Data Integration in a PLM Perspective for Mechanical Products

Sihem Mostefai[1], Abdelaziz Bouras[2], and Mohamed Batouche[1]
[1]Department of Computing Science, University Mentouri of Constantine, Algeria
[2]CERRAL, IUT Lumière, University of Lyon II, France

Abstract: *One of today's hottest topics in information technology is integration. In this paper, we deal with the problem of data integration in a Product Lifecycle Management (PLM) vision. That is, how to integrate product data throughout the entire product lifecycle, ranging from conception, through design, to manufacture, operation and destruction. This paper presents three approaches studied in the context of mechanical products to show how the problem of integration is dealt with. The study is mainly based on some examples of activities (or phases) taken from the product development lifecycle. Including the entire set of activities is out of the scope of this paper. Nevertheless, the three proposed approaches; meta-data, features, and ontologies show enough flexibility and potential to be generalized quite easily to other phases.*

Keywords: *Multiple view feature modelling, collaborative design, data exchange, PLM, data integration.*

Received February 21, 2004; accepted July 6, 2004

1. Introduction

Mechanical products' design and manufacturing are at the heart of many industrial companies. These activities relied greatly at their first stages on solid modelling techniques which aimed at giving a complete and non-ambiguous representation of mechanical parts to be easily usable by the product engineering processes. However, these representations suffered major limitations: They provided initially low level semantics and lacked relevant information for different downstream applications. This has led to a shift from solid modelling to product modelling or more generally product development.

The term "product development" actually includes a variety of business processes associated with taking a product from a concept to reality. These include requirement management, design, engineering, manufacturing, production, maintenance and more. Ideally it includes all the activities appearing in the product lifecycle. This is where Product Lifecycle Management (PLM) comes in. PLM delivers application support for conceptual design, design engineering, manufacturing planning, service and maintenance. It is designed to neatly connect to and accommodate the workings of other processes, including: Customer Relationship Management (CRM), Supply Chain Management (SCM), Enterprise Resource Planning (ERP). The benefits of PLM are realized once previously disparate systems such as those stated above, are integrated.

This paper aims to propose some solutions to the problem of data integration for a PLM perspective and to achieve some of its benefits. The paper is organized as follows. Section 2 gives a more profound investigation on the PLM concept and some of the main problems encountered in the integration process. Section 3 gives our solution through three approaches; meta-data, features, and ontologies. Finally, a conclusion is given with a discussion about the efficiency of the proposed approaches and some interesting perspectives in section 4.

2. PLM Provenance and Scope

This section is concerned with the concept of PLM. We explain how, driven by changes in the market and advances in technology, PLM has emerged in the early years of the 21st Century as the way to develop, sell and support products in the DASAMASA environment. The acronym for Develop Anywhere Sell Anywhere Manufacture Anywhere Support Anywhere DASAMASA is the aim of leading manufacturers in these early years of the new millennium.

2.1. Historical Background

A brief overview of the history of the second half of the 20th century and the most significant events that gave birth to PLM is given here. Each decade had its own issues, systems and solutions, but the product development and product support/service processes rarely had high priority among business managers until recently. The situation could be summarized as illustrated in [14] to the following:

- The 50s and 60s: Following on from a World War, these were golden years for manufacturing industry.

Demand exceeded production capability. Most industrial goods were sold in the land where they were designed and produced. Computer Aided Design (CAD) systems, were in their infancy.

- The 70s and 80s: The Oil Shock led to inflation and currency fluctuations. Companies in the West were worried by the increasing presence of high-quality, low-cost Asian products. Networks became increasingly powerful. Computer Aided Design and Manufacturing (CAD/CAM) systems were no longer centralized. Product Data Management (PDM) appeared. Most companies weren't aware of customer requirements, hence, customers received products with unwanted functionality and little support.
- The 90s: Business process reengineering led companies to review their product development processes. Globalization hit. A wave of imports from low-cost countries led to the price of goods dropping. In response, production was outsourced to low-cost countries. Concurrent engineering morphed into collaborative development. The World Wide Web, e-commerce, B2B, and trading exchanges appeared. CAD functionality became a commodity. PDM brought some order to all the product data. Customer focus became a buzzword. Some way of enabling real service was needed. Product Lifecycle Management appeared.
- 2001: Governments and consumers are forcing manufacturers to focus on the entire product lifecycle right through to recycling. Air freight, travel, telecommunications, video conferencing and the Web make it easy to work with people anywhere.

