Proceedings of DETC'97
1997 ASME Design Engineering Technical Conferences
September 14-17, 1997, Sacramento, California

DETC97/EIM-3720

PRODUCT DATA MANAGEMENT SYSTEMS: STATE-OF-THE-ART AND THE FUTURE[1]

Taner Bilgiç
Enterprise Integration Laboratory
Department of Mechanical and Industrial Engineering
University of Toronto
Toronto, Ontario, M5S 3G9, Canada
Email: taner@ie.utoronto.ca

Dennis Rock
Boeing Defense and Space Group
Helicopters Division
Philadelphia, Pensylvenia
Email: dennis.rock@boeing.com

ABSTRACT

We survey the current state-of-the-art in (commercial) Product Data Management (PDM) systems. After identifying the major functions of PDM systems, we indicate various shortcomings of the current PDM technology. An important shortcoming is in the representation and use of functions. We review the functional representation literature in the context of PDM technology. Systems management aspects of an engineering project is also commented on. We believe these two areas are the next two challenges awaiting PDM technology in the near future.

PRODUCT DATA MANAGEMENT

Product Data Management (PDM) systems are tools that help engineers manage the data and the processes related to product development life cycle. As sophisticated and automated design tools (e.g. CAD systems) became available, the amount of data accumulated about the designed artifact increased dramatically. PDM systems offer a technology to answer the need of managing such data. PDM systems keep track of various product data that already exists in the enterprise in various forms. One can view PDM systems as having *meta-knowledge* about the product development life cycle. This meta-knowledge is in the form of knowledge about product structure, processes, and access/change management rules.

Basic (desired) functions of a PDM system are:

- **Design Release Management:** the process of controlling design data with an *electronic vault* with check-in/check-out, release level maintenance, access security and review and approval management. This function encompasses the management of all forms of digital product data—CAD files, geometric models, images, documents, etc.
- **Product Structure Management:** the ability to define, create, maintain, modify and display multiple versions of the product structure, including design options and activities over the product data life cycle.
- **Change Management:** the ability to define and manage changes to product data over the life cycle. Change Management is process oriented, defining the events in the cycle of reviewing and approving changes.
- **Classification:** ability to classify parts by their structure, function or processes for manufacturing.
- **Systems Management:** usually perceived as the use of project-oriented scheduling techniques with work breakdown structures but which should be able to manage any facet of systems design (cost, quality, risk, in addition to work flow).
- **Impact analysis:** the ability to detect the effects of a design change to the overall product design life cycle.

Most of the commercially available (see the PDM information center[2]) PDM systems provide the first three functionalities and the classification to a degree. The PDM systems are trying hard to have an edge in the market by con-

[1]THIS WORK IS SUPPORTED BY DARPA RADEO PROJECT CONTRACT NUMBER 70NANB6H0074

[2]http://www.pdmic.com

centrating on the workflow and their "document-centered" view of the design process. That is, PDM systems view the design process as managing documents (reports, drawings, analyses, etc.) throughout the design life cycle through a vaulting mechanism to facilitate making the right data available to the right person at the right time. Although this is a much needed functionality in practice the current Product Data Management conceptualization has several shortcomings.

SHORTCOMINGS OF PDM SYSTEMS

Ambiguous product representation

The PDM systems do not have a formal representation of the product that unambiguously describes its function, behaviour and structure. All systems can represent the Engineering Bill of Materials and the Manufacturing Bill of Materials and usually leave the representation at that level. Most of the valuable information about the products (function, behaviour, requirements, geometric representations, etc.) stays buried in the "documents" the PDM system is managing. This type of "deeper" and richer knowledge does not come to the surface in the current PDM technology. We will comment more on the function representation later.

The product representation in the PDM systems must be as thorough (at least consistent if not complete) as possible in order to answer "common sense" queries. One approach for such an approach is to use ontologies which are agreed upon conceptualizations and definitions of the domain. By providing axioms in the definitions one can answer common sense queries easily by reasoning on these axioms (Fox 1993, Fox & Gruninger 1994, Bilgiç, Chionglo, Fox, Gupta, Gwidzka, Leizerowicz, Lin & Sen 1996)

PDM vendors are starting to see the importance of this level of detail by promising using emerging international standards (STEP) in the future. However, in order for the PDM technology to tackle this problem it has to represent this "deeper" knowledge about products. Usually such knowledge stays embedded in the CAD system. Using international standards to better couple PDM systems with the CAD systems is one path to follow.

Lack of impact analysis

PDM systems, by concentrating on workflow and document management, fall short in the analysis of the *impact of proposed changes.* This analysis must be a key step in the process.

