

NISTIR 7289

The Role of Standards in Product Lifecycle Management Support

**Eswaran Subrahmanian
Sudarsan Rachuri
Abdelaziz Bouras
Steven J. Fennes
Sebti Foufou
Ram D. Sriram**

NIST

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 7289

The Role of Standards in Product Lifecycle Management Support

**Eswaran Subrahmanian
Sudarsan Rachuri
Abdelaziz Bouras
Steven J. Fenves
Sebti Foufou
Ram D. Sriram**

January 2006



U.S. DEPARTMENT OF COMMERCE
Carlos M. Gutierrez, Secretary
TECHNOLOGY ADMINISTRATION

Michelle O'Neill, Acting Under Secretary of Commerce for Technology
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
William Jeffrey, Director

The Role of Standards in Product Lifecycle Management Support

Eswaran Subrahmanian¹, Sudarsan Rachuri², Abdelaziz Bouras³, Steven J. Fenves⁴,
Sebti Foufou⁵, Ram D. Sriram⁶

Manufacturing Systems Integration Division,
National Institute of Standards and Technology
Gaithersburg, MD 20899-8263

Abstract

A major challenge of any product engineering project is to support the creation, exchange, management and archival of information about product, process, people and services across the networked and extended enterprise covering the entire product lifecycle spectrum. An information support system for Product Lifecycle Management (PLM) requires a move from product data exchange to product information and knowledge exchange across different disciplines and domains. PLM support systems will need to have both syntactic and semantic interoperability of computer systems and people through well defined standards.

We begin this paper with a model of communication between two agents and then extend this model to describe the information flows in PLM so as to serve as the basis for understanding the role of standards for PLM support systems. Support of PLM requires a set of complementary and interoperable standards that cover the various aspects of PLM. We identify an initial typology of standards relevant to PLM support. The typology primarily addresses the hierarchy of existing and evolving standards and their usage. The typology identifies a suite of complementary standards supporting the exchange of product, process, operations and supply chain information.

Given the nature of the task of developing and deploying a set of standards for PLM support systems, we argue that open standards with wide participation are the key to their realization.

Keywords: Product Lifecycle Management, product realization, PLM systems, interoperability, data exchange, standards, open standards

¹ Eswaran Subrahmanian is a Research Professor at Carnegie Mellon University, Pittsburgh. He is currently a guest researcher at MSID, NIST, Gaithersburg, MD 20899, Email: sub@cs.cmu.edu.

² Sudarsan Rachuri is with George Washington University, DC. He is currently a guest researcher at MSID, NIST, Gaithersburg, MD 20899, Email: sudarsan@gwu.edu.

³ Abdelaziz Bouras is with Lumiere University, Lyon. He is currently a guest researcher at MSID, NIST, Gaithersburg, MD 20899, Email: abdelaziz.bouras@univ-lyon2.fr.

⁴ Steven J. Fenves is Emeritus University Professor at Carnegie Mellon University, Pittsburgh. He is currently a guest researcher at MSID, NIST, Gaithersburg, MD 20899, Email: sfenves@cme.nist.gov.

⁵ Sebti Foufou is with LE2i Laboratory, University of Burgundy, B.P. 47870, Dijon France. He is currently a guest researcher at MSID, NIST, Gaithersburg, MD 20899, Email: sfoufou@u-bourgogne.fr.

⁶ Ram Sriram is leading the Design and Process Group at NIST, Email: sriram@cme.nist.gov.

1 Introduction

Product lifecycle management (PLM) as a management paradigm is a strategic approach to creating and managing a company's product-related intellectual capital, from the product's initial conception to the product's retirement. As an information technology (IT) undertaking, PLM support entails modeling, capturing, exchanging and using information in all PLM decision-making processes.

PLM has come to signify what some call the 21st century paradigm for product development. The management of a product from inception to disposal is the strategic initiative that defines 21st century product development. This concept is gaining acceptance primarily because of the emergence of the networked firm and the networked economy, in contrast to the market- or hierarchy-based organizations that used a transactions-cost model as the cornerstone for the choice of organizational structure [1].

PLM entails the management of product design, manufacturing and service knowledge that goes beyond the interaction of suppliers with the system integrator. PLM reaches into the sales, customer service and product disposal activities that participate in the larger network. Without a comprehensive information base capable of providing the information required by the different stakeholders in the product's entire lifecycle, overall efficiencies that can in principle be achieved in the network cannot be realized.

The concept of the lifecycle of a product is illustrated through the metaphor of epicycles in Figure 1. The nodes on the periphery stand for the major stages in the product's lifecycle: conceptualization; organization of resources; design; analysis, simulation and verification; manufacturing and distribution; in-use performance and customer feedback; and eventually disposal. The wide clockwise links on the periphery stand for the "normal" information flows from stage to stage, while the links on the inside stand for the feedback information that creates the various epicycles or sub-loops of reappraisal, redesign and product improvement or evolution. The feedback loops shown in the figure are only suggestive to convey the idea of interdependence among the stages and are not comprehensive.

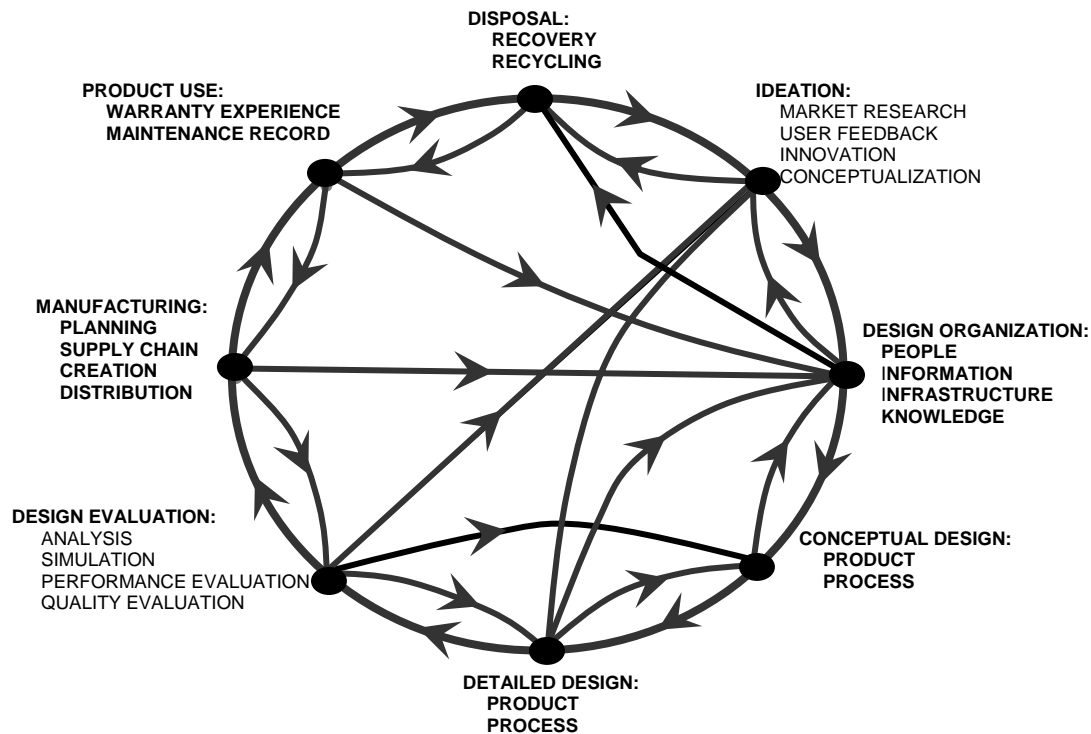


Figure 1: Epicycles in product lifecycle development⁷.