2.2. Definitions and Main Objectives

The PLM concept originates from the PDM concept. The later is the acronym of Product Data Management. PDM systems were developed in the mid 1980s to support the management of CAD centric data and workflow, they were mainly used in a centralized way in the engineering and the manufacturing organisations. Despite their undeniable usefulness in the field of CAD/CAM integration, PLM systems had two major drawbacks:

- Accessibility for other than design engineering departments was difficult since PDM was CAD centric. The software was not usable without extensive training and the files could only be viewed in the native CAD system.
- Accessibility from outside the organization was difficult. Therefore, the result was often a heterogeneous fragmented multi-system environment.

For these reasons, PDM systems have been subject to many improvements that finally led to the development of a new class of software: Product Lifecycle Management, or PLM for short.

PLM is intended to manage the product across its entire lifecycle both inside and outside an organisation. This is the main feature that distinguishes it from PDM. The difference between PLM and PDM is discussed in more details in [13].

In order to have a better idea of the PLM concept, we need a sound definition. However, this is not an easy task since such definitions abound in the PLM literature, here are some of them:

- PLM is the business activity of managing an organization's products all the way across their lifecycles in the most effective way. It helps a company get its products to market faster, provide better support for their use, and manage end-of-life better [3].
- A PLM system can be described as an enterprise-wide Information Technology (IT) "infrastructure to support management of product definition throughout its complete lifecycle" (from initial concept to product obsolescence) [11].
- PLM means "the management of comprehensive, accurate and timely information over the entire product lifecycle" in order to realise collaborative product development [5].

PLM systems provide a consistent view on product related information in the extended enterprise and among geographically dispersed individuals and groups. Consequently PLM is defined as a class of software and services that uses Internet technologies to permit individuals – no matter what role they have in the commercialization of a product, no matter what computer based tool they use, no matter where they are located geographically or within the supply net - to collaboratively develop, build and manage products throughout the entire lifecycle [5].

PLM requires a holistic approach melding product-related application systems, data, processes, techniques and skills [3] as shown on Figure1. Consequently, different data types and formats associated to these applications, ranging from text files to geometric CAD files, simulation data, administrative data… need to be integrated. For all these data to be integrated correctly in a PLM vision, we need methods to organize, store, access, convert and exchange these data correctly and seamlessly to their users: This is interoperability. So the ultimate goal of a PLM system is not achievable before a real data organization and integration. That's our major preoccupations in this work.

3. Collaboration as a Solution to Data Integration

The Main objective of this work is to propose a solution for the problem of data integration in a PLM system consisting of many partners, each of which

acting in an activity of the product lifecycle The whole system can be viewed as a network of activities exchanging data and information with each other. Our principal preoccupation is the study of some mechanisms that allow an efficient exchange and cooperation between these partners. In order to achieve this goal, we have opted for a collaboration solution that we shall illustrate through 3 approaches, as shown on Figure 2, namely:

- The meta data approach.
- The features' approach.
- The ontological approach.

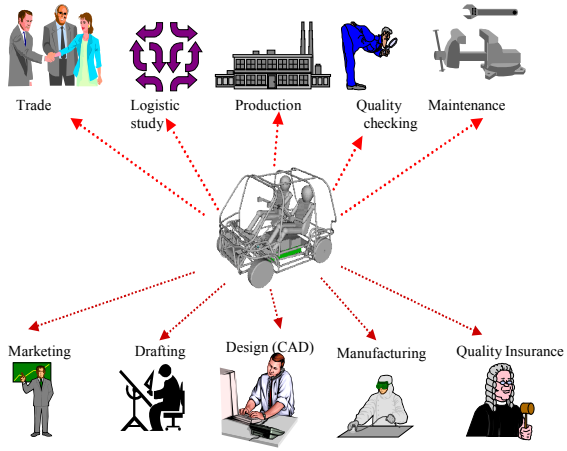

Figure 1. Some of the activities appearing in a product lifecycle.