This shortcoming is closely tied to the question of level of representation mentioned above. Since PDM systems mainly concentrate on managing *documents* and do not explicitly represent what is in those documents, they cannot measure the impact of design changes. In order to measure such impacts one needs a more detailed representation which gives rise to a constraint system (see for example the UTKAD system (Bilgiç *et al.* 1996)).

Lack of functional classification

One of the major challenges to what PDM systems can do is what is called the *functional classification*. To capture the product structure (i.e., the component hierarchy in discrete manufacturing) is mostly not the biggest problem of PDM systems. What is more important to know is what the components are for! This requires notions of function and behaviour and a classification of products based on those notions. This is not captured in any way with the current PDM technology.

It is important to distinguish why there is a need to represent functions. In many cases, one only requires a *classification* of the product by its functions. This is particularly the case when a certain product structure is going to be *reused* for a new design. However, sometimes a representation of function is required *to ask "what if" questions* (e.g. what happens to the payload attached at the end effector if the shoulder joint provides a rotation of 60°?[3]. This requires the ability to *reason on* functions.

The representation of functionality in design is a challenging and active research area. We give a very brief survey of the area later in the paper emphasizing the objectives of representation within the PDM framework.

Lack of reuse of design knowledge

Another shortcoming of the PDM technology is in reusing engineering design knowledge. Once again, this is closely tied to the level of representation and particularly to the lack of functional representation.

Engineering design knowledge reuse is usually formulated as a case-based reasoning or analogical reasoning problem. The main idea in case-based reasoning is to *represent a set of cases* which encode design information (most importantly lessons learned, successes and failures as well as structural, functional, and behavioural representations) and then when a new design problem appears to *match* it to the case-base to find a "similar" solution, *adapt* the case for the new problem at hand, and *store the new case* in the case-base.

[3]It should be noticed that this question is more of a qualitative nature than a question like 'What is the effect of using a hydraulic brake system on the end effector under given conditions?' which will require a full-fledge, quantitative simulation of the artifact.

Although current PDM technology is capable of representing a flat case library which is comprised of documents, drawings and annotations, for most engineering work, this level of detail is not sufficient to build engineering case-bases (Maher, Balachandran & Zhang 1995).

Lack of complete systems management

PDM technology mainly concentrates on the workflow and attempts to manage the project schedule. This temporal management is only one phase of the general systems engineering framework. Particularly, resource (cost, capacity, etc), performance (technical performance, quality, etc.) and risk management are overlooked.

There is no doubt that providing for such management tools calls for richer knowledge representation than that is currently available in the PDM technology.

Another challenging aspect of the systems management appears in the presence agent-based architectures. When many human and/or software agents are collaborating in engineering design problems the systems management problem becomes more complex. In this case, effective communication and timely notification arise as critical components of the systems management.

Extending/renewing the technology to perform better systems management functionality is another hurdle awaiting PDM technology in the near future.

REPRESENTATIONS OF FUNCTION

In this section we give a brief survey of approaches to represent function in engineering design. Determining a functional representation is usually possible using a top-down approach. First, the overall task of the artifact is identified and then the overall function corresponding to the overall task is decomposed into sub-functions (Pahl & Beitz 1988, Prasad 1996). The resulting structure is called the *function structure*.

It is important to identify *why we need to represent functions*.

- classification: In engineering design, sometimes there is a need to *store* artifacts with respect to what they do. In such a case, the function representation is part of the product representation. It is another *view* of the product information.
- finding a design solution: Design search space is usually traversed by function. The designer has a function in mind and the conceptual stage of the design is dominated by searching for the right concept which provides the right functionality.
- design validation: When the aim is to validate a design concept one can take a detailed simulation approach or simply use qualitative simulation at the generic functional level.

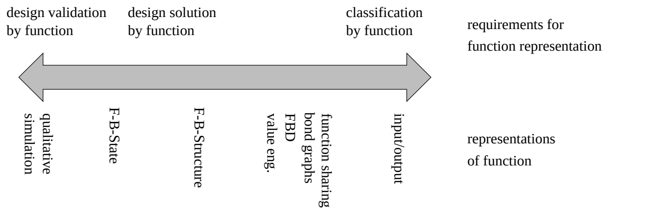

Figure 1. A CLASSIFICATION OF FUNCTION REPRESENTATIONS

Each of these reasons requires a different level of function representation.

