required to carry out the tasks. After a set of case studies and final validation, all software to be used are clarified including the data to be issued.

- Validation case studies:

The last phase consists in a range of case studies including a scenario which allows the final test and the validation of the design chain implementation. If the scenario issues are concluding then the chain implementation is deployed.

Indeed, both PDM implementation and CAx integration phases need the complete achievement of the capture and analysis of process and data phase (Fig. 1). These two major phases can be parallelized in order to reduce the global implementation time.

4. Product structure modeling

All the information belonging to the product lifecycle phases (design, manufacturing, assembly, etc.) are managed and stored within the product structure. The definition of this structure is based on metadata for the data classification and configurable links for the specialization of relationship between the metadata [7]. The product structure is considered as a hierarchical parts list including: product, sub-assemblies, and basic components. Regarding their types (CAx or MS-Office files), documents are attached to these parts via metadata links. In VPM-Chains system, the product structure allows the management of bills of material and product configurations [2].

4.1. Specification of product structure

According to the turboprop development process, two kinds of breakdown viewpoints have been defined:

- product structure based on design viewpoint,
- product structure based on assembly viewpoint.

These two viewpoints of the breakdown only partially provided a clear understanding of the full product structure. For example, the mechanical analysis of a high pressure compressor (HPC, Fig. 2) is made on only one specific sub-assembly. But a HPC is based on a spool: disk sub-assembly; and blades are also assembled on this spool. Regarding the assembly product structure, the right disk has to be matched with the right blade. Then, no consistent links could be found between them in the product structure. Previously, a joint product structure has been specified for an HPC including three stages. But new problems occurred: multi-instantiation of data shared between viewpoints (design, manufacturing, or assembly) could occur during the product structure implementation.

![Fig. 2. High pressure compressor rotor.](image-url)
A mixed modeling solution was chosen for describing the HPC (Fig. 3). The top of the product breakdown (spool and stages) is common to every viewpoint (design, manufacturing, assembly). Concerning each leaf (basic component: blade or disk) used in a lot of specific activities is dedicated to a single viewpoint.

This modeling solution avoids duplicating data: the shared data are integrated into the common part of the product structure. The instantiated breakdown does not even have to be browsed when searching specific document related to a specific part used in a given activity. The query will use a given part linking to the document and the activity. The main goal is to allow the workflow engine to automatically retrieve the right data without ambiguity.

The class diagram of the product structure (Fig. 4) provides not only static information but also helps to understand the methods and relationships existing between metadata. The class diagram also specifies how the product definition is clarified in the system and how the user may handle it. The product structure will be composed of Part class, its sub-classes and Document class. The performer will be able to use these classes during his activity. The relevant data will be characterized with Feature class and its sub-class.

4.2. Product data and documents lifecycle

In product data management a key attribute is the information status. It represents the current issue of a state change of a data or a document during its lifecycle. It provides a very important maturity criterion for data transfer and persistency. Then user may have access or not to the data and documents depending on their status.

For a document, four possible statuses exits (Fig. 5):

- “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 in the system.

5. Design process modeling

The VPM-Chains system includes workflow functionality [16]. The main goal is to build (model) and run the same generic process with controlled data for each new turboprop development. The workflow ensures airworthiness authority and customers of ISO oriented quality and information traceability. It also allows a project team member to find every relevant input data of a given activity.
Fig. 4. Class diagram of the product structure.

Fig. 5. State diagram of a document.
5.1. Capture of activities and processed data

In order to clarify the workflow, interviews with experts are carried out, and the process split up into basic activities [27, 26] according to two criteria: an activity is under the responsibility of one person and lasts only a few days. Then all the business processes have to be redefined with all the relevant data (input or output data of each activity) [9]. A difficult point is to identify which activities should be considered as "push" and which ones as "pull". At the end of "push" activities the following one is triggered. Mainly due to concurrent engineering [29, 25], the need to pull data in order to launch activity is required: activities may start as soon as the required input data are available rather than waiting the end of an activity for triggering the following one. This "pull" approach is currently what the users really needed, but nowadays only the "push" one is included in VPM-Chains system.