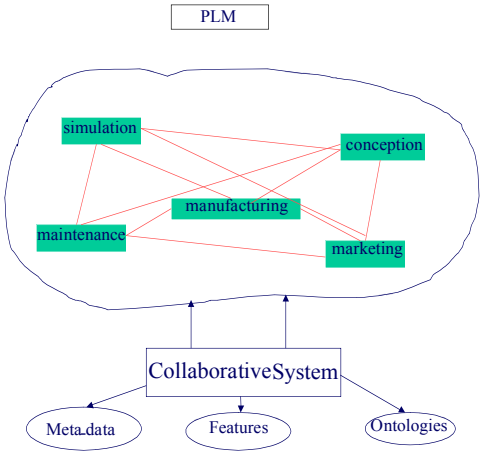

Figure 2. Collaboration as a mechanism for integration.

3.1. The Meta-Data Approach

Historically, product data management systems primarily focused on managing engineering drawings and CAD files in the design phase of product development. However, this view has been expanded to include more activities throughout the product lifecycle ranging from early design phases to product maintenance. One of the problems encountered in this field is the standardization needed to exchange data of different types and formats between the diverse applications. That is where we need to use a meta-data approach which must rely on a mechanism to represent the data exchanged whatever format it is, to convert it to a standard format (the meta-data) and to ensure its proper exchange between partners where it can be reconverted to its original format and processed conveniently. Achieving this aim requires the use of a methodology that is adaptable to the current situation and extensible enough to meet future requirements and new technologies as they emerge. To ensure broad adoption, the technology selected needs to be widely and freely available. The Extensible Markup Language (XML) developed by the World Wide Web Consortium [16] provides such a freely available, widely transportable methodology for well-controlled data interchange. XML and its related Document Object Models (DOM) standards [17] are provided with mechanisms (APIs) to access and manage data represented in XML.

Advances in 3/3 based architectures middleware technologies have greatly contributed to improve "person to person" collaborative applications, by enabling people in different locations to work together as if they were in the same office, exchanging seamlessly information in different formats. A description of some mechanisms that allow such exchanges, based on XML and Java technologies, can be found in [2].

In the present approach, we have focused on the collaborative aspect and created a prototype intended to include the partners implied in four CAD activities or views typical to the product lifecycle taken as an example, namely: Conceptual design, assembly design, part detail design and part manufacturing design. More details on these views can be found in [10]. The main tools that we have used in the implementation of this prototype are: 3/3 based architecture, Enterprise Java Beans (EJBs) from Java technology and XML as an exchange format. As a result, we have developed a multi-platform environment containing the previously described views and a technical database containing the product information necessary to these views and to downstream applications. The architecture of our system is divided into 3 tiers or layers, as shows in Figure 3:

- The first tier: User interface layer is dedicated to the designers of different kind (conceptual phase, assembly…) in the client side. Users are connected through a web browser to visualize all the storage knowledge needed via a convenient graphical user interface. Before being transferred to the second and the third tiers, the data gathered from this tier are translated first to XML format, then they are sent to the other tiers. Conversely, when data are received from the other tiers, they are in XML format and are translated back to the original format

local to the client. It is precisely in this tier that the meta data approach is best illustrated through the use of XML.

- The second tier: Multi-view interface layer. This is mainly the middleware layer and consists of a set of software programs in the server side, supported by Java technology (java servlet and Enterprise Java Beans), which control the inter-operability between the databases and the client system and select elements and download them as requested by the different users.
- The third tier: Database layer contains remote common database in the server side from which data is exported to the clients, in order to respond to their requests or to be used in building their own local databases for instance. The access method to the database depends on its type and organization and we have opted for a relational database for their availability and the ease of their use.

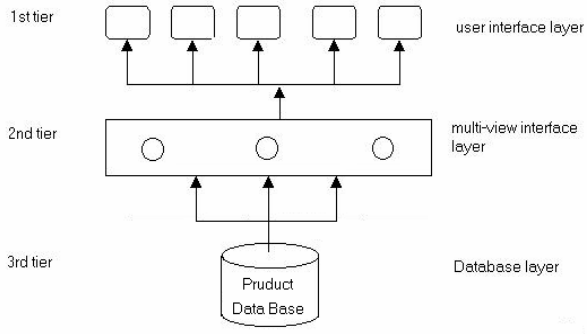

Figure 3. Proposed architecture for the meta-data approach.

More details of this approach can be found in [9]. This prototype has been tested with different categories of users, each of which is authenticated via a user name and a password then he web server creates a personalized session for each kind of user depending on its category and profile. The prototype has been tested with different profiles of users and at different levels. Validation tests have not been completed yet but we intend to extend them to the entire product database in the near future. The approach is flexible and extendible to other views in the product lifecycle. The flexibility is illustrated by the use of a 3/3 architecture and the separation between the second and the third tier, so that it is easy to add other views by only providing the program units and treatments that handle these views (mainly servlets and EJBs) without reconstructing the whole system from scratch. This makes this approach scalable and adaptable to the growing needs of its users.

