NISTIR 90-4295

A PROPOSED INTEGRATION FRAMEWORK FOR STEP (STANDARD FOR THE EXCHANGE OF PRODUCT MODEL DATA)

William F. Danner

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
National Engineering Laboratory
Center for Building Technology
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 90-4295

A PROPOSED INTEGRATION FRAMEWORK FOR STEP (STANDARD FOR THE EXCHANGE OF PRODUCT MODEL DATA)

William F. Danner

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
National Engineering Laboratory
Center for Building Technology
Gaithersburg, MD 20899

April 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Abstract

This paper presents a proposed integration framework for product data modeling. The framework provides for the representation of functional, programmatic, and physical product data across all phases of a product's life cycle. It provides a single coherent approach to product data modeling for the specification of application views. Most importantly, it creates an open system that encourages the innovative use of information.

The framework has as its major feature an integrated product information model with four conceptual levels. These include generic product description, property description, representation & presentation, and mathematical resources. A generic product data model is the key element of the framework. It is composed of application-independent facts common to all products. The generic model is a distillation from the models currently under consideration by the STEP[1] and PDES[2] projects.[3] The generic product data model meets the requirements of multiple application areas by providing for the interpretation of generic facts in specific contexts. It also provides a logical structure for the integrated product information model which is used by application models to fulfill user requirements.

Keywords: Framework, Information Modeling, Integration Framework, IPO, PDES, Product Data, Product Data Modeling, Product Modeling, Standard for the Exchange of Product Model Data, STEP

---

1 The Standard for the Exchange of Product Model Data (STEP) project of the International Organization for Standardization (ISO) Technical Committee on Industrial Automation Systems (TC 184) Subcommittee on Manufacturing Data and Languages (SC4) Working Group 1 (WG1).

2 The Product Data Exchange under STEP (PDES) project of the Initial Graphics Exchange Specification (IGES)/PDES Organization (IPO).

3 These projects have as their common goal the development of an international standard for the exchange of product data.

Acknowledgments

The author wishes to thank Dr. Kent Reed, Leader of the Computer Integrated Construction (CIC) Group for his continued support.

Special appreciation is extended to all those who contributed through many hours of discussion. They include: Buzz Bloom, Bill Burkette, Wim Gielingh, Howard Mason, Mary Mitchell, Augusto Nieva, John Owen, Mark Palmer, Bob Parks, Curt Parks, Ron Schuldt, Nigel Shaw, Joan Tyler, Anne Williams, Peter Wilson, Yuhwei Yang, John Zimmerman and many other members of the PDES and STEP projects.

Appreciation is also extended to the members of the Steering Committee of the IGES/PDES Organization for the opportunity to present the material contained in this proposal at their December 1989 meeting. The discussions resulting from that presentation served as the catalyst for the preparation of this document.

Finally, the author wishes to acknowledge that the proposed framework presented at the IGES/PDES Organization meeting in New Orleans and at the ISO TC184/SC4/WG1 meeting in Paris has been considerably refined and improved. Major contributions were made by members of the IPO Integration Resource Committee and by ISO Subgroups 5,6, and 7 (Data Architecture, Integration, and Product Data Modeling).[4]

[4] This document does not represent an official position of either the IGES/PDES Organization or ISO TC184/SC4 nor any of the committees or working groups of these organizations. Rather, it is presented as a proposal for consideration by these organizations. It has been assigned document number N440 of ISO TC184/SC4/WG1.

Table of Contents

The presentation of a proposed integration framework within the IGES/PDES Organization (IPO) and ISO TC184/SC4/WG1 is best discussed within a historical perspective. This paper begins with the 1986 PDES Initiation Effort and outlines developments in both PDES and STEP up to April 1990.

1. PDES Initiation Effort

The PDES Initiation Effort was a "proof of concept" project. It was to validate the methodology by which PDES would develop a product data exchange specification. The Initiation Effort concentrated on two aspects; formal descriptive languages and creating a methodology for the description of product data. The Initiation Task Group of the PDES Logical Layer Committee was to formulate and test the methodology for the description of product data. Conceptual modeling was the principal technology used in formulating the methodology. It was the Initiation Task Group that first addressed in detail the issue of a framework for product data modeling in PDES.

The Final Report of the Logical Layer Initiation Task [1] was a major product of the PDES Initiation Effort. It contained the definition of an information model architecture with three distinct categories of models (fig. 1). These included discipline models, resource models that collectively defined a logical layer model, and an intermediary category containing global models. The discipline models were to capture the application[1] specific view of discipline experts. Resource models were to represent aspects of product description that were commonly used by multiple discipline models (e.g., geometry and topology). The resource models of the logical layer were to contain only generic entities and structures common to many application areas (i.e., no discipline specific entities were to be present in the logical layer). The global models represented an informal description of the correspondence between the discipline models and the logical layer model.

The technical details concerning the development of the global models were not completely understood. Therefore, in the absence of an overall plan that addressed this issue, model development proceeded independently on discipline and resource models. Discipline models were developed for architecture, engineering, & construction (AEC) , electrical , and mechanical products. Resource models included geometry, topology, and other models that dealt with common aspects of a product's description.

---

1 An application is here defined as a process that puts product data to use. An application view is the meaning attributed to product data based on its use.