Some of the feedback loops are unavoidable consequences of product complexity, the bounded rationality of the participants, and the rapid changes in technology. How well these feedback loops function is critical to the learning process associated with the management of information on a product's lifecycle. Breakdowns in information transfer causing additional feedback loops may affect both the efficiency of the operation and the quality of the product. The conceptualization of product development presented in Figure 1 provides credence to the analogy of today's product lifecycle management systems to a control system [3]. This is especially true when the above product development lifecycle is seen from the point of view of the globalized market, characterized by changing customer preferences and by product variety, manifesting themselves in versions and series of the same product. The environment in which the product is designed and produced is constantly changing, requiring timely identification and communication of failures, anomalies, changes in technology and other important influences. For such an adaptable organization to function, an information infrastructure that supports well-defined information exchange processes among the participants is critical.

A study of knowledge and information management in a large distributed R&D organization has shown that local and global knowledge in such organizations is seldom integrated [4]. The recognition of the need for integrated information systems has resulted in proposals for maturity models for interoperability to assess and evaluate an organization's ability to provide an interoperable environment [5], [6].

⁷ This figure is adapted from [2].

The report is organized as follows. In Section 2 we present models of communication and of PLM support. We extend this model to the PLM context and describe the current state of support for PLM and the needs for information exchange for supporting PLM. In Section 3, we present a typology of standards. The typology primarily addresses the hierarchy of the standards and is divided into three levels. In Section 4, we present the coverage of current standards and their convergence. In Section 5, we present the challenges in developing information standards for PLM support. Finally, we present our conclusions in Section 6.

2 Models of Communication and PLM Support

To understand the role of standards for PLM support systems, we first postulate a model of information exchange between two agents (human or computer). We will use this model to make the case that supporting PLM is akin to supporting a composition of information exchanges across time, space and multiple disciplines.

2.1 A model of communication between agents

Communication between two agents requires exchanges that convey the content of the information through a language. A model of communication proposed by Flower et al. accommodates the semantics of the exchange [7]. In this model, the exchange between receiver and sender is dependent on the understanding of the mental model of the receiver by the sender, who has to transform his/her mental model to that of the receiver. Both mental models are contextualized by awareness, familiarity and other personal experiences. The objective of the sender is to ensure that she/he communicate to the mental model of the receiver. When the mental models of the receiver and the sender are matched, what is communicated takes on a standardized form of exchange. This form of standardized exchange behavior within a specified set of conventions is called a protocol. The language of a protocol has a form (syntax), function (scope) and the ability to convey as unambiguously as possible an interpretation (semantics) when transferred from one participant to the other.

We term the ability of a language to convey a precise semantics the expressiveness of that language, more specifically the processible expressiveness as used by Webster to mean the computability of the language [8]. Expressiveness of a language is not related to the level of abstraction/detail it uses in describing the domain of interest.

Highly expressive languages are best suited for use within a well demarcated domain. Mathematics has served this purpose in many disciplines. Mathematics is a means of expressing the physical world with a certain amount of precision and parsimony [9]. While mathematics as a meta-language has transcended disciplines, mathematical forms used and interpreted in a domain adhere to the disciplinary vocabulary integral to the linguistic world of discourse (also known as common ground or domain of discourse) of the domain.

Similarly, the visual language of geometry has allowed for the exchange of information from designer to fabricator. Visual language has its own protocols, vocabulary and mental models and has changed from the traditional drawing on paper to the use of computer-based drawing/drafting [10]. In software, the Unified Modeling Language (UML) [11] resulted from a consolidation of languages dealing with efforts to manage software development [12]. Each language has symbols, rules, conventions and a vocabulary that attempt to ensure effective exchange.

Figure 2 illustrates the exchange of the content in a given domain of discourse between the producer and consumer of information and highlights the role of their mental models. The figure incorporates the Flower et al. model along with the choice of languages with a given processible expressiveness to represent content in a domain of discourse.

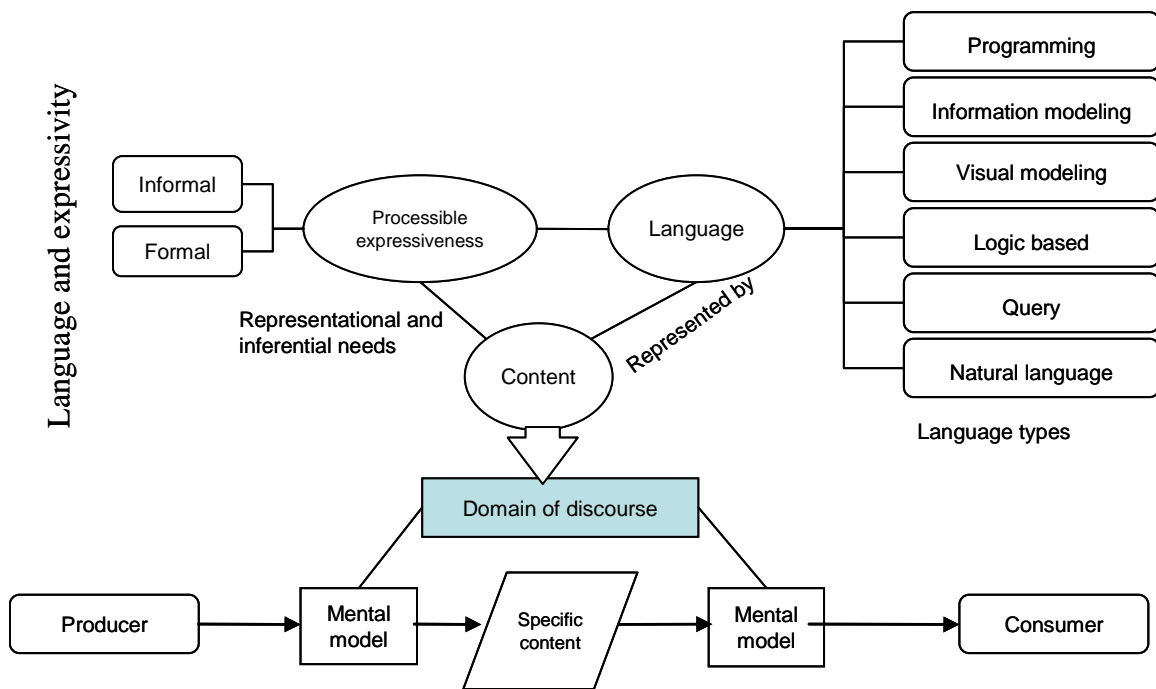


Figure 2: Language, Processible Expressiveness and Content

Exchanges between two agents result in the creation and use of a common linguistic world with multiple symbolic languages that serve as a means for efficient exchange of content [13; 14]. When the linguistic worlds are not the same, the possibilities of misinterpretation and consequently the actions implied by the language (behavior) are mismatched.