3.2. The Feature Approach

The feature approach is based on the concept of feature. What is meant by a feature? Many definitions of a form feature or a feature for short in the fields of CAD/CAM can be found in the literature. One of them is the following: "a feature is a representation of shape aspects of a product that can be mapped to a generic shape which is functionally significant for some product lifecycle phase" [4]. Another definition is that a feature is "a basic entity that couples the geometric details of a product to their meaning according to a viewpoint" [6]. A viewpoint means generally one phase of the product lifecycle.

The form feature concept originally defined for manufacturing purposes is a basic tool in product modeling environments [12], as it couples geometric details with their meanings in application contexts. Independently of the application, a feature model can be obtained in one of four ways, namely:

- Automatic feature recognition: This approach is historically the first one to provide feature models. It extracts features (generally manufacturing ones) from a geometric model (generally a Boundary representation Brep).
- Design by features: Primitive features are incorporated during the design phase via parameterized libraries of features. The user utilizes them directly as design primitives. The main drawback of these methods is that features are domain dependant, so a feature conversion mechanism is required whenever many views have to co-exist.
- Feature conversion: Assumes an existing feature model belonging to a particular view and converts it to a model of another one. This is indeed one of the most suitable ways for collaborative environments consisting of multiple views to co-exist. However, it is not an easy task to provide such conversion mechanisms. Most of them are restricted to at most two views, without a possibility of generalization.
- Hybrid approaches do combine two or more of the previous methods described above, generally because of the insufficiencies of a single one. See [6] for more details and examples.

The feature approach for integrating multiple views that we adopted is based on a common geometry shared by several applications which is interpreted as a set of features holding a special semantic for each view. Thus, each actor in the product lifecycle has his own descriptive language based on features. Moreover, one may see the product model as an overlapping of several feature-based models in which each model relates to one actor viewpoint in the product lifecycle. This approach aims to connect geometric data to different product semantics by using the shared geometry handled by a semantic analyzer that uses composition and abstraction techniques under viewpoint rules [15]. We can illustrate this approach through a simple example shown on Figure 4. This part may actually have multiple interpretations and associated semantics depending on the point of view, for the manufacturing

view for instance it is seen as a block piece with a rotational depression constituted of a blind hole formed by a couple of co-linear cylinder and cone. The semantic is completed by adding information on how to thread the part, or about the material to be used for instance. For the assembly design view which is concerned with the description of the physical connections between parts, the same part is interpreted with an emphasis on the assembly properties through the surfaces of contact involved. The semantic is completed by adding information on how to connect the part if it had to be connected to another one (a screw for instance).

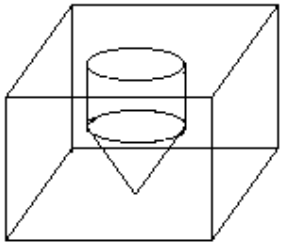

Figure 4. Example part.

One application of this feature based-integration is presented in [1], the authors present a form of features based tool to aid the integration of analysis during the design process. It allows producing an analysis model out of a part solid model.

The feature approach presented here is mainly based on the feature concept. This implies that all the product lifecycle phases have to be described in terms of specific features. This description is not always available indeed. Except for the phases of design, assembly or manufacturing where this concept has an explicit meaning related to the product geometry, for other phases like simulation or maintenance, specific feature definitions have to be supplied. Hence a feature would not only mean gathering geometric information and giving them a special meaning, but rather presenting new concepts with a higher degree of semantic specific to a product lifecycle phase. Making such an emphasis on semantics leads to the next approach: The ontological approach.

3.3. The Ontological Approach

Just like the features approach is based on a common geometry the ontological approach is based on the idea of a common knowledge or a common semantic shared by the views. Observing that schemas representing the views of various engineering disciplines are rarely disjoint and contain equivalent information content, but each might represent it differently. For example, a finite element analysis of a component and a design view of the same component may each make reference to the component's structural parts. These references may be to information units in their respective viewpoint, yet the things to which they refer are, or should be the same. The most straightforward solution in engineering processes that share information is then to do so under a common schema one which encompasses each of the viewpoints.