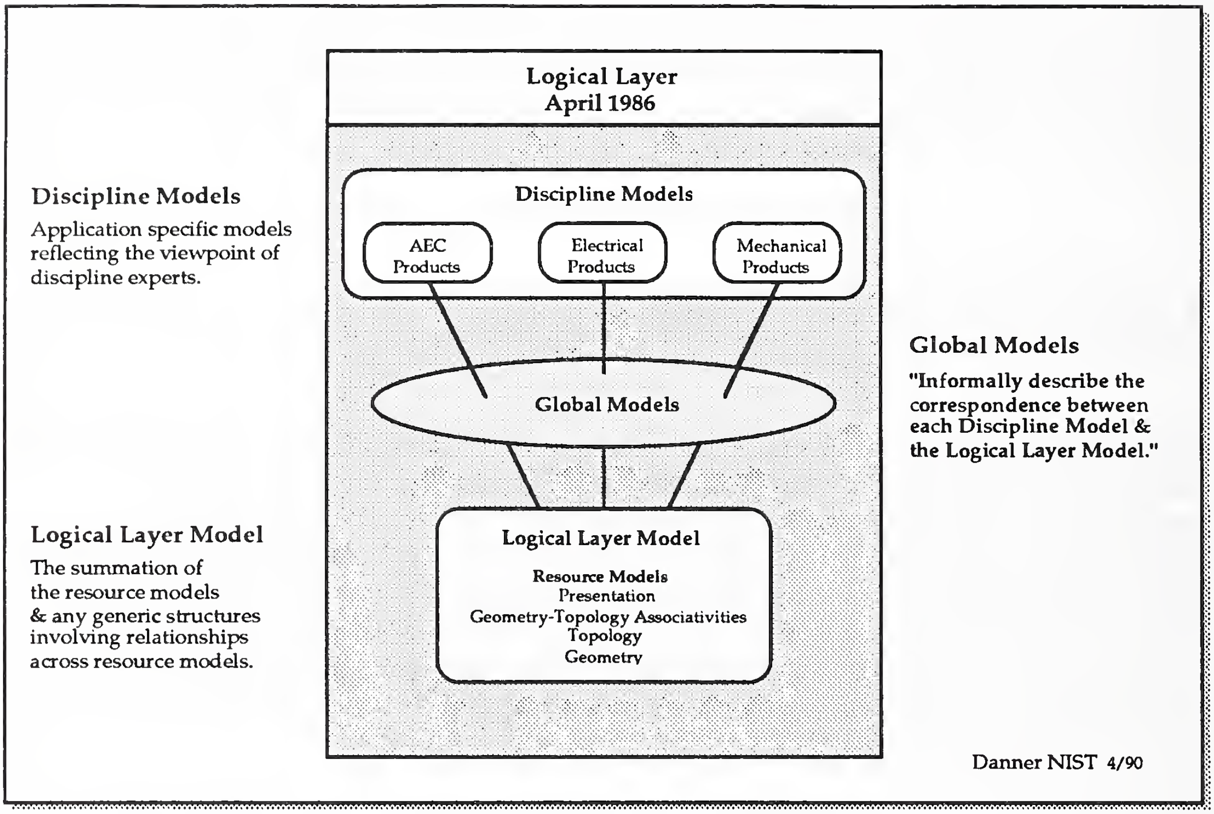

Figure 1. Initiation Effort Architecture

2. Integrated Product Information Model

By October 1988, the information models developed within the PDES and STEP projects had been assembled into a single conceptual model, the Integrated Product Information Model (IPIM) [2]. This was undertaken within the ISO Subgroup (SG) 6, Integration. The IPIM was the summation of all "resource" models (fig. 2). Due primarily to the modular approach adopted for its specification [3], every model or entity could potentially serve as a resource to any other model. The IPIM was viewed as an "entity pool" from which entities could be drawn on an ad hoc basis to create new models. Models effectively established partitions within the IPIM between what was considered relevant and what was not to achieve a given objective. This modular approach provided considerable flexibility for model developers. However, it required careful attention to the creation and contents of the partitions. Models developed in relative isolation of one another could create multiple and potentially conflicting ways of accomplishing the same objective. What was flexibility for the modelers could easily become ambiguity and redundancy for the implementors and users.

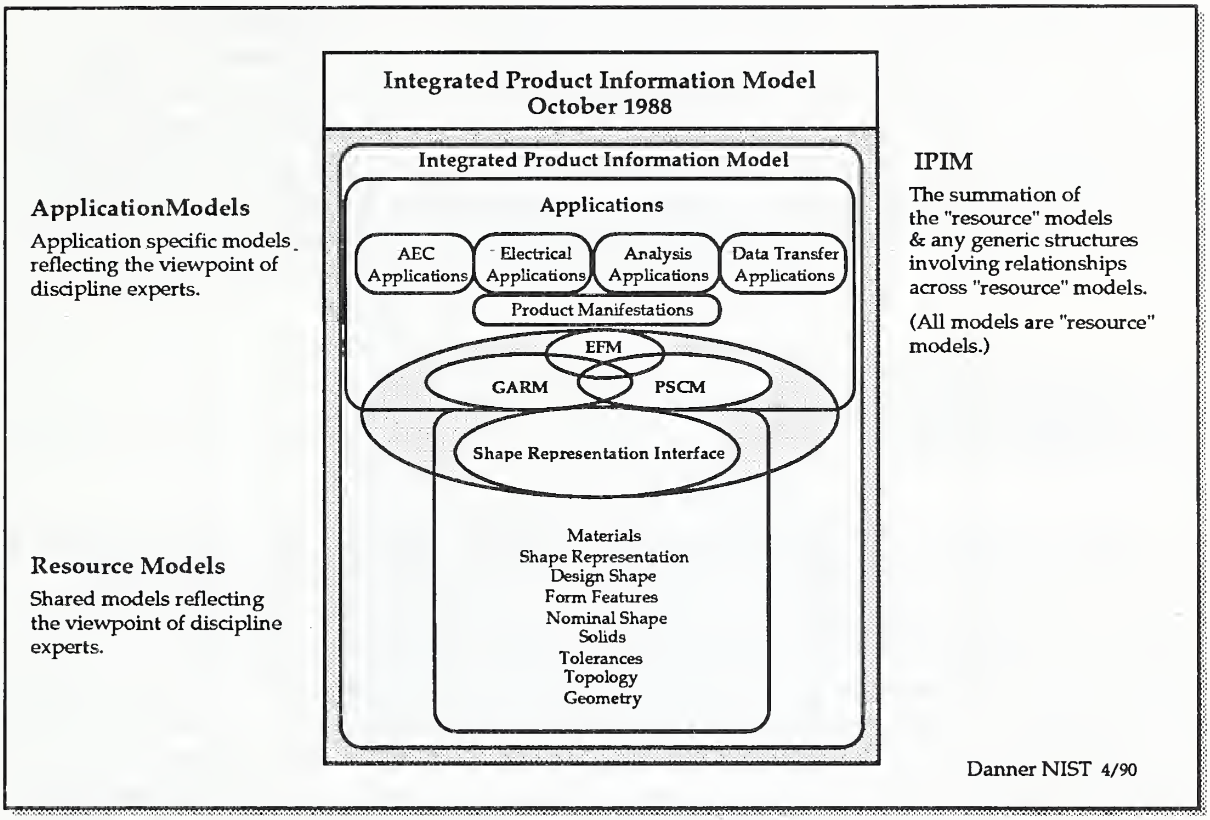

Figure 2. IPIM Architecture

Integration in the IPIM was limited to combining the "surface structure" of the models. That is, the meaning of the entities was reviewed in a literal sense defined by the modelers. If two models used different names for the same object or used the same name for different objects, a naming conflict existed that required resolution. An analysis of the underlying meaning of concepts was not undertaken, however. Therefore, conflict resolution was not required to resolve differences, for example, between the AEC and electrical disciplines' modeling of connectivity. Under the framework defined by the Initiation Task Group, a concept such as connectivity would have been a candidate for development as part of the logical layer. Generic entities would have been defined that were common to both disciplines.