In describing the role of protocols in computing, Galloway takes the approach that it is a language with a set of conventions that governs the set of possible interpretations (behaviors) within a heterogeneous environment. In this sense a language of exchange is a technique for achieving voluntary regulation within an environment with many

contingencies [15]. Given the nature of communication in a networked world, protocols (specialized and standardized languages) are means for distributed management that allow for control to exist within a heterogeneous environment [15].

The design of a protocol in the context of information exchange is dictated by:

1. **Content**: the information to be communicated. Content includes the model of information in the domain and the instances in the domain and explicates the relationship between the message and the behavior it intends to elicit from the recipient. For example, an ontology of a domain uses a meta-model to create the domain specific linguistic world of discourse, leading to the ability to describe the information exchanged in a discourse.
2. **Processible Expressiveness**: the degree to which a language mechanism supports machine understanding or semantic interpretation. Natural language has very low expressiveness; by contrast, first order logic and other higher order logic represent high levels of expressiveness.
3. **Language**: the symbols, conventions and rules for encoding content with known expressiveness.

Expressiveness is closely connected to the scope of the content that can be expressed and to the precision associated with that content. The debate on expressiveness⁸ as measured by the level of conformance to first order logic is illustrative in the development of semantic-web-based languages such as Web Ontology language (OWL) [17] and Resource Description Framework (RDF) [18]. A good example is the standard for dimensioning and tolerancing for manufacturing [19; 20], which uses English, mathematics and geometry to explain and codify the standard. The standard represents the language of a theory of dimensioning, which serves as the shared mental model in the interpretation of the standard [21].

We emphasize that there may not be a single representation that can capture all the structure and conceptual abstractions of the content in question. Often, compromises need to be made in the expressiveness and the level of formalism when deciding on the languages to be used in information exchange and on their standards.

The language chosen or designed for the task is not static and may evolve with technology. For example, in geometry the move from traditional drawing to CAD evolved in terms of the languages and mental models required in the use and exchange process [10]. HyperText Markup Language (HTML) has evolved from its early days, and so on. All these languages evolve to incorporate new knowledge and needs as these are identified and codified. Given the social nature of knowledge generation and protocol creation, as Galloway writes, “By design, protocols such as the internet protocols (*product information exchange languages*)⁹ cannot be centralized” [15]. Here the need

⁸ The idea of expressiveness has been debated in the knowledge representation community under the two terms of “representational adequacy” and “inferential adequacy.” [16].

⁹ Italicized text added.

for agreement across a large number of users is required. We will address the issue of how protocols (standards) are created in a later section.

2.2 Context of PLM support: A complex information management and exchange system

PLM support is a multi-dimensional information exchange undertaking, requiring that a variety of contents be exchanged within and across a number of disciplines and functions over time and space. To apply the model presented in the last section to the world of PLM support, it is necessary to address exchanges among multiple stakeholders who live in multiple non-overlapping and overlapping linguistic worlds.

The intersection of the multiple linguistic worlds of the stakeholders participating in the lifecycle of a product creates a new product-centric linguistic world in turn. This new linguistic world creates maps among the disciplinary linguistic worlds. This product-centric linguistic world evolves along with the product and the process development. In the traditional serialized model of mass production engineering this was not the case. It was assumed that the interfaces between the different phases of product development were relatively static; that there was no continual improvement in the content of the linguistic worlds for exchange, and that the evolution of linguistic structures between interfaces was designed in the hierarchical control-based structuring of the organization [22].

The advent of concurrent engineering brought about the recognition of the importance of dynamic information exchange across functional roles so as to ensure the continuing validity of the shared linguistic worlds the participants use in their exchanges [23]. This dynamic exchange points to the need for: (a) the systematic characterization of the different linguistic worlds that participate in the exchanges that take place throughout the product lifecycle process; and (b) the ability to create and use context-specific linguistic worlds to support the collective task of product lifecycle management across organizational boundaries, functions and disciplines. The need for codifying these linguistic worlds across the lifecycle has become critical due to two factors at work: (a) globalization has spread the need for mass customization of products and services to different markets; and (b) global environmental concerns have led to the heightened consideration of the servicing and disposal of products as integral parts of the product development lifecycle.

The implication of the above is that the content description in specific information exchange contexts requires the customization of languages of different expressiveness to make explicit the scope of the exchanges. It is clear that supporting PLM is not a well defined task as all of the information needed will not be known *a priori*. The needs will evolve and get incorporated into the system based on the usefulness of the information for specific tasks. For example, the representation of the geometry of products has evolved from simple drawings to analytical geometry and to more computable representations such as boundary representation (B-rep), solid modeling and features.

Here the languages evolved to achieve expressiveness for the description for particular aspects of the product development process. In the context of supporting PLM, it is inevitable that new languages for different purposes will arise, requiring different levels of expressiveness for particular contexts and driven by technological needs and possibilities.