Among the many representations of function the following are outstanding for engineering design work: function as input/output (Pahl & Beitz 1988, Sturges(Jr.), O'Shaughnessy & Reed 1993, Sturges, O'Shaughnessy & Kilani 1996, Sasajima, Kitamura, Ikeda & Mizoguchi 1995), function sharing (Ulrich & Seering 1988), value engineering (Miles 1972), influence diagrams representations (Sycara & Navinchandra 1989, Sycara & Navinchandra 1992), bond graphs (Karnopp & Rosenberg 1975, Finger & Rinderle 1989, Bracewell, Chaplin, Langdon, Li, Oh, Sharpe & Yan 1996), function-behaviour-state (Umeda, Tomiyama & Yoshikawa 1990, Tomiyama, Umeda & Yoshikawa 1993, Umeda, Ishii, Yoshioka, Shimomura & Tomimaya 1996), function-behaviour-structure (Gero 1990, Qian & Gero 1996), qualitative simulation (Kleer & Brown 1984, Forbus 1984).

Figure 1 depicts a summary of the appropriateness of different representations to various requirements for representation.

It should be noticed that this classification is not absolute (i.e., input/output representation can be used in design validation to a degree but qualitative simulation is a better representation for that task.)

For PDM systems an input/output model for the purposes of classification seems appropriate. Further reasoning, which requires richer representations are usually performed on the CAD system.

Table 1. PDM COMPARISON: MODELLING

Data Modelling		Meta phase II (SDRC)	Sherpa	OPTEGRA (Computer vision)	Meta Phase	Work Group
Activities	Abstraction	?	?	?	?	?
	Time	?	?	?	?	?
	Causality	?	?	?	?	?
	Simulation	?	?	?	?	?
	Versions	?	?	?	?	?
	BOM	Y	Y	Y	Y	Y
Part & Product	Versions/revisions	Y	Y	Y	Y	Y
	Constraints	N	N	N	N	N
	Requirements	?	N	N	N	N
	Structure	Y	Y	Y	Y	Y
Organization	Roles	?	?	?	?	?
	Responsibilities and empowerment	?	?	?	?	?
	Plans	?	?	?	?	?
	Committed/Used	?	?	?	?	?
Resources	Continuous/ Discrete	?	?	?	?	?
	Pools/Individual	?	?	?	?	?
	Resources	N	N	N	N	N
Cost	Actual	N	N	N	N	N
	Project	N	N	N	N	N
	Traceability	?	?	?	?	?
Quality	Specification	?	?	?	?	?
	ISO9000	Y	Y	?	?	?
Document		Y	Y	Y	Y	Y

COMPARISON OF VARIOUS COMMERCIAL PDM SYSTEMS

We have reviewed some[4] of the commercially available PDM systems on several categories. This comparison is given in Tables 1 and 2. The major categories of the comparison are Data Modelling, Query Processing, Analysis, Design Management and administrative tasks which are all claimed to be provided by commercial PDM systems. The choice of sub-categories in each category is made so that we can compare the PDM framework with the UTKAD system we are developing at the University of Toronto.

Stemming from an inadequate representation of product, project and organization knowledge, the shortcomings of the current PDM technology alluded to earlier in the paper is reflected in Tables 1 and 2.

[4]We used the PDM Information Center Survey available from http://www.pdmic.com/sur1ares.html and chose the systems with more than 10% market share as indicated in that survey.

Table 2. PDM COMPARISON: OTHER AREAS

		Meta phase II (SDRC)	Sherpa	OPTEGRA (Computer vision)	Meta Phase	Work Group
Query Processing						
Query processing	Parametric retrieval retrieval	Y	Y	Y	Y	Y
	Deductive query processing	N	N	N	N	N
	Case-based retrieval	N	N	N	N	N
Analysis						
Analysis	Impact Analysis	N	N	N	N	N
	Cost Analysis	N	N	N	N	N
	Risk Analysis	N	N	N	N	N
Design Mgmt.						
Information Distribution	Subscription services	Y	Y	Y	Y	Y
	Change notification	Y	Y	Y	Y	Y
	Planning	N	N	N	N	N
Project Management	Scheduling	N	N	N	N	N
	Execution (work flow mgmt.)	Y	Y	Y	Y	Y
	Work history	Y	Y	Y	Y	Y
Product Management	Version mgmt.	Y	Y	Y	Y	Y
	Constraint propagation	N	N	N	N	N
	Concurrency	N	N	N	N	N
	Collaboration & Negotiation	N	N	N	N	N
Requirements management		N	N	N	N	N
Quality management		N	N	N	N	N
Cost management		N	N	N	N	N
Risk management		N	N	N	N	N
Interface	Graphical visualization	Y	Y	Y	?	Y
	Data translation	Y	Y	Y	Y	Y
	STEP/EXPRESS	Y	?	?	Y	?
	MRP (interface)	Y	Y	Y	Y	Y
	DBMS	Y	Y	Y	Y	Y
	KQML/KIF (inter-agent comm.)	N	N	N	N	N
	Image services	Y	Y	Y	Y	Y
System administration		Y	Y	Y	Y	Y

THE UTKAD SYSTEM

At the University of Toronto, we have been developing a Knowledge Aided Design (UTKAD) system to support the engineering design process. The full details of the UTKAD system are given in (Bilgiç *et al.* 1996). In this section we briefly summarise the UTKAD system from the viewpoint of the categories we used in the PDM comparison emphasizing what has been achieved and avenues to pursue for further research.