Two principal criteria were applied in the entity level review of the IPIM. The first was schema independence of entities: each entity name was unique throughout the IPIM. The second was context independence of entity constraints: each entity included only constraints that were independent of context. Both of these criteria maintained the ability to use any combination of entities in developing discipline specific models (i.e., modularity was the governing consideration). Other criteria included minimal redundancy, but time and resource limitations prohibited their thorough use.

Information models that were being developed from the viewpoint of a particular discipline were referred to as application models. Resource models were those with capabilities used by other models. However, this distinction could only be made in a relative sense because few models used no other models. The distinction did not seem to be particularly important to the IPIM approach. The formal distinction between discipline models and a logical layer model with correspondences established between them had apparently been abandoned.

3. Shape Representation Interface

The Shape Representation Interface Model was the result of the PDES integration effort that had been completed by January 1988 and was therefore part of the IPIM. The Shape Representation Interface was created by the Integration Task Group of the PDES Logical Layer Committee (later to become the PDES Integration Committee). The task group included participants from committees with models that had been elevated to draft status within PDES. Application and resource models were candidates for integration. Models to be integrated were chosen based on both model development status and stability. Six models were chosen. They included the PSCM, Finite Element, Materials,[2] Tolerances, Form Features, Geometry, and Topology Models.

The integration was between two application models (i.e., the PSCM and Finite Element Models) and five resource models (fig. 3). The PSCM model was a general model, since it was developed to be applicable for any product. This meant that although the model was considered an application model, it had the potential for being used by more specific application models. It was concerned with the definition of a product in terms of its structure (i.e., in terms of the parts of which it is composed) and in terms of configuration management information about that structure. The Finite Element Model was more specific than the PSCM but portions of it could also be used by more specific models.

Analysis revealed that among the models chosen for integration gaps and unacceptable constraints were the major issues requiring resolution. The interface points between the models involved the definition of a product's shape. Shape could be represented in many different ways (e.g., wireframe, constructive solid, etc.). A means was sought of defining a product's shape independent of its representation. The shape representation interface model was created to provide that capability.

---

2 The Finite Element Model was the only model that dealt with materials. The Integration Task Group suggested that the FEM be divided into Materials and Finite Element Models.

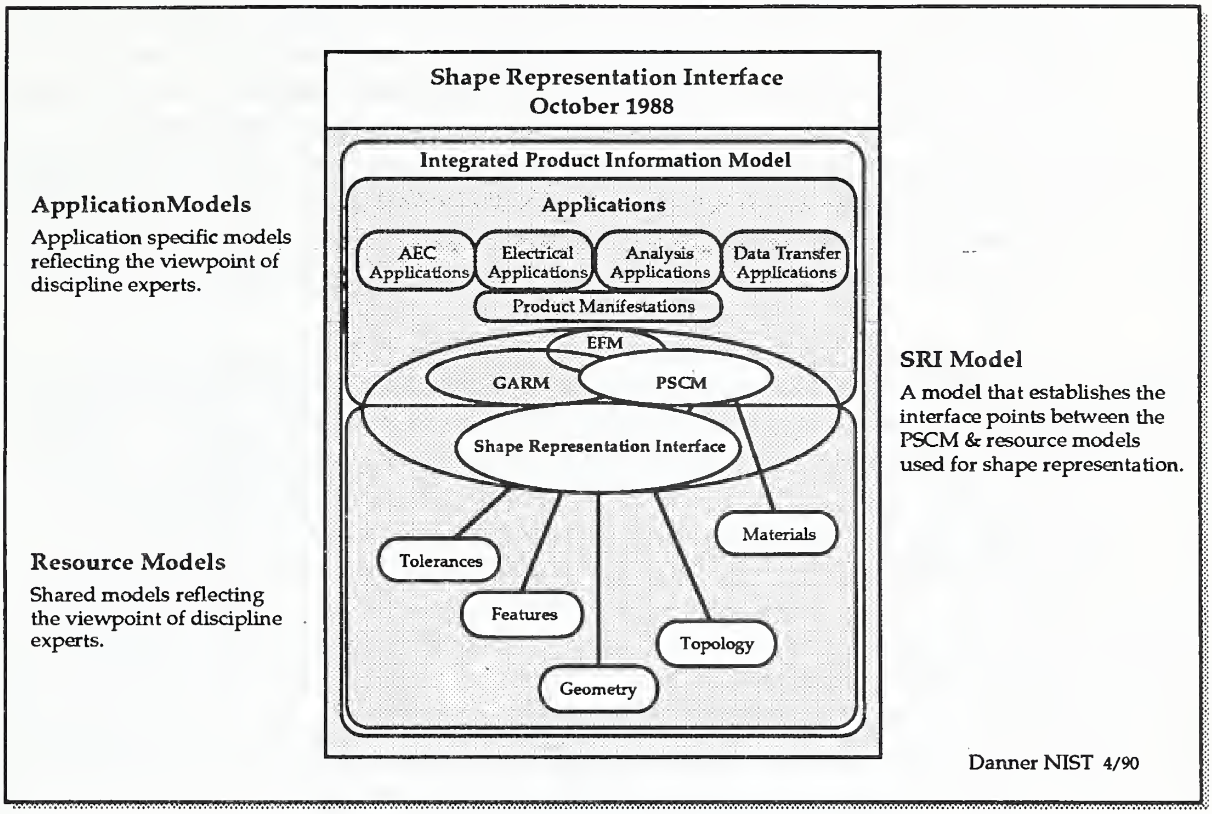

Figure 3. The Shape Representation Interface

The distinction between application and resource models made by the Initiation Task Group had been reaffirmed by the PDES Integration Task Group. The resource models provided required functionality for defining the shape of a product in terms of its representation. An application model was concerned with the definition of a product. Product definition could be general in nature, such that other application models could share its capabilities. Shape definition was one such general aspect of a product's definition. Material definition was another. Product structure and configuration management information about that structure were additional aspects. Other models within the IPIM also provided such general product definition capabilities.

4. Product Definition Models

By July 1988, the Integration Task Group of the Logical Layer Committee had become the Integration Committee. In January 1989 it was organized into two subcommittees, Integration Resource and the Integration Practice. Integration

Resource was to serve as a forum for the discussion of technical issues confronting the PDES project regarding the integration of models. It was responsible for developing a strategy for integration which forms a major aspect of this paper. Integration Practice was to execute the strategy developed by Integration Resource. Integration Practice was to include modeling experts and members of technical model development committees who were charged with the responsibility of acting as experts on particular models during the integration process. Integration was to take place in small working task groups.