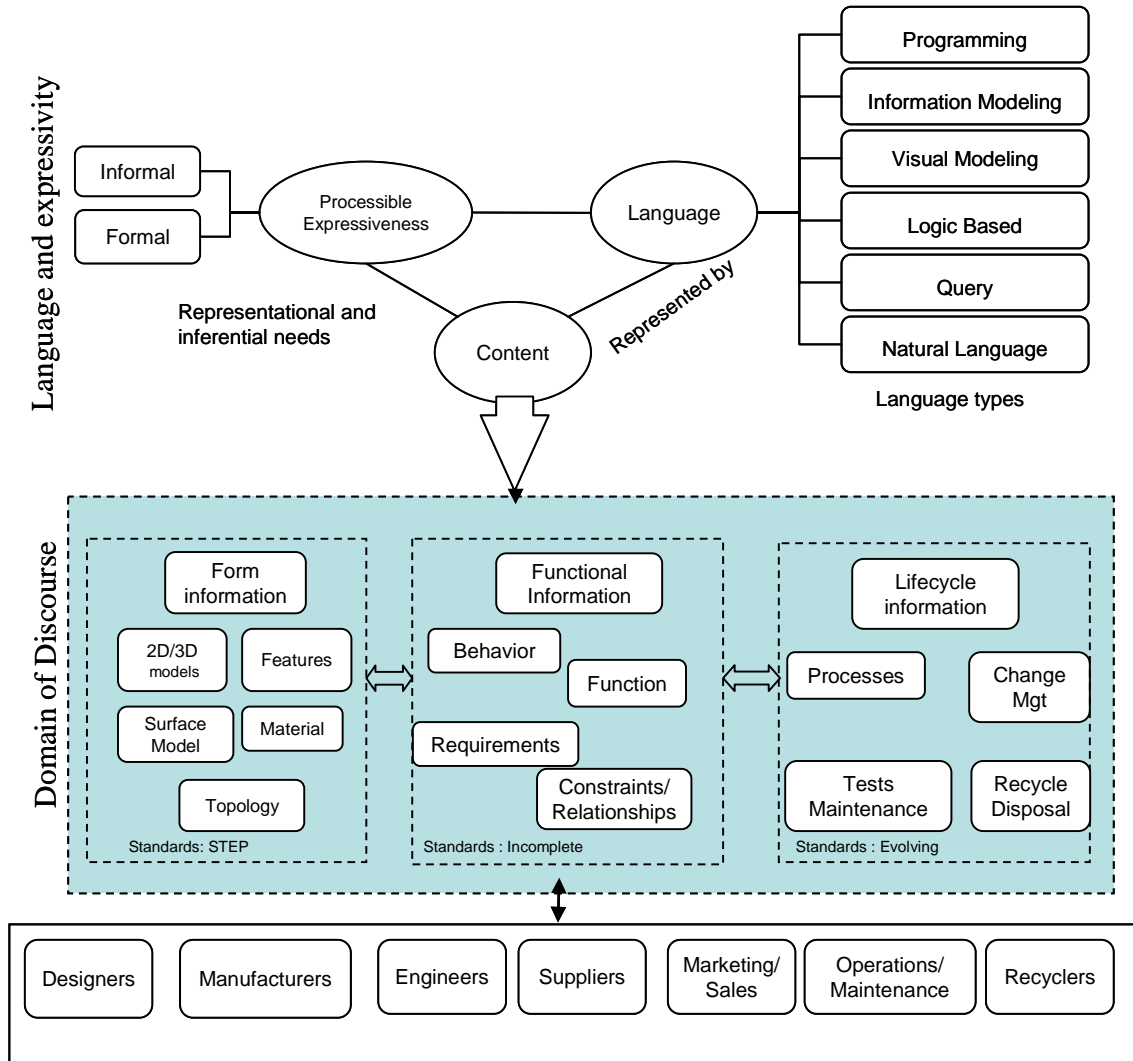


Figure 3: Communication and information exchange for the PLM

Figure 3 illustrates how the content of product information is used by various stakeholders in a product's lifecycle. The information content is increasing in two dimensions: (a) in complexity because it is being authored and acted on by an increasing number of stakeholders; and (b) in scope from form to function and eventually covering all aspects of the product during its lifecycle. The increased scope and complexity of the content will require languages at various levels of processible expressiveness. We emphasize the point that there is no single representation that can capture all the

structural and conceptual abstractions of the content in question for all venues of information exchange.

Support for our model can be seen in the work on developing virtual product models, as well as in the development of specific standards in design and manufacture of products. As an example, the MOKA (Methodology and tools Oriented to Knowledge-based engineering Applications) Project [24], addressing virtual product information, explicitly identifies the need to choose languages with the expressiveness required for a specific content. The MOKA project itself uses UML as the base language but adds other features to create MOKA Modelling Language (MML) [24] to suit the level of expressiveness needed in engineering design.

The model of communication for PLM presented here will be used for defining a typology of standards that form the basis for effective information exchange and interoperability of various systems within PLM. In the next section we describe the current state of support for PLM.

2.3 State of the art of PLM support

The IT industry that supplies PLM support systems is currently vertically integrated. Vertically integrated support systems do not provide for opportunity of full diffusion of new innovations across the entire community of users. A study of PLM support provided by a representative set of major software vendors shows that the availability of support tools is partial and incomplete [25]. The study covers several areas of PLM. Some vendors cover several areas, while there are areas that are poorly covered or not covered at all by any vendor. Relying on a single vendor to cover all areas of PLM support would not provide the kind of innovation needed by PLM customers. The study reveals that there is a lack of interoperability across tools and that there are barriers to entry for software developers that could provide a plug and play approach to PLM support. Currently only a few IT companies with vertically integrated tool sets are able to provide facilities that are even partially integrated.

In an extended enterprise context, PLM support needs to connect the product design and analysis processes to the supply chain processes, including: product data management (PDM), component supplier management (CSM), enterprise resource planning (ERP), manufacturing execution systems (MES), customer relationship management (CRM), supply and planning management (SPM), and others that will undoubtedly follow. Most of the activities of the systems linked to the supply chain processes are centered on bill of material (BOM) management after release through obsolescence, and several stages of a BOM may exist: as-planned, as-ordered, as-built, as-shipped, as-maintained, to cite a few. Maintaining the accuracy of these stages requires extensive transaction processing for recording substitutions, changes, and the like. The benefits of PLM will be realized only when these disparate systems are horizontally integrated. Furthermore, as of today, there is no commonly agreed-upon linkage between the spatially oriented product representations used in design and the BOM based representations used in downstream processes.

3 Towards a typology of PLM support standards

As stated before, PLM support requires a move from product data exchange to product information exchange, across different disciplines and domains.

Several typologies of standards relevant to PLM support have been proposed:

- (1) according to the *stages of the product lifecycle* the standards address: (a) product development standards; (b) product production standards; (c) product use standards; and (d) product identification standards linked to product lifecycle traceability [26].
- (2) according to the *scope* of the standards: (a) PLM commercial best practices and specifications; (b) standards related to specific applications; (c) standardized data models to represent product data; and (d) domain standards [27].
- (3) according to the *origin* of the standards: (a) open standards; (b) industry standards; and (c) de facto standards, that are widely accepted and used and results generally from widespread consensus [28].
- (4) according to the *development process* of the standards: (a) de facto standards; (b) regulatory standards created by regulatory agencies to ensure uniformity in processes that are not driven by market forces; and (c) consensus standards, developed or used by voluntary consensus standard development organizations (SDO) [29].
- (5) according to the *intent* of the standards: (a) measure or metric standards; (b) process-oriented or prescriptive standards, which provide tests in a consistent and repeatable way; (c) performance-based standards, where process is not specified but the ultimate performance is; and (d) interoperability standards, where sometimes process and performance are not explicitly defined, but a fixed format is always specified [29].

Using the model of communication presented in Section 2.1 and extended to PLM support in Section 2.2, we present a hierarchical typology of standards. The typology primarily reflects the *content* to be communicated and implies the appropriate expressiveness and language choices for each type. Within each type, individual standards may be classified as to origin, intent, development process and, to some extent, scope by the typologies listed above. The typology is based on the one initially presented in [30]. For illustration, we give some examples of standards in each type.

Type Zero: Standards for implementation languages

These standardized languages include programming, scripting, assembly level and other computable languages used to implement the Type One, Type Two, and Type Three

standards presented below. Examples include, Basic, FORTRAN, C, C++, Java, C#, Prolog, Perl, Tcl/Tk, OpenGL.

Type One: Information modeling standards

Semantically rich modeling language standards, based on different forms of logic, include the Knowledge Interchange Format (KIF), OWL and RDF that support reasoning over the information representing a content domain [17; 18; 31]. OWL includes the RDF/XML interchange syntax and has three sub-languages of different expressiveness and complexity (OWL Lite, OWL DL, OWL Full). All of these efforts are directed towards building formal ontologies that are expected to aid semantic interoperability.

EXPRESS [32] and UML (Unified Modeling Language) are two examples of information modeling languages. EXPRESS is used in the STEP-based systems [33], while UML is based on the object-oriented methodology [12]. UML is primarily intended for specifying, visualizing, constructing, and documenting components of software systems as well as for business modeling and other non-software systems. The expressive power of EXPRESS is comparable to the combination of UML and the Object Constraint Language (OCL), a formal language used to describe expressions in UML models [34].

XML Schema is becoming popular for expressing the structure and typing constraints for data embedded in XML documents. XML Schema offers a higher level of expressiveness than the earlier DTD (Document Type Definition) descriptions.

A NIST focus area is the standardization of the representation of manufacturing processes, called the Process Specification Language (PSL) [35]. PSL uses first order logic and Ontology Web Language (OWL)-like representations [17].