Main objectives of the UTKAD system are:

- Provide a representation that stores, integrates and manages the various types of design knowledge. It is important that engineers work in a common language and representation so that their designs can be integrated without conflicts in the underlying semantics. The representation should be able to model and represent information such as requirements, versions, design rationale, etc. and have the ability of reasoning about them.
- Provide a shared environment in which engineers can explore space of alternative designs and communicate their design in a uniform manner into the shared design. The environment should provide each engineer with a private working space where the engineer can explore his design in his own will while the work of different engineers can be integrated into the shared design through a common protocol.
- Reuse past design knowledge, experience and lessons learned in new design situations to reduce the overall life cycle of the design.
- Manage the systems engineering process by providing adequate communication and coordination capabilities, in order to improve the productivity and quality of design process.
- Ease the access to and acquisition of information and knowledge from the representation. Acquiring design information/decisions is difficult due to the barriers existing between engineers and computers. If the information technology is to be a design process participant, it must address the barriers to the adoption of technology by engineers.

To achieve these goals, the TOVE ontologies[5] on products, activities, organization, resources, cost, quality and document are provided in a *Knowledge Network* which is central to the system (see Figure 2). The Knowledge Network incorporates a host of services like access management, query management, constraint management which makes it *active* (by way of a constraint network) and *deductive* (by reasoning on the axioms of the definitions). Two subsystems work on the Knowledge Network: the Case-Based retrieval (to reuse engineering knowledge) (Bilgiç & Fox 1996) and the Systems Management Agent (SMA) (to manage different facets of the systems engineering process). The system has a user interface written in Java which may be accessed over the Internet and an Electronic Engineering Notebook (EEN) interface to ease the acquisition and access to Knowledge Network and other services.

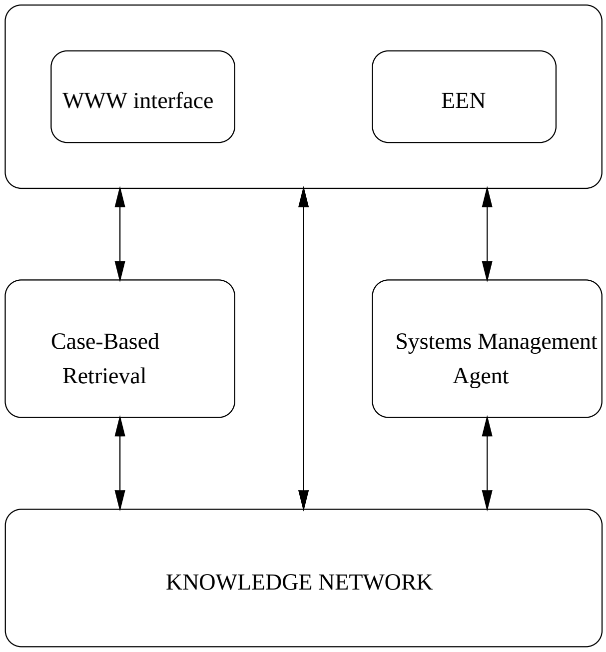

Figure 2. ARCHITECTURE OF THE UTKAD SYSTEM

CONCLUSIONS

The UTKAD system is developed to have a 'Y' for all rows of Tables 1 and 2. However, it is a research prototype and leaves much room for improvement. Deriving from the UTKAD experience and the survey of the PDM technology presented in this paper, we envision two major challenges to the PDM technology in the near future:

- Richer knowledge representation on which impact analysis, case-based reasoning and functional classification modules can be built.
- Better systems management including cost, capacity, quality, and risk management which is able to manage various facets of the systems engineering in the presence of software/human agents.

[5]TOVE ontologies are described in an on-line manual accessible from `http://www.ie.utoronto.ca/EIL/comsen.html`. The same URL also contains a link to the Ontolingua code for all TOVE ontologies.