Early in the discussions held by Integration Resource, it became evident that application models in the IPIM varied along a continuum of generalization (i.e., the degree to which they included generic rather than specific entities). Some models were very specific, such as the Ships Structural Systems Model. Others were more generic, such as the General AEC Reference Model (GARM), the Electrical Functional Model (EFM), and the PSCM Model (fig. 4) developed by experts from the AEC, Electrical, and Mechanical products disciplines.

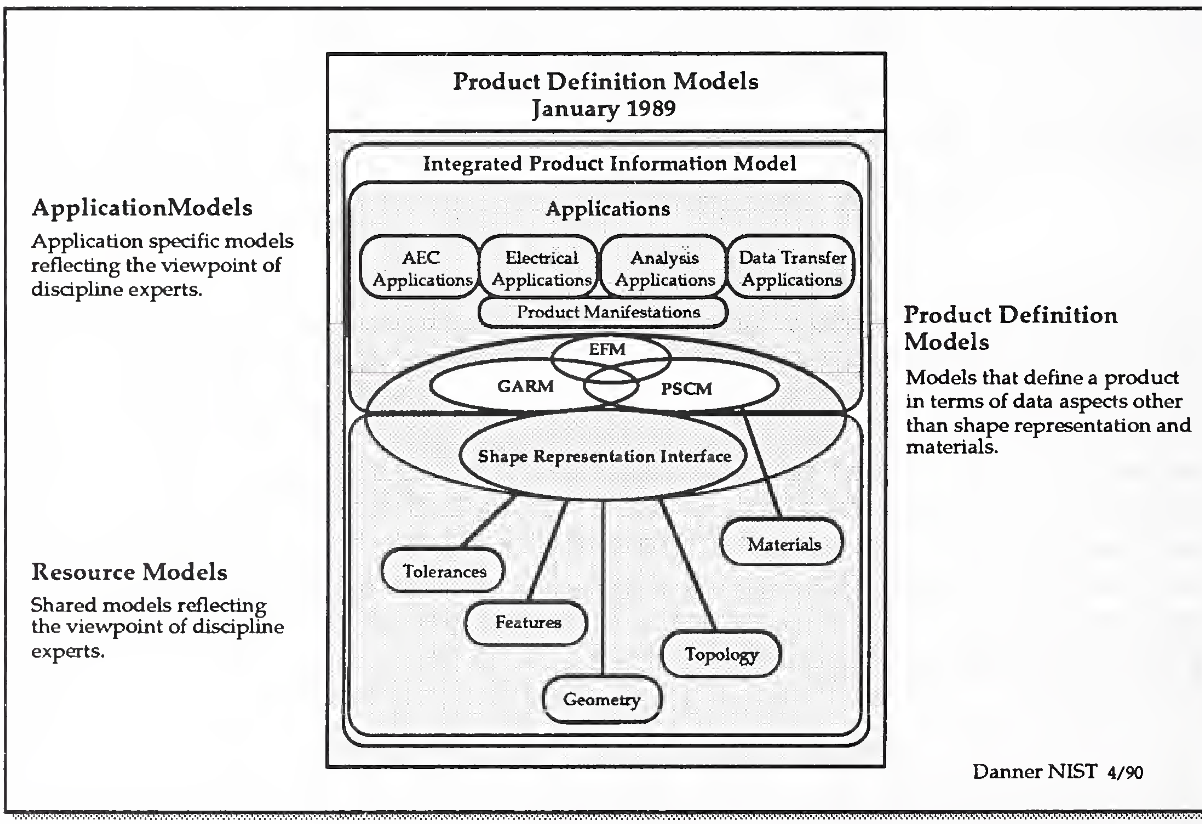

Figure 4. Product Definition Models

The general models appeared to provide on an ad hoc basis the function of identifying generic concepts envisioned for the logical layer model. Multiple approaches to the specification of generic aspects of a product were developing. The EFM was the most specific model (although it dealt with connectivity, a product aspect with broad applicability). The PSCM was generic in nature, focusing on a clear specification of product structure and its configuration management ramifications. The GARM was generic in nature but had a different approach to modeling product structure. Also the GARM included general product characterization and many entities that resulted from its consideration of product life cycle.

These models were viewed as being both complimentary and conflicting in their specifications. The importance of the generic specification of product data suggested a combined effort within the PDES and STEP projects. The initial focus was on the levels of generalization and the degree to which models at different levels were working together.

The integration of the very specific application models with the more generic models was unclear. Within the AEC Committee, for example, the integration of a specific model like the Ship Structural Systems Model with the General AEC Reference Model was proceeding slowly. A methodology for integrating models at different levels of generalization was absent. This suggested that the function of establishing correspondences between specific and generic aspects of a product was still relevant within the context of the IPIM.

5. Application Protocols

In June 1989, at the Frankfurt meeting of the ISO/TC184/SC4/WG1, application protocols [4,5] were acknowledged to serve an important role in determining how STEP should proceed. The concept of an application protocol (AP) was developed within the IGES/PDES Application Validation Methodology Committee. Its purpose was to 1) state explicitly the information needs of a particular application, 2) specify an unambiguous means by which information is to be exchanged for that application, and 3) provide a basis for standard conformance verification.

The elements of an application protocol are presented in Table 1. The scope and requirements documentation together with the application reference model serve as an explicit statement of the information needs of an application for which consensus is achieved. The application interpreted model, and usage guide serve to specify an unambiguous means by which information is to be exchanged for that application. The conformance requirements documentation provides the basis for standard conformance testing.

Table 1. Elements of an Application Protocol
Scope & Requirements Documentation A statement of application domain information needs that must be accommodated for useful information exchange. (An activity model is an important part of this documentation.)
Application Reference Model (ARM) An information model that specifies conceptual structures and constraints used to represent application information.
Application Interpreted Model (AIM) An information model that specifies the information represented by an associated ARM using standardized data constructs (i.e., IGES or STEP).
Conformance Requirements A description of conformance criteria and test purposes designed to verify the compliance of of computer systems to the standard.
Usage Guide Information on employing the AP for data exchange.