Type Two: Content standards - domains of discourse

Content standards pertain to information models specifically defined for particular domains using a generic information modeling language (Type One) or an extended one (for example, UML with its extension, SysML). The Systems Engineering Modeling Language (SysML) is directed towards the specific domain of systems engineering. SysML is derived from the basic UML to cover the requirements, structure, behavior, parametrics, and the relation of structure to behavior (allocation) [36]. Content standards might use general Type Zero languages for implementation.

Content standards subdivide into several categories based on the specialization of the content addressed. The principal categories are briefly described below.

Product information modeling and exchange standards. Standards such as ISO 10303, informally known as the STandard for Exchange of Product model data (STEP) deal with product structure and geometry and part-related information [37]. STEP uses the EXPRESS information modeling language to define a generic product model. In the modular approach of STEP, information models form modules and integrated resources

(IRs), from which specific content standards (application protocols or APs) are developed (for example, AP 214 [38]). These specific content standards can use specific catalogs. As another example, UML and its extension, SysML, can be used to define specific system (say, control system) standards. This hierarchy is illustrated in the following figure (The arrows represent “uses”).

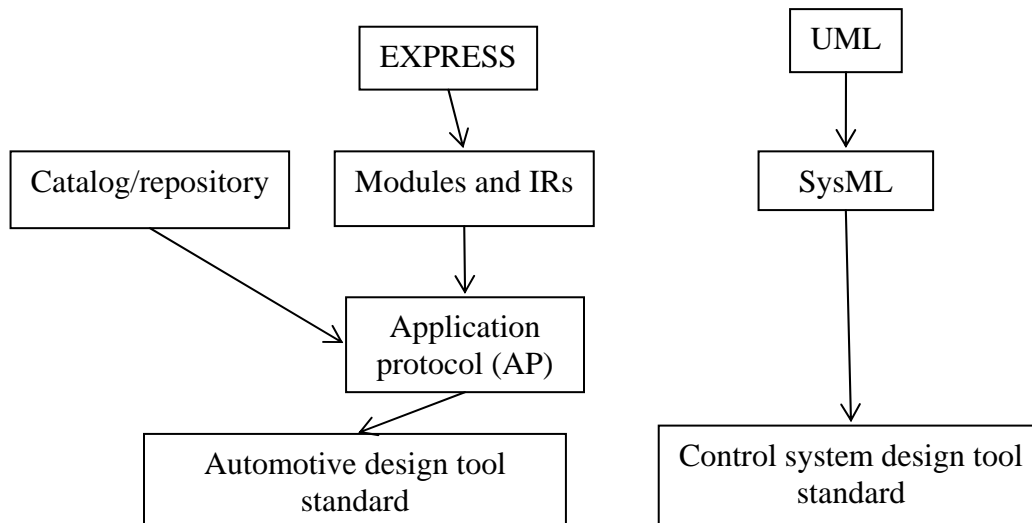


Figure 4: Examples of Content Standards

STEP AP239, called PLCS (Product Lifecycle Support), is dedicated to product support and is based on an extension to the STEP PDM Schema capability [39]. PLCS provides mechanisms to maintain the information needed to support complex assets (such as ships, aircraft or engines). PLCS builds upon the functionality defined by other standards relevant to product support (these include AECMA S1000D, AECMA 2000M, United States Military Specification 1388, United Kingdom Defense Standard 00-60).

Information exchange standards. XML-based protocols are being developed to exchange content among various stakeholders. Examples of information exchange standards are the electronic data interchange (EDI) [40] and the simple object access protocol (SOAP) [41] standards. Among the specialized versions of these exchange standards are: (a) STEPml [42], a library of XML specifications based on the content models from the STEP standard; (b) the Product Data Markup Language (PDML) [43] being developed as part of the Product Data Interoperability (PDI) project [44]; (c) PLM XML [45], a set of XML schemas serving as a transport protocol; and (d) BPML (Business Process Modeling Language) [46], a meta-language for the modeling of business processes.

Product visualization standards. The U3D graphics standard is a simple format for interactive viewing and sharing of 3D data [28]. X3D is an XML-enabled 3D standard to enable real-time communication of 3D data [47]. JT is a CAD neutral data format for product visualization, collaboration and data sharing [48]. JT Open is a library of Java classes supporting the client/server and Internet programming models. JT2Go is a JT

format viewer. OpenGL (Open Graphics Library) is a low-level graphics library for 3D data visualization [49]. OpenML (Open Media Library) is a programming environment that supports the creation and playback of digital, audio, video and graphics [50].

E-business and value chain support standards. Many extensions to XML have been developed to describe the business activities associated with all phases of satisfying customer demand, such as electronic business XML (ebXML) [51] to replace traditional EDI standards and commerce XML (cXML) [52]. Related extensions include the Chemical markup language (CML) [53] and the Materials Markup Language (MatML) [54].

RosettaNet is an example of standardizing eBusiness interfaces to align processes between supply chain partners [42]. The IPC-2570 standard series is a complementary effort to foster application integration through encoding scheme that enables a total product definition to be described at a level appropriate to facilitate supply chain interactions [55].

Another interesting initiative for supply chains is the SCOR reference model, which defines standard metrics to measure process performance and management practices [56]. A new model called VCOR aims at providing a unified and universal approach to organizational analysis and help in consolidating enterprise processes [57]. VCOR is intended to be a de facto standard Value Chain Operational Reference Model [58].

Security standards. The earlier types of standards focus on what is to be represented, how it is to be represented and how it is to be exchanged. What is missing is how much of the information needs to be exchanged and with whom. This is important from the point of view of information overload, intellectual property rights, and security. DRM (Digital Rights Management) refers to technologies that have been specifically developed for managing digital rights. As an example, XrML (eXtensible rights Markup Language) provides a universal method for specifying rights and issuing conditions associated with the use and protection of content [59]. Various organizations such as the NIST Information Technology Laboratory and the World Intellectual Property Organization (WIPO) are establishing international standards in this area [60].

Type 3: Architectural frameworks standards

To achieve interoperability between the standards within the PLM context, it is imperative that the different types of standards described in this section be reconciled and made convergent. In integrating these types of standards, it is necessary to take into consideration the architectural frameworks for creating integrated support systems. A number of integration framework standards have been proposed, such as the Zachman Framework [61], the Department of Defense Architecture Framework (DoDAF) [62] and the Federal Enterprise Architecture Framework (FEAF) [63]. These frameworks do not yet provide the full spectrum of viewpoints needed to address the overall interoperability concerns. Another interesting framework is the ISO RM-ODP Reference Model for Open Distributed Processing [64]. This model has been used as a framework for CORBA-based

distributed applications management, and defines five architectural viewpoints that address a wide range of interoperability concerns from policies and procedures to engineering solutions: the enterprise, information, computational, engineering and technology viewpoints.

4 Convergence of PLM support standards

The current state of the underlying linguistic structures populating PLM support is fragmented and incomplete in coverage. Beyond incompleteness, the incompatible linguistic structures have evolved bottom up (based on localized needs and vendor-centric definitions) in developing support for particular aspects of the product lifecycle. This has led to the lack of interoperability, an issue that has become the Achilles heel of integration for PLM support.