REFERENCES

Bilgiç, T., Chionglo, J., Fox, M. S., Gupta, L., Gwidzka, J., Leizerowicz, W., Lin, J. & Sen, L. V. (1996). Knowledge-aided design-in-the-large. Technical Report describing the UTKAD system. Submitted for publication.

Bilgiç, T. & Fox, M. S. (1996). Constraint-based retrieval of engineering design cases: context as constraints, *in* J. S. Gero & F. Sudweeks (eds), *Artificial Intelligence in Design '96*, Kluwer Academic Publishers, Dordrecht, pp. 269–288. Proceedings of AI in Design '96, June 24–27 1996, Stanford University.
*http://www.ie.utoronto.ca/EIL/public/aid96/cbret1.html

Bracewell, R. H., Chaplin, R. V., Langdon, P. M., Li, M., Oh, V. K., Sharpe, J. E. E. & Yan, X. T. (1996). Integrated platform for AI support of complex design (part 1): Rapid development schemes from first principles, *in* J. E. E. Sharpe (ed.), *AI System Support for Conceptual Design*, Springer, London, pp. 170–188.

Finger, S. & Rinderle, J. R. (1989). A transformational approach to mechanical design using a bond graph grammar, *in* W. H. Elmaraghy, W. P. Seering & D. G. Ullman (eds), *Design Theory and Methodology – DTM'89*, pp. 107–115.

Forbus, K. (1984). Qualitative process theory, *Artificial Intelligence* **24**(3): 85–168.

Fox, M. S. (1993). Issues in enterprise modelling, *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Le Toquet, France.

Fox, M. S. & Gruninger, M. (1994). Ontologies for enterprise integration, *Proceedings of the 2nd Conference on Cooperative Information Systems*, Toronto, Ontario.
*http://www.ie.utoronto.ca/EIL/public/onto.ps

Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design, *AI Magazine* **11**(4): 26–36.

Karnopp, D. & Rosenberg, R. (1975). *System dynamics: a unified approach*, John Wiley and Sons, New York.

Kleer, J. & Brown, J. S. (1984). A qualitative physics based on confluences, *Artificial Intelligence* **24**(3): 7–83.

Maher, M. L., Balachandran, M. B. & Zhang, D. M. (1995). *Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, USA.

Miles, L. D. (1972). *Techniques of Value Analysis and Engineering*, McGraw-Hill, New York.

Pahl, G. & Beitz, W. (1988). *Engineering Design: a systematic approach*, Springer-Verlag, Berlin. Translated by A. Pomerans and K. Wallace.

Prasad, B. (1996). *Concurrent Engineering Fundamentals: Integrated product development*, Vol. 2 of *Prentice-Hall international series in industrial and systems engineering*, Prentice-Hall, New Jersey.

Qian, L. & Gero, J. S. (1996). Function-Behaviour-Structure paths and their role in analogy-based design, *AIEDAM* **10**(4): 289–312.

Sasajima, M., Kitamura, Y., Ikeda, M. & Mizoguchi, R. (1995). FBRL:a function and behavior representation language, *Proceedings of the IJCAI '95*.

Sturges, R. H., O'Shaughnessy, K. & Kilani, M. I. (1996). Computational model for conceptual design based on extended function logic, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing* **10**(4): 255–274.

Sturges(Jr.), R. H., O'Shaughnessy, K. & Reed, R. G. (1993). A systematic approach to conceptual design, *Concurrent Engineering: Research and Applications* **1**(2): 93–106.

Sycara, K. P. & Navinchandra, D. (1989). Integrated case-based reasoning and qualitative reasoning in engineering design, *in* J. S. Gero (ed.), *Artificial Intelligence in Design*, Springer-Verlag, New York, pp. 231–250.

Sycara, K. P. & Navinchandra, D. (1992). Retrieval strategies in a case-based design system, *in* C. Tong & D. Sriram (eds), *Artificial Intelligence in Engineering Design*, Academic Press, New York, NY, chapter 4, pp. 145–164. (Volume II).

Tomiyama, T., Umeda, Y. & Yoshikawa, H. (1993). A CAD for functional design, *Annals of the CIRP* **42**(1): 143–146.

Ulrich, K. T. & Seering, W. P. (1988). Function sharing in mechanical design, *Proceedings if AAAI-88*, pp. 343–346.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. & Tomimaya, T. (1996). Supporting conceptual design based on the Function-Behaviour-State modeler, *AIEDAM* **10**(4): 275–288.

Umeda, Y., Tomiyama, T. & Yoshikawa, H. (1990). Function, behaviour, and structure, *in* J. Gero (ed.), *Applications of Artificial Intelligence in Engineering V vol. 1*, Springer-Verlag, Berlin, pp. 177–193.