The adoption of the application protocol methodology reestablished the basic architecture of the Initiation Effort (figs. 1,5). Application reference models were application specific models with clearly defined scopes and functional requirements. This constituted a refinement of the discipline model concept. Applications could be subdiscipline, discipline, or multidiscipline in scope. Resource models were to be evaluated in an application context. They were models used to provide the required functionality of application protocols. The application interpreted models were described as intermediary models that

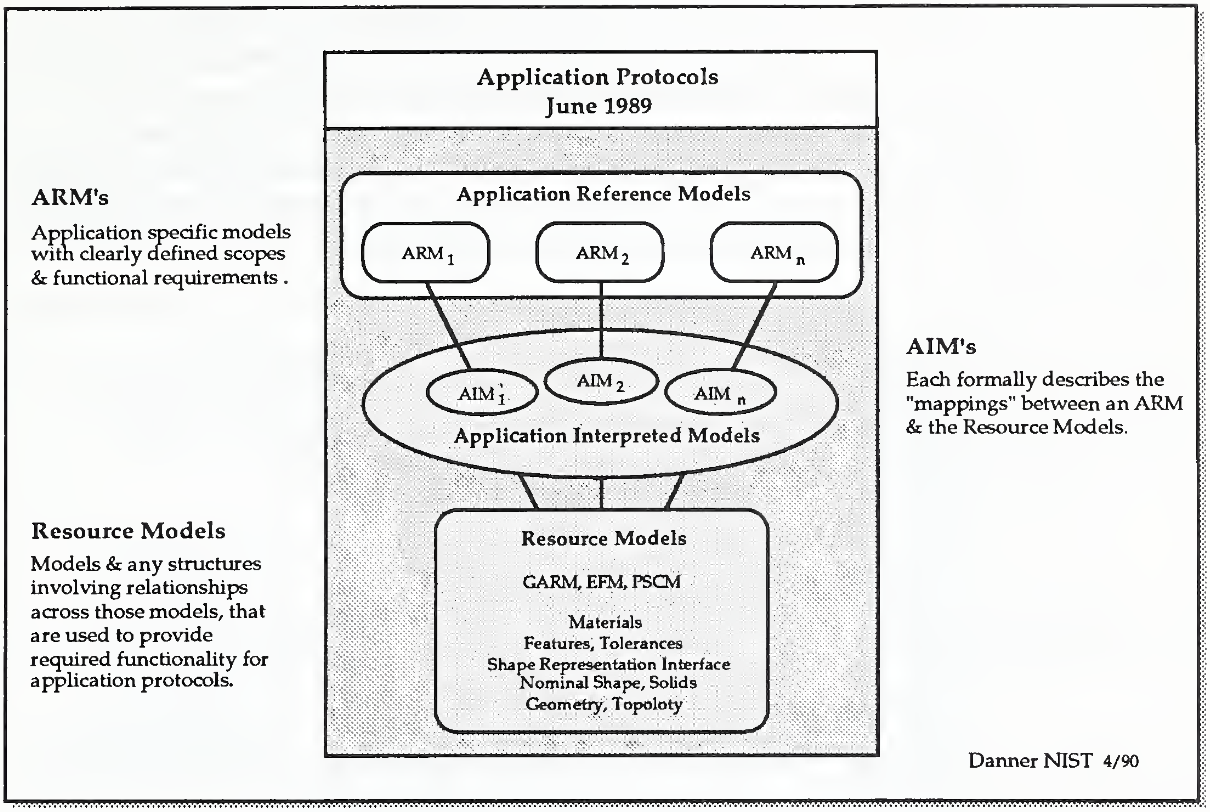

Figure 5. Application Protocols

provide a formal description of the mapping between the application reference model and the resource models.

The informal description of correspondence identified by the Initiation Effort had been refined by the AP methodology. A formal description that established equivalence of intent between an application reference model and an application interpreted model was required if the methodology was to be used successfully for testing. The exact nature of the formal description was still not understood, but it was recognized as essential.

The methodology emphasized explicit and well documented elements for an application protocol. Two significant outstanding questions remained, however. The first arose from the incomplete understanding of the technical details of the process by which equivalence was to be maintained (i.e., correct interpretation). The second was that the IPIM had alternative ways of representing the same information resulting from the uncoordinated approach adopted during its development.

Each application protocol could develop a unique way of achieving its goal. That might involve the use of an abstract model (e.g., PSCM or GARM). It might, alternatively, choose to develop its own way of representing information even though several solutions may already exist. Application protocols could provide unambiguous communication for an application in an environment where more than one solution existed since the methodology had been originally developed to deal with the same situation in IGES. However, the potential for separate application protocols that were fundamentally incompatible raised once again the issue of planned rather than ad hoc development within the PDES and STEP projects.

The development of compatible rather than "peacefully coexisting" (i.e., incompatible) application protocols was desirable. Compatible AP's could be easily merged and altered as information requirements changed. A single coherent representation of common product data was required.

6. Planning Model

Numerous attempts had been made to develop a planning model for the PDES and STEP projects. By early 1989, considerable progress had been made toward arriving at a consensus in ISO SG5, Data Architecture. Criteria had been established, and a planning model was identified as having met those criteria [6].

The planning model was presented as a first step toward explicitly stating the scope and nature of the generic information requirements of the PDES and STEP projects. As such, it could be used to analyze current models, to identify areas of strength and weakness, and to plan a strategy for future development.

The planning model described the PDES and STEP projects as having three distinct modeling tasks. They include modeling data used for the definition, representation, and presentation of a product. Definition data captures the essential qualities of a product independent of whether or not a computer is used. The decomposition of a product into parts (already addressed by both the PSCM and the GARM) and the characterization of a product in terms of properties were the two initial types of product definition data identified. Both of these were seen to vary over the life cycle of a product. Representation data was described as data required by computer systems (particularly to capture the shape of a product). Presentation data was described as data used to display product definition and representation data for a user.

The SG5 also held discussions concerning the combination of the proposed planning model with a detailed product data classification system [7]. The classification system was not limited to the current scope of the PDES and STEP projects, but rather was designed to classify all product data. It was envisioned as providing assistance in defining an appropriate scope for STEP. It also elaborated on the details of the product definition data identified in the planning model. Functional, physical, and programmatic definition data were identified (fig. 6). Physical definition data was further divided into that used to define material and shape properties. The product data in this system was also expected to vary over the life cycle of a product.