The importance of interoperability across the phases and functions in PLM among the multiplicity of languages dealing with the varied contents comprising the complete product description has been recognized by a number of institutions including NIST, the US Department of Defense (DOD), the European Ministries of Defense and, more recently, by the vendor and end-user communities [65; 66]. While there has been articulation of the need, the issue has not been fully addressed due to the divergence of interests on how interoperability should be achieved. The challenge is to create standards and protocols that allow legacy systems as well future technological innovations to interoperate seamlessly.

Today's standards, particularly in the area of CAD, have produced direct improvement in productivity, especially in the manufacturing arena, by reducing transaction costs and even more so by increasing the richness of interactions between supplier and customer [67]; [68]; [69]. The real cost of the lack of interoperability is difficult to measure and is often buried in day to day operations of individuals needing the information or needing to transmit the information.

In the context of PLM the need for standards at different levels of expressiveness becomes critical in supporting the processes of information exchange. The achievement of the PLM objectives requires near decomposability of systems, as was observed by Herbert Simon in his essay, "The Architecture of Complexity" [70]. We claim that the complexity of the information exchange issue and the diversity of participants and their perspectives make it highly improbable that a single disciplinary perspective can accomplish the task of supporting PLM. For these reasons we believe that much of the stable part of the information base supporting PLM should be developed within the framework of open standards. The recent document called Roadmap for Open ICT Ecosystem [71]¹⁰, defines open standards as standards that are created by standards-setting organizations including consortia such as the Internet Engineering Task Force (IETF), the World Wide Web Consortium (W3C), and the Organization for the

¹⁰ released by the Berkman Center for Internet and Society at the Harvard Law School

Advancement of Structured Information Standards (OASIS) and formal standard development organizations (SDOs) such as the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). Krechmer provides more detailed definitions of open standards, open source, and open architecture; and defines ten rights that will enable open standards [66].

The incompatibilities and gaps that exist among current standards can arise at different levels. These can be explained using the typology defined in Section 3. The incompatibility may arise at the level of implementation by the choice of Type Zero standards. The choice of Type One Standards and the underlying representation formalism is another major source of incompatibility. The choice of Type Three standards may lead to different architectural frameworks which might introduce more incompatibility. The gaps among standards can be understood by differences in the domain of discourse and in the scope of standards along the PLM spectrum.

It is clear from Figure 5 that there is no standard that provides full coverage of the PLM support spectrum. Note also that some standards, for example SysML and PSL, cover some aspects of PLM with notable discontinuity in their scope.

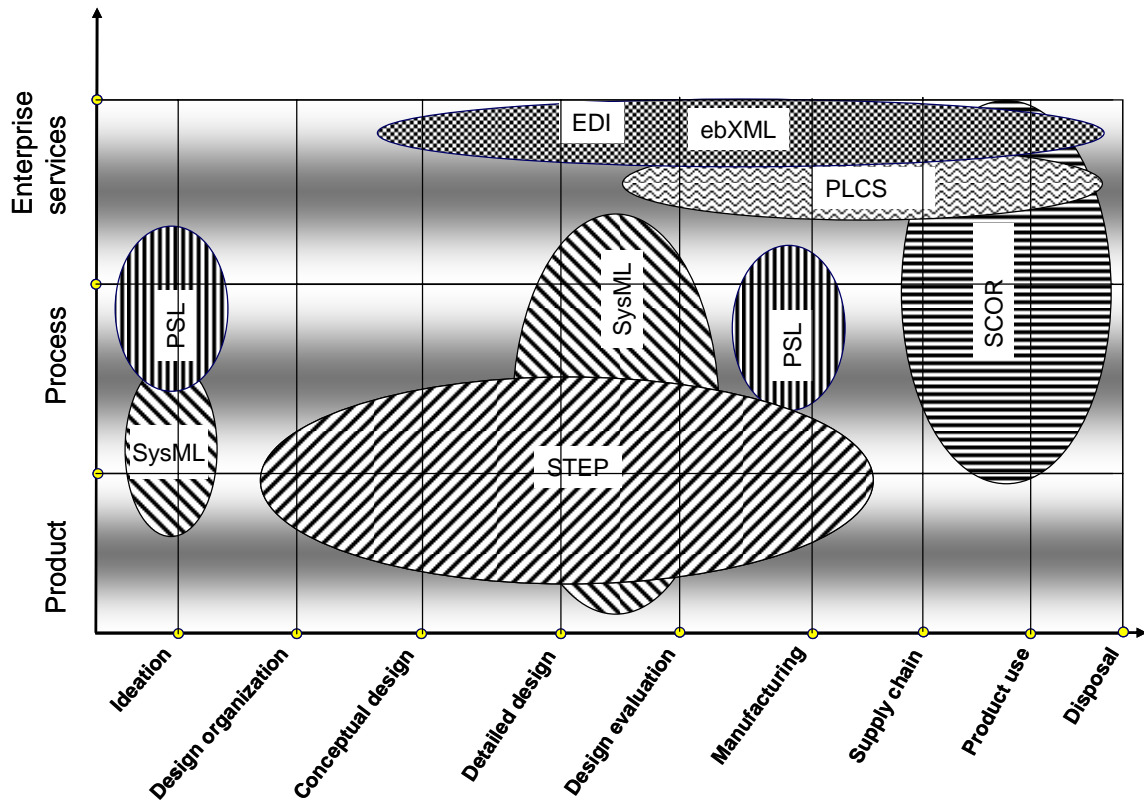


Figure 5: An example of current standards and their coverage

5 Challenges in standards development for PLM

As the type and scope of the standards needed for PLM support becomes clearer, there is increased interest in the models that have emerged in the distributed internet world for standards development and software production. Specifically, it is increasingly clear that a combination of open source models and TCP/IP-like open standards approaches is needed [15; 65]. The Internet Engineering Task Force (IETF) open standards approach ensured that the features incorporated were demonstrated to be useful not just for one organization but for all who are affected and interested enough to participate in the process. Membership in these standards development organizations is voluntary; the reputation of the contributor through prior reliable and useful contributions and active participation leads to influence and power. Open source models that have evolved with the rise of the internet have addressed the large scale distributed design of complex products. The primary reason for success of the open source model has been its ability to scale that transcends “Brooks’ Law” [72]. According to Brooks’ law, the propensity for errors increases geometrically when resources are added linearly. In the case of open source, this constraint on the use of resources is resolved through the voluntary nature of participation and task selection [65]. Further, the availability of the complete record of the decision-making process and of the results of testing and use allows for the design-use-maintenance cycle to be rather quick in turnaround. The major success of open source comes from the recognition of the scale and diversity of skills through modular design; minimizing costs of bad local decisions; and the ability to mobilize people of diverse skills [65].

The development of standards for PLM support requires many people with different skills and expertise to participate. These people cannot be assembled in one place -- virtual or real -- to create a centralized top-down set of standards. Standards are best created in an open process and within a voluntary participatory model. Earlier experiences with standards development in the distributed, connected world point to open standards and open source models as an alternative model of consensus building and production [66]. Who should be responsible for creating these communities is an open question. Many existing standards have been created by individuals or groups of individuals. Some of these efforts also involve regional economic development zones.

There are enormous benefits due to network effects that can be realized socially by open approaches; this will allow university, industry (small, medium, and large), and laboratories to draw out of the same well, minimizing incompatibilities and increasing the ability to create a plug-and-play system. This approach will foster innovation at the modular level without requiring a drastic redesign of the overall system. Repeated drastic design changes for both standards and software lead to increased costs in the creation, training, maintenance and archiving of the information and knowledge created in the life of a product. The open source and open standards models seem to address these issues in the best manner given the scale and scope, and provides us with alternative models for standards development.