5.2. Specification of generic process

5.2.1. Workflow based on generic process

The build-time VPM-Chains functionality is used for modeling “templates” describing generic design processes. Then UML-based processes should be as generic as possible in order to ensure a huge level of reuse [18]. It includes generic sub-processes as specified in the class diagram of the design process (Fig. 6). The process and the product structure diagrams contain the generic names of the relevant data (in the product structure, data is also identified by the name of the father’s part). The link between a product and the considered process enables query on the needed data.

5.2.2. Specification of design workflow

The design process is strongly non-linear because of design change and iterations between expertises (mechanical and CFD analyses). According to the basic classification of [16], it is considered as a collaborative workflow. It is not possible to instantiate a predictable and structured workflow: the choice of one activity or another is often based on an expert decision-making. Currently, the VPM Chains build time provides only basic sequencing functions between activities, such as iteration, “and” diverging and creation of sub-processes.

New specifications are currently under writing:

- The process has to control and monitor the product structure evolution.
- The data pull for controlling the start of activity should be enabled.
- As the design process could not be predefined and predictable, optional tasks are required in order to allow alternative choices of following activities introducing a little bit of workflow flexibility.

5.2.3. Clarification of transition criteria

In order to take in account of transition criteria in the specification of design workflow, the activity diagram has been semantically enriched with a notation derived from mathematics notations and Object Constraint Language [32]. In the activity diagram, when the criterion is true then it is a trigger for the following activity. If no criteria are specified on the link between two activities then the validation of the previous activity launches automatically the following one in the same approach than Petri nets. The only handled objects are the features of the product structure. In opposite to OCL, the access to object is specified via the notation of the object and its attribute: object attribute = value of attribute.

As several objects may have the same name, each called object (feature) should be identified regarding its belonging product structure and associated workflow. One or several features may be concerned by one criterion. In order to clarify the considered finite number of objects the classical mathematics notations (∀, ∃, ∃!, etc.) are used. The decision making is called as a method with brackets in which appears a list of features based decision-making: decisionmaking (feature 1, ..., feature n). The aim of this notation is not to develop a formal language but simply add to the activity diagram all required data for workflow implementing. Fig. 7 shows and example of activity diagram enriched the proposed notations.

6. Implementation based on UML specifications: business case study

Based on a design chain case study, the aim of this section is to illustrate the use of the proposed method in the industrial implementation project of VPM chains system. Consultancy business partner helps Snecma Moteurs
for specifying, customising, and implementing the PDM requirements. The implementation of product structure and workflow will be detailed and it will underline how UML specifications can be used as guideline for this work [12]. The introduction of UML as a formal modeling notation enabled a global understanding and a common viewpoint of the project progress shared by all the team members. UML seems to be a good first step for sharing some structured ideas about clarified business process and data.

During the modeling of the process and the data, UML provides relevant diagrams for describing the link between the object to be implemented in the system and the data handled in 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 modeling and to ask clarifications to the users who are usually very busy with their daily tasks.

6.1. Product structure implementation

Based on the product structure requirements clarified on Fig. 4, the instantiation of object composing the product structure is simplified and the final implementation in VPM-Chains system is therefore easier (Fig. 3).
The first step in the product structure implementation is the setup of a consistent static view of the product breakdown. During users' interviews, the need of two complementary views has been clarified (Fig. 3): design view and manufacturing view.

High pressure compressor case study clearly depicts the need of various views regarding involved expertises. Rotor design is based on the design of each HPC stage in mechanical analysis and CFD analysis. At the same time, manufacturing at these stages does not have sense. The only useful parts for the manufacturing are the: blades, spool, DAM (blades and disk).