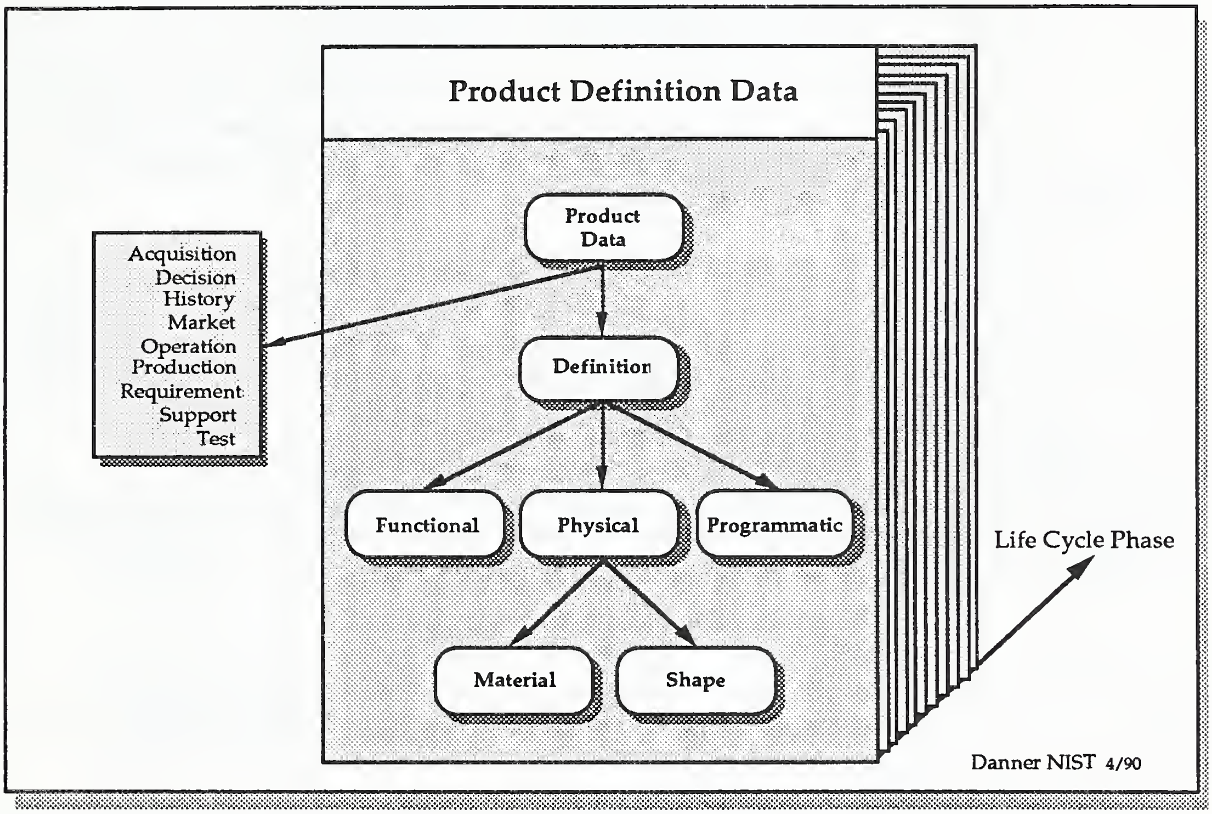

Figure 6. Product Data Classification

The classification system for product definition data was consistent with the classes of models identified by the PDES Integration Committee. The task group had integrated models dealing with the physical definition of a product. The PSCM Model was a product definition model (that included many life cycle aspects). The Shape Representation Interface and the Materials Models were property definition models. They served in the definition of shape and material properties. Representation models provided details about shape representation.

The same fundamental structure was developed independently by two groups; a model classification and planning group and a model integration group. This structure was evidence of an emerging IPIM architecture.

7. Generic Product Data Model

By October 1989, deep structure integration was being used as a means, within the PDES Integration Resource Subcommittee, of uncovering fundamental concepts within product data models. The term deep structure integration draws by analogy from a distinction made in linguistics between surface structure and deep structure representations of meaning [8]. The models of the PDES and STEP projects needed to be examined both in terms of the surface representation of the particular discipline for which they had been developed, and in terms of more fundamental underlying structures applicable to products in general. Deep structure integration provided the means by which the IPIM architecture could be refined and its models truly integrated.

The results of the deep structure approach to integration were consistent with previous work that had reaffirmed and refined the framework of the Initiation Effort (figs. 1,3,5,7). The proposed integration framework has a Generic Product Data Model (GPDM) as its central feature. The GPDM captures in a single coherent representation, common elements of product data. It provides a context independent description of a product in terms of generic product definition facts[3] (i.e., facts that apply to any product). It, therefore, serves as the foundation for application interpreted models that are context dependent forms of GPDM product definition facts.

The GPDM provides a structure for the models of the logical layer that are resources for application protocols. Only these models are integrated into the IPIM. They include generic product definition, property definition, representation & presentation, and mathematical resource models. The explicit logical structure provided by the integration framework constitutes a refinement of the IPIM with an appropriate emphasis on information about products.

Application interpreted models formally describe the interpretation of generic facts about a product in a specific application context. They make use of lower level resource models through the GPDM. Application reference models have access to the GPDM by way of application interpreted models.

The GPDM provides access to other model classes of models within the IPIM. Property definition models provide access to representation and presentation models which in turn provide access to mathematical resource models. The product definition facts are the fundamental elements of this structure which provide an integration focus for the IPIM architecture.

---

[3] Product definition facts are canonical relations that exist between the various types of product definition data. (See [9,10,11,12].)