Maintenance and use require conformance to standards and certification of the same. Institutions such as NIST can play a major role in being the honest broker and archivist of

the process of standards creation, maintenance, and evolution. To encourage this process, the Manufacturing Interoperability Program of the Manufacturing Engineering Laboratory at NIST is working on achieving convergence among standards. Beyond the convergence of the types of standards referred to in this paper, standards for other aspects such as traceability, validation, verification and other audit and archival functions will have to be considered in the support system for PLM [73]. A good example is the Open Archival Information System (OAIS) reference model which facilitates a much wider understanding of what is required to preserve and access information for the long term [74].

The current disparate standards with differing assumptions and purposes are not easily reconciled; neither can they be resolved by any single entity. The extraction of positive network externalities in the networked manufacturing economy can only be achieved by the free flow of ideas and the exchange of knowledge in a public or semi-public space to create new innovations in the new knowledge economy [75]. To exploit the efforts of a large number of researchers, practitioners, users and students to continuously integrate their work into the larger vision of full PLM support, there is no choice but to develop a pragmatic mechanism for supporting the development of standards in an open environment where the participation of all parties concerned will become critical.

The convergence of standards can only take place in an open environment given the complexity of the task ahead. This realization can be seen in the publications of information technology vendors such as IBM, end users such as DOD and engineering consultants making a case for open standards for the information base required to support the underlying IT infrastructure to accommodate legacy and changing technologies [5; 76; 77].

6 Conclusions

In this paper, we start by describing a model of communication between agents and extend the model for PLM support among various stakeholders. This model of communication is influenced by: (1) the choice of a particular language with a required level of processible expressiveness; (2) the choice of representational formalism and an appropriate information modeling language; (3) structuring of the domain knowledge using the above choices; and (4) architectural frameworks as a medium of communication. Using this model we defined a typology of standards.

We also provided an overview of the current state of IT standards for PLM support. The incompatibility and gaps that exists among standards could be due to various reasons that can be explained using the typology defined in Section 3.

We make a case that the nature of the task needs open standards with wide participation. It is always said that markets are the best determinants of standards and that, if history is a guide, even market forces lead to co-operation among competitors to work towards open standards. Publications in the popular press lead us to believe that IT vendors are ready for a move towards open standards. As is evidenced in the evolution of the Internet, the development of standards can only flourish through an open but centralized

participatory model of development. A single monolithic/monopolistic structure is not adaptable and flexible to address the variety and complexity inherent to this issue.

Standards cannot be advocated by just a select few in the organization. While advocating open standards, Srinivasan claims that the natural tendency of PLM software vendors is to make their product the de facto standard, which allows them to control the content and the price of their products; and that customers, on the other hand, push for open standards, which allow them freedom of choice and flexibility in running their businesses [78]. He states that this tension is not likely to disappear any time soon and must be treated as a part of the standardization process. How you overcome this tension by accommodating the needs for intellectual property rights for sustainable business and the needs of end users are questions that are still being answered in the open source community.

7 Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipments, instruments, or materials are identified in this report to facilitate better understanding. Such identification does not imply recommendations or endorsement by the National Institute of Standards and Technology, nor does it imply the materials or equipment identified are necessarily the best available for the purpose.[65]

References

1. Powell, W. W., Neither market nor hierarchy: network forms of organization. Staw, B. M and Cummings, L. L. 295-336, 1990. Greenwich, CT, JAI Press. *Research in Organizational Behavior*.
2. Lederberg, J., Introduction: Reflections on Scientific Biography. Lederberg, J. [3], xvii-xxiv, 1990. Annual Reviews Inc, CA, USA. The excitement and fascination of science: Reflections by eminent scientists- Part 1.
3. Westerberg, A., Subrahmanian, E., Konda, S. L., and Reich, Y., Designing the Design Process. Special Issue of Computers and Chemical Engineering 21, s1-s9, 1997.
4. Davis, J., Subrahmanian, E., and Westerberg, A., "The Global and the Local in Knowledge Management," *Journal of Knowledge Management, Special Issue on Technology*, Vol. 9, No. 1, 2005, pp. 101-112.
5. C4ISR Architecture Working Group, "Levels of Information Systems Interoperability (LISI): C4ISR Architecture Working Group Final Report," 1998.
6. Ingward, B., Falk, M., Edwin, S., and Erik, M., Project iViP: Integrated Virtual Product Creation - Final Report. Frank-Lothar, K, Trac, T, and Irich, A. 2002. Organization responsible for production and manufacturing technologies (PFT), Research Centre Karlsruhe GmbH.
7. Flower, L., Stein, V., Ackerman, J., Kantz, M. J., McCormick, K., and Peck, W. C., *Reading-to-Write Exploring a Cognitive and Social Process*, Oxford University Press, 1990.

8. Dallas E. Webster, "Mapping the design information representation terrain," *IEEE Computer*, Vol. 21, No. 12, 1998, pp. 8-23.
9. Gregory, B., *Inventing Reality: Physics as Language*, John Wiley & Sons, New York, 1988.
10. Henderson, K., *On Line and on Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*, MIT Press, Cambridge, MA, USA, 1998.
11. OMG, UML 2.0 Superstructure Specification. <http://www.omg.org/cgi-bin/doc?ptc/03-08-02>, 2003.
12. Booch, G., Rumbaugh, J., and Jacobson, I., *The United Modeling Language User Guide*, Addison-Wesley, 1997.
13. Clark, H. H., *Using Language*, Oxford University Press, 1996.
14. Minneman, S. L., and Leifer, L., "Group engineering design practice: The social construction of a technical reality," 1993.
15. Galloway, A. R., *Protocol, how control exists after decentralization*, The MIT Press, 2001.
16. Minsky, M., "Commonsense-based Interfaces," *Communication of ACM*, Vol. 43, No. 8, 2000, pp. 67-73.
17. Web Ontology Language (OWL). <http://www.w3.org/2004/OWL/>, 2005.
18. RDF- Resource Description Framework. www.w3.org/RDF/, 2005.
19. ISO 1101:1983, Technical drawings -- Geometrical tolerancing -- Tolerancing of form, orientation, location and run-out -- Generalities, definitions, symbols, indications on drawings. International Organization for Standardization, Geneva, Switzerland.
20. ASME, *Dimensioning and Tolerancing Y14.5M*, ASME, 1994.
21. Srinivasan, V., "A geometrical product specification language based on a classification of symmetry groups," *Computer-Aided Design*, Vol. 31, No. 11, 1999, pp. 659-668.
22. Waring, S. P., *Taylorism Transformed: Scientific Management Theory Since 1945*, University of North Carolina Press, 1994.
23. Finger, S., Konda, S. L., and Subrahmanian, E., "Concurrent Design Happens at the Interfaces," *AIEDAM*, Vol. 9, 1995, pp. 89-99.
24. MOKA, MOKA: A Framework for structuring and representing engineering knowledge. <http://www.kbe.coventry.ac.uk/moka/miginfo.htm>, 1999.
25. Burkett, M., Kemmeter, J., and O'Marah, K., "Product Lifecycle Management: What's Real Now," *AMR Research*, September 2002, 2002.