The following implementation step consists in implementing the change rules of the product data lifecycle (dynamic views) based on version for parts and revision for documents (Fig. 8). In order to manage changes of version and revisions, the product structure has to be the unique and referenced list of product data. It will also allow the query on wished
data according wished maturity. Change rules have been consistent regarding properties of the static view of product structure.

Fig. 6 shows the current way of implementing the product structure regarding the needs of static and dynamic views and Fig. 9 presents its implementation with ENOVIA VPM.

6.2. Workflow implementation

The validation of the UML diagrams with all team members involved in the design processes is strongly suggested. The clear specification of the transition criteria in order to enable the translation of the process model into the workflow is also recommended. The above proposed formulation for transition criteria is very useful for an efficient implementation of workflow.

Fig. 10 shows the lifecycle 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 by a mechanical designer. If the issued mechanical profile is not compliance with CFD specifications, the CFD designer can carry out new design iteration in order to improve the issues. Then he/she freezes (validates and promotes) the profile. Concerning the process modeling, 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 modeling of data flows, two diagrams have been used: state diagram, describing the data lifecycle, and class diagram, specifying all the attributes needed for an efficient deployment.
The workflow implementation requires the association with a product structure and team members identified as performer. Features of implemented product structure will be allocated to the performer and linked to an action with father document and also its father part. Fig. 7 summaries with process class diagram the link between the various objects.

Regarding an implemented workflow (Fig. 11), the relevant data of the “Mecapale” activity are: Blade CAD as input, Blade Meshing as output and CATIA as resource. The “Rotor HPC” part (Fig. 9) is associated to this workflow. When the “Mecapale” activity is triggered at the end of “Aeropale” activity, an action is automatically created in VPM. The system searches all the relevant data children of the “Rotor HPC” part and attaches them to the action. Then this action will reference:

- as input: “Blade CAD” feature, a document with the same name (only the latest: revision – 2) and “Blade RM1” part,
- as output: “Blade Meshing” feature, a document with the same name and “Blade RM1”,
- as resource: the Catia V5 part.

6.3. Synthesis of the 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. The activity diagrams have seemed to be a good way to specify, discuss, and establish common views of the workflow to be implemented.

The UML-based method allows to guide the project team members through the various phases of the PDM implementation. The clarification and capture phases ensure team members to clearly understand the “As Is” Product Information System but also the global and detailed turboprop design chains. Based on UML specifications the implementation phase is simplified as the generic attributes of each object are clarified in the class diagram. Considering the workflow, the data specified in diagrams are fairly satisfying. But the implementation allowed in the workflow engine is currently to poor in order to efficiently support the dynamic of design process and also its required flexibility while the turboprop development process are strictly specified due to the quality and safety requirements of aeronautics industry.
7. Conclusion

The VPM-Chains system provides a secured management for CAD or CAE data linked with product structure and workflow. Based on the ENOVIA VPM and LCA functionalities merging, the digital mock-up manager and workflow engine clearly meet the user requirements. The business process analysis has highlighted the need for more flexible tools concerning the build time (workflow modeling) and run time (workflow engine) modules.

UML-based method for PDM implementation has been proposed and validated within the business case study of design chain. The method efficiently guides project team and helps them in carrying out such a complex implementation process. 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 clarified with use case diagrams and generic processes with activity diagrams.

Based on the experience presented in the paper, UML can be considered as a relevant and efficient notation enabling the modeling, specification, and implementation of PDM system especially concerning the product structure and workflow. This modeling notation limits misunderstanding problems for the various dimensions of the implementation phase. Preliminarily to this step, enterprise modeling methodologies such as ARIS, CIMOSA, GRAI or PERA should be carried out for the global understanding of the company organization and management, and for the re-engineering its key business process. Considering the modeling, specification and implementation steps of dedicated system such as PDM, ERP, or SCM, UML can be considered as an unquestionable supplementary modeling 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