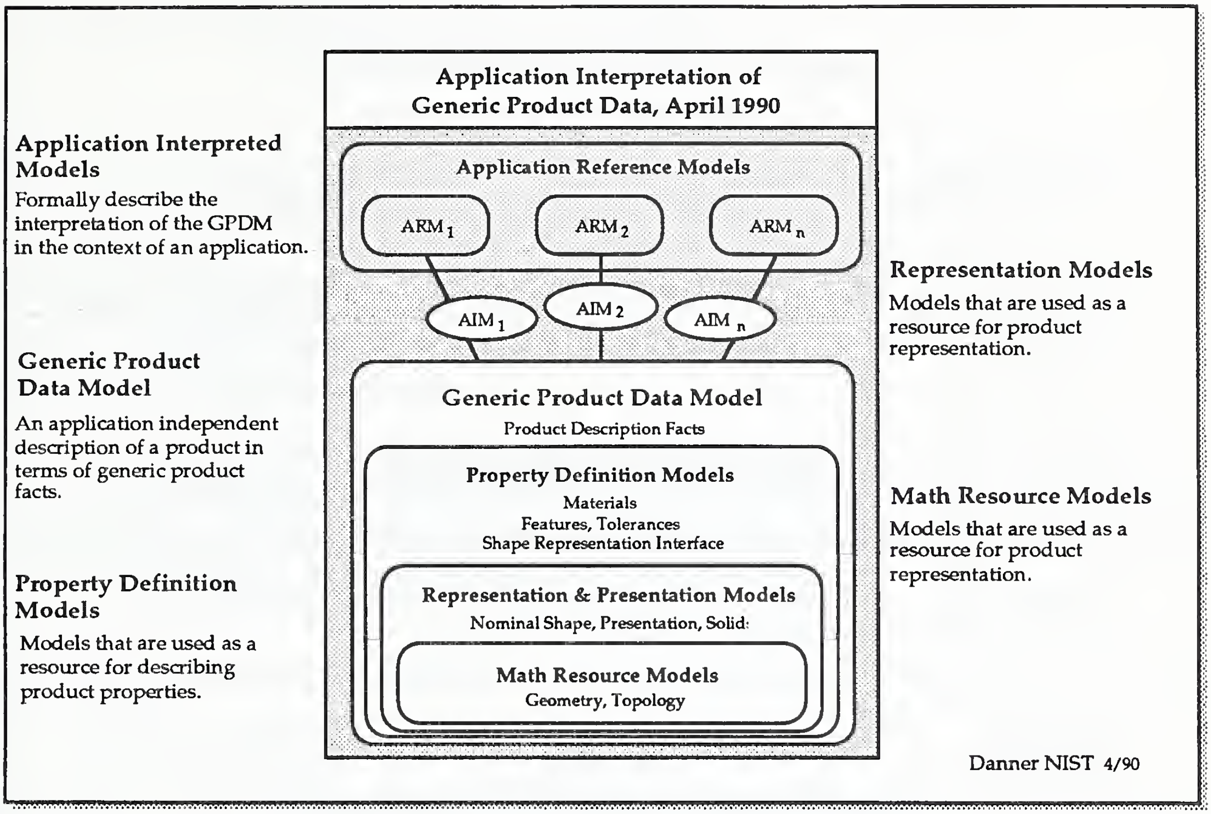

Figure 7. Application Interpretation of Generic Product Data

Product Description Facts

As of December 1989, a preliminary list of generic facts had been identified for inclusion in the GPDM (table 2). These facts are represented as elemental conceptual structures [10,11] that are used to represent a product in terms of functional, programmatic, and physical aspects [7, 12].[4] Each conceptual structure embodying a generic fact is capable of being interpreted in application contexts.

The product definition facts are the elemental building blocks from which applications construct the definition of specific products in interpreted models. The GPDM specifies the standard product definition facts explicitly and stipulates the ways in which these facts can be combined. The GPDM, therefore, functions both as a library of precisely modeled facts from which an application can draw, and as a template for more complex structures among these facts to be used both by application and integration efforts.

---

4 As an example, a product's ability to sustain loads is functional, its configuration management is programmatic, and its shape is physical.

Table 2. Product Definition Facts	
Accumulation	A functional element is defined in terms of a collection of other functional elements, expressions, and constraints.
Assembly	A collection of physical objects that are joined.
Composition	A physical object is defined in terms of a collection of parts and compositional constraints.
Connectivity	Functional elements have associated connectivity linkages.
Functional Definition	A physical object is defined by its functional elements.
Idealization	A physical object has an associated shape representation other than that which describes the space it occupies.
Location & Orientation	A physical object has an associated location and orientation in terms of a reference.
Material Definition	A physical object has associated description of matter of which it is made.
Programmatic Definition	A physical object has associated programmatic information.
Shape Definition	A physical object has an associated shape representation that describes the space it occupies.

8. STEP Integration Framework

Figure 8 presents the seven classes of models contained within the proposed STEP integration framework. Four context independent classes of models form the IPIM; the GPDM, the property definition models, the representation and presentation models, and the mathematical resource models. Three context dependent classes of models are used in the development of application protocol models (APMs); application reference models, general context models, and application interpreted models.

All classes of models need to be explicitly documented, checked for consistency, and maintained as changes take place over time. All seven classes of models are essential for meaningful product data exchange. Therefore, each class of model should be identified as a part of the STEP documentation.

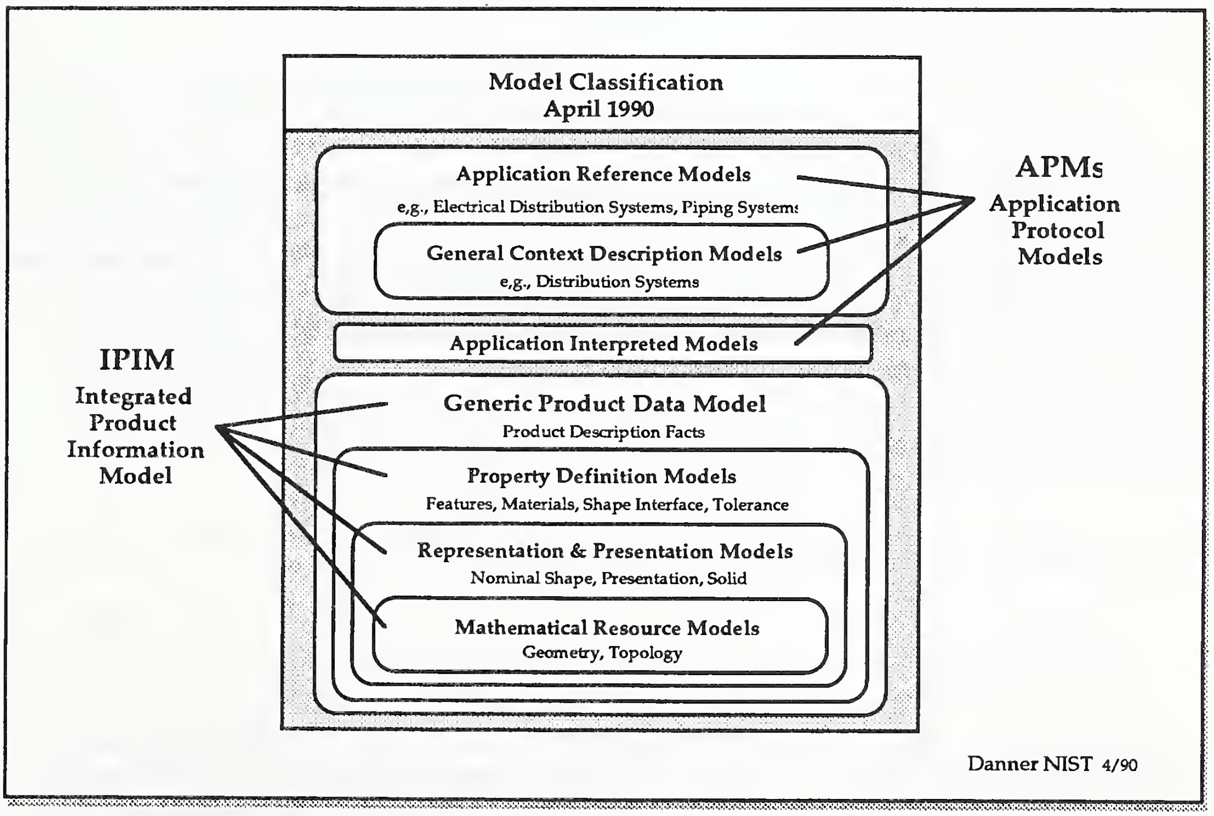

Figure 8. Model Classification

8.1 Context Independent Models

The Integrated Product Information Model is composed of context independent models. It includes the four classes of models.