26. Terzi, S., Elements of Product Lifecycle Management: Definitions, Open Issues and Reference Models. 2005. Politecnico di Milano, Italy and Henri Poincarre University of Nancy, France.
27. Iyer, R. G., and Gullledge, T., "Product Lifecycle Management for the US Army Weapon Systems Acquisition," Inderscience Publishers, PLM: Emerging solutions and challenges for Global Networked Enterprise, ISBN 0-907776-18-3,2005, pp. 553-564.
28. Lubell, J., Peak, S. S., Srinivasan, V., and Waterbury, S. C., "STEP, XML, AND UML: Complementary Technologies," Vol. CIE-2004-141, Proceedings of DETC 2004: ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference September 28–October 2, 2004, Salt Lake City, Utah USA,2004.
29. Allen, R. H. and Sriram, R. D., "The Role of Standards in Innovation," *Technological Forecasting and Social Change*, Vol. 64, 2000, pp. 171-181.
30. Gruber, T. R., "A translation approach to portable ontologies," *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 199-220.
31. KIF-Knowledge Interchange Format. <http://logic.stanford.edu/kif/kif.html> , 2005.
32. ISO 10303-11: 1994, Industrial automation systems and integration -- Product data representation and exchange -- Part 11. Description methods: the EXPRESS language reference manual.
33. Schenck, D., and Wilson, P. R., *Information modeling: the EXPRESS way*, Oxford University Press, New York, 1994.
34. OMG, UML 2.0 OCL Specification. <http://www.omg.org/cgi-bin/doc?ptc/03-10-14> , 2004.
35. Process Specification language (PSL) . <http://www.nist.gov/psl> , 2005.
36. Bock, C., "Systems Engineering in the Product Lifecycle," *International Journal of Product Development, Special Issue on Product Lifecycle Management*, 2004.
37. Kemmerer, S., "STEP: The Grand Experience, (Editor)," NIST Special Publication 939, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA, 1999.
38. ISO/DIS 10303-214:2000, Product data representation and exchange: Integrated application resource: Application protocol: Core data for automotive mechanical design processes. International Organization for Standardization (ISO), Geneva, Switzerland.
39. The STEP PDM Schema Home Page. http://www.pdm-if.org/pdm_schema/ , 2005.
40. UN/EDIFACT: United Nations standards for Electronic Data Interchange for Administration, Commerce and Transport. <http://www.unece.org/trade/untdid/welcome.htm> , 2005.
41. SOAP Specifications. <http://www.w3.org/TR/soap/> , 2005.
42. STEPml - STEP markup Language. <http://www.stepml.org/> , 2005.

43. Product Data Markup Language (PDML). <http://www.PDML.org/pdmlintro.html> , 2005.
44. Joint Electronic Commerce Program Office (JECPO) re-named as Defense Electronic Business Program Office. <http://www.acq.osd.mil/scst/index.htm> , 2005.
45. PLMXML. www.plmxml.org , 2005.
46. Business Process Modeling Language (BPML). http://www.service-architecture.com/web-services/articles/business_process_modeling_initiative_bpml.org.html , 2005.
47. Web3D: Open Standards for Real-Time 3D Communication. <http://www.web3d.org/> , 2005.
48. Wang, M. Y. and Wang, X., "A level-set based variational method for design and optimization of heterogeneous objects," *Computer-Aided Design*, Vol. 37, No. 3, 2005, pp. 321-337.
49. OpenGL - The Industry Standard for High Performance Graphics. <http://www.opengl.org/> , 2005.
50. OpenML - The Standard for Dynamic Media Authoring. <http://www.khronos.org/openml/> , 2005.
51. ebXML. <http://www.ebxml.org/> , 2005.
52. cXML. <http://www.cxml.org/> , 2005.
53. CML. www.xml-cml.org/ , 2005.
54. Materials Markup Language (MatML). www.matml.org , 2005.
55. Shop floor communications (PDX): IPC 257x. <http://webstds.ipc.org/2571/2571.htm> , 2005.
56. SCOR: Supply Chain Operations Reference-model. <http://www.supply-chain.org/index.www> , 2005.
57. VCOR: Value-Chain Operations Reference. www.value-chain.org , 2005.
58. Roy, U. and Liu, C. R., "Establishment of functional relationships between product components in assembly database," *Computer-Aided Design*, Vol. 20, No. 10, 1988, pp. 570-580.
59. eXtensible rights Markup Language (XrML). www.xrml.org/ , 2005.
60. The World Intellectual Property Organization (WIPO) . <http://www.wipo.org/> , 2005.
61. Sowa, J. F. and Zachman, J. A., "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No. 3, 1992, pp. 590-616.

62. DoD, DoD Architecture Framework Working Group, DoD Architecture Framework Version 1.0, Volume I: Definitions and Guidelines. http://www.defenselink.mil/nii/doc/DoDAF_v1_Volume_I.pdf , 2004.
63. A Practical Guide to Federal Enterprise Architecture. <http://www.cio.gov/archive/bpeaguide.pdf> , 2001.
64. DSTC RM-ODP Information Service. http://www.dstc.edu.au/Research/Projects/ODP/ref_model.html , 2005.
65. Weber, S., *The Success of Open Source*, Harvard University Press, Cambridge, MA, 2004.
66. Krechmer, K., "Open Standards requirements," *The International Journal of IT Standards and Standardization Research*, Vol. 14, No. 1, 2006.
67. Robertson, D. and Allen, T. J., "CAD system use and engineering performance," *IEEE Transactions on Engineering Management*, Vol. 40, No. 3, 1993, pp. 274-282.
68. Liker, J. K., *The Toyota Way: 14 Management Principles from the world's greatest manufacturer*, Mc Graw Hill, 2004.
69. Gregory, T., "Interoperability Cost Analysis of the U.S. Automotive Supply Chain," National Institute of Standards and Technology, Department of Commerce, USA, 99-1 Planning Report, 1999.
70. Simon, H. A., "Architecture of complexity," *Proceedings of the American Philosophical Society*, Vol. 106, 1962, pp. 467-482.
71. The Berkman Center for Internet and Society at Harvard University, "Roadmap for Open ICT Ecosystems," 2005.
72. Brooks, F. P., *The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition*, Addison-Wesley Professional, Boston, MA, USA, 2005.
73. Denno, P. and Thurman, T., "Requirements on Information Technology for Product Lifecycle Management," *International Journal of Product Development - Special Issue on PLM*, Vol. 5, No. 1, 2005.
74. CCSDS, "CCSDS 650.0-B-1: Reference Model for an Open Archival Information System (OAIS). Blue Book. Issue 1. ISO 14721:2003.," 2005.
75. David, P. A. and Foray, D., "Economic Fundamentals of the Knowledge Society, Policy Futures In Education," *An e-Journal : Special Issue: .Education and the Knowledge Economy*, Vol. 1, No. 1, 2003.
76. Mick, R., Defining and Measuring Interoperability. ARC Insights, Insight 2004-43 , 2004.
77. Srinivasan, V., Open Up PLM Systems. Design News , 2004.
78. Srinivasan, V., "Open Standards for Product Lifecycle Management," Inderscience Publishers, in *PLM: Emerging solutions and challenges for Global Networked Enterprise*, ISBN 0-907776-18-3, 2005, pp. 475-484.