The tendency nowadays is governed by the emphasis on semantics rather than data types or formats, especially in the internet and the way we use it. Thank to ontologies, the semantic web has become a reality. Thus, integration is viewed as a knowledge management process rather than a data or information one.

Integration in a global engineering environment like that of a PLM can then take advantage of this aspect. The approach outlined here wants to illustrate how ontologies could help to support the integration process.

Ontologies can be seen as nowadays most advanced knowledge representation model. The term ontology is borrowed from philosophy where (initially) an ontology is a systematic account of existence. In artificial intelligence, what exists is that which can be represented. An ontology is an explicit specification of a conceptualisation [7].

A conceptualisation is the set of objects, concepts and other entities that are assumed to exist in some area of interest together with the relationships that hold among them. A conceptualisation is an abstract simplified view of the world that we wish to represent for some purpose [7]. Since the process of designing or writing ontologies is an abstraction process, the first preoccupation we had initially is to consider the different phases in the product lifecycle, to take some views for example and to try to abstract them to create their respective ontologies. Then we will consider the common aspects between them in order to create a common ontology based on a common vocabulary and semantic. Nevertheless, this would not be possible without taking into account some important considerations like:

- How well are we meeting requirements?
- What is the product seen from view Y?
- What information does discipline Z need?…

Regardless of the domain, an ontology consists of several components, the most important are: Concepts, relations and attributes, instances and axioms, see [7] for details.

The views we have considered for integration as an example are those described in [10], we can summarize them in the following (concepts are underlined):

- A *product* consists of many *views.*
- A *view* consists of a *feature model* and a library of *feature classes* of which *instances* are created to build the feature model.

Product and views hierarchy

1. A product can be seen as a set of views:
   - View 1: Conceptual design.
   - View 2: Assembly design.
   - View 3: Part detail design.
   - View 4: Manufacturing planning.
2. A view consists of:
   - A library of feature classes.
   - A feature model: Specified from instances of the feature classes in the feature library of the view plus additional entities like model validity conditions.
3. Implicit axioms:
   - View 1 and 2 are related in the sense that they describe the same product.
   - View 3 and 4 are also related for they represent the same part.

4. Discussion and Conclusion

We conclude this work by an assessment of the different approaches we have proposed in order to show their main advantages and drawbacks. These observations are intended to direct our future investigations. Our assessment of this work could be summarized in the following:

- The meta-data approach is quite completely surrounded [9], some of its advantages come from the use of XML which makes it possible to exchange data of whatever format, provided that it is well interpreted by the partners each in his domain. However, among its limitations is inherent to XML itself: Exchange is only restricted to data or information syntax without knowing anything about its semantic, hence, the users have to manage the semantic aspects by themselves and not automatically and that's where this approach fails. Nevertheless, this approach is very flexible and allows adding new phases quite easily.
- The feature approach solves the problem of data integration by considering the concept of feature which has been clearly defined in different fields of CAD/CAM as a set of geometric entities with a special meaning, and this is where this approach shows the maximum of efficiency because it is tightly related to the product geometry. However, the product lifecycle phases are not only interested in the product geometric data but include many other types of information as well, especially in phases like simulation, maintenance… So to cope with such situations, the feature concept has to be redefined to be representative of either geometric or non-geometric data. It has to be sufficiently generic to be adaptable to the semantics associated to the different product lifecycle phases. This emphasis on semantics has led to the adoption of the subsequent approach: The ontological approach.
- The ontological approach has been explained thru an example and is now being implemented. Our main objective was to give an idea of the principle of this approach though real results and assessments have not been obtained yet, this constitutes a basis of a subsequent article we may publish. Nevertheless, we can say that the main strength of such an approach is the semantic aspect, through a higher degree of abstraction. But this is obtained at the price of a good comprehension and abstraction of the information semantics involved in the different lifecycle phases which tend to be more complex.

In summary, each one of the approaches previously described performs better in some typical situations: The meta-data approach is best suited in data exchange, as far as the activities involved in the exchange are interested in preserving just the syntactic aspect of the exchanged data and manage the semantic locally, this is true for administrative or commercial data for instance. The feature approach performs better in applications where the feature concept has a clearly established meaning, typically in CAD/CAM where it is tightly related to the product geometry. On the other hand, the ontological approach is very promising as a means of knowledge management and shows its full potential in all applications mastered by knowledge, with no limit or restriction. This is indeed a new way of "thinking" in the field of data and knowledge integration and is thought of as being "the way" of integration at present and in the near future.

References

[1] Belaziz M., Bouras A., and Brun J. M., "Morphological Analysis for Product Design," *Computer Aided Design*, vol. 32, pp. 377-388, 2000.

[2] Bouras A., Ouzrout Y., and Wacquet M., "Web-Based SCM Information Exchange," *in Proceedings of the 6th Annual International Conference on Industrial Engineering Theory, Application and Practice*, San Francisco, CA, USA, November 2001.

[3] Bourke D., "The PLM Software Scene: Tips for Smart Shopping," *2PLM e-zine*, vol. 6, no. 12, January 2004.

[4] Bronsvoort W. F., Van Den Berg E., Bidarra R., and Noort A., "Essential Developments in Feature Modelling," *in Proceedings of the CAD/Graphics'2001, Seventh International Conference on Computer Aided Design and Computer Graphics*, Kunming, China, pp. 6-15, August 2001.

[5] Brück H. J., "The Impact of Organisational Change Management on the Success of a PLM

Implementation," *Master Dissertation*, University of Lincoln, March 2002.
[6] De Martino T., Falcidieno B., Giannini F., Hassinger S., and Ovtcharova J., "Feature-Based Modelling by Integrating Design and Recognition Approaches," *Computer Aided Design*, vol. 26, no. 8, pp. 646-653, 1994.
[7] Gruber T. R., "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199-220, 1993.
[8] Kifer M., Lausen G., and Wu J., "Logical Foundations of Object-Oriented and Frame-Based Languages," *Journal of the ACM*, vol. 42, no. 4, 1995.
[9] Mostefai S., Batouche M., and Bouras A., "Multiple View Feature Modelling in a Collaborative Environment," *in Proceedings of the Conférence Internationale sur la Productique (CIP'2003)*, Algeria, October 2003.
[10] Noort A., "Multiple View Feature Modelling with Model Adjustment," *PHD Thesis*, Tudelft University, The Netherlands, 2002.
[11] Portella J., "Collaborative Management of the Product Definition Lifecycle for the 21st Century," *in Proceeding of the PDT Europe Conference*, Noordwijk, May 2000.
[12] Pratt M. J. and Wilson P. R., "Requirements for Support of form Features in a Solid Modelling System," *CAM-1 Report R-85-ASPP*, June 1995.
[13] Stackpole B., "There's a New App in Town," *CIO e-zine*, http://www.cio.com/archive/051503/app.htm, May 2003.
[14] Stark J., "Of CPC/PLM, DASAMASA and the Slim-Line Corporation," *2PDM e-zine*, vol. 4, no. 5, October 2001.
[15] Tehari A., "Morphological Analysis of Models for Feature-Based Product Description," *PHD Thesis*, LIGIM, Claude Bernard University of Lyon 2, January 1999.
[16] World Wide Web Consortium, "Extensible Markup Language (XML): W3C Recommendation," http://www.w3.org/org/TR/REC -xml.
[17] World Wide Web Consortium, "XML and its Related Document Object Model: DOM," http://www.w3.org/TR/REC-DOM-LEVE L-1.

Sihem Mostefai is a lecturer in the Computing Science Department at University Mentouri of Constantine, Algeria. She received her MSc in computer graphics in 1993, in addition to an engineer Diploma in computer systems in 1989 from the same university. She is also preparing a PhD in product lifecycle management applied to the case of mechanical products. Her research interests are in CAD/CAM, feature-based modelling, and data exchange.

Abdelaziz Bouras is a full professor at IUT Lumière Lyon II, and head of CERRAL Center, at PRISMa Laboratory, Claude Bernard University, Lyon, France. He was in 1998 at the Manufacturing Engineering Laboratory (MEL) of National Institute of Standards and Technology (NIST) as a guest researcher. He received his PhD in computer science from Claude Bernard University in 1992 and his engineer Diploma in mechanical engineering in 1987 from The National Institute of Mechanical Engineering (INGM) of Boumerdas, Algeria. His research interests are in solid and feature based modeling, collaborative design, and data exchange.

Mohamed Batouche received his MSc and PhD degrees in computer science from the Institut National Polytechnique de Lorraine, France, in 1989 and 1993, respectively. Currently, he is a full professor at the University Mentouri of Constantine, Algeria. His research areas include artificial intelligence and pattern recognition.