Generic Product Data Model

An information model that provides for the description of generically applicable aspects of products.

The GPDM, described in section seven, is the central feature of the proposed structure for the IPIM. It places the emphasis of STEP clearly on the exchange of data that describes products. This approach can serve both the immediate industry needs as well as providing conceptual structures appropriate for new generations of computer systems. A modular approach to product description facts provides the necessary foundation for STEP.

Property Definition Models

Information models that describe fundamental traits and characteristics.

The product definition facts of the GPDM are by design limited to elemental concepts. Physical product definition data have proven to require entire models for their specification. The shape definition and material definition facts serve as integration points to these models. They integrate shape properties from the Shape Representation Interface, Features, and Tolerances Models as well as material properties from the Materials Model within the GPDM. Similarly, interfacing facts in the Shape Representation Interface integrate shape aspect definitions with representation models. Interfacing facts are expected to also integrate presentation models with property definition models.

Representation and Presentation Models

Information models that describe an image which stands for a real world object or concept.

Shape representation data has been captured within the Nominal Shape and Solids Models. Presentation data has been captured within the Presentation Model. These models serve as representation and presentation resources to the property definition models. These models, in turn, make use of mathematical resource models.

Mathematical Resource Models

Information models that provide formal mathematical descriptions.

The mathematical resource models include the Geometry and Topology Models. They act as resources to the representation and presentation models.

8.2 Context Dependent Models

Application protocol models are by definition context dependent models. They serve to specify the use of information in a particular context.

Application Reference Models

An information model that specifies conceptual structures and constraints used to represent application information.

General Context Description Models

An information model that specifies common conceptual structures and constraints usable by a number of more specialized ARMs.

Application Interpreted Models

Information models that describe the information represented by an associated ARM using IPIM data constructs.

9. Document Composition

The STEP document composition was developed at the June 1989 meeting of WG1. Model classes based on the integration framework can serve as a logical basis for the IPIM portion of the document composition (fig. 9). The document composition serves as a graphical depiction of how the documentation is divided into meaningful parts.

For there to be a meaningful first version standard, each of the model classes needs to be present. This includes both the context independent and dependent models. The most comprehensive approach begins with the development of application protocols to drive the development and testing of necessary IPIM

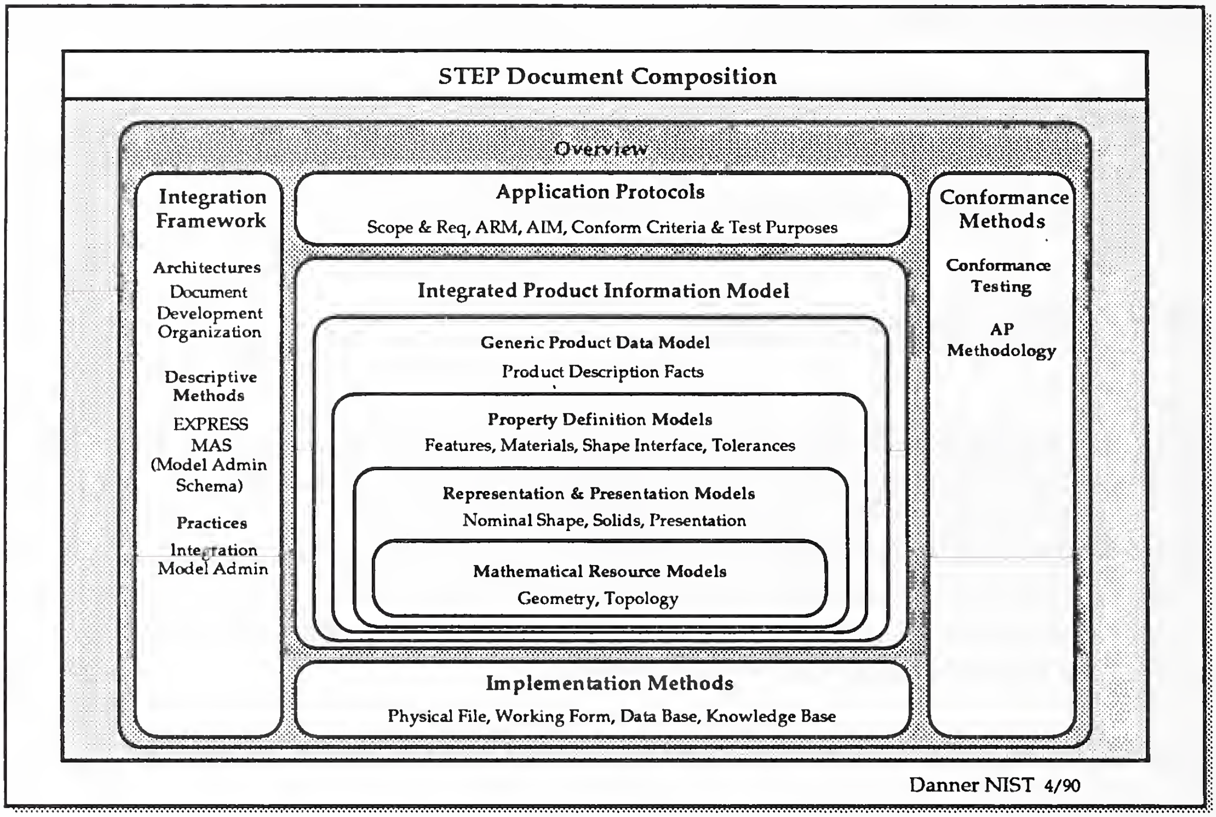

Figure 9. STEP Document Composition

components [13]. The application protocols must represent how the standard is to function.

Application protocols with limited but well defined scopes are required. They will call upon each level of the IPIM, providing a focus for integration efforts at each level.

The first version should include only tested application protocols and the tested entities of the IPIM used by these application protocols [4,14]. These first APs are also used to validate the integration framework and application protocol methodologies. It also serves to test the Express language. Express provides the modular approach to model specification. The proposed integration framework provides the structure that establishes coherent meaning among the developed modules.

10. STEP Development Process

By January 1989, it was accepted that a number of coordinated functional activities were required to further develop the IPIM and begin the development of APs. Figure 10 summarizes the principle functions to be performed in the development of the STEP IPIM and APs.[5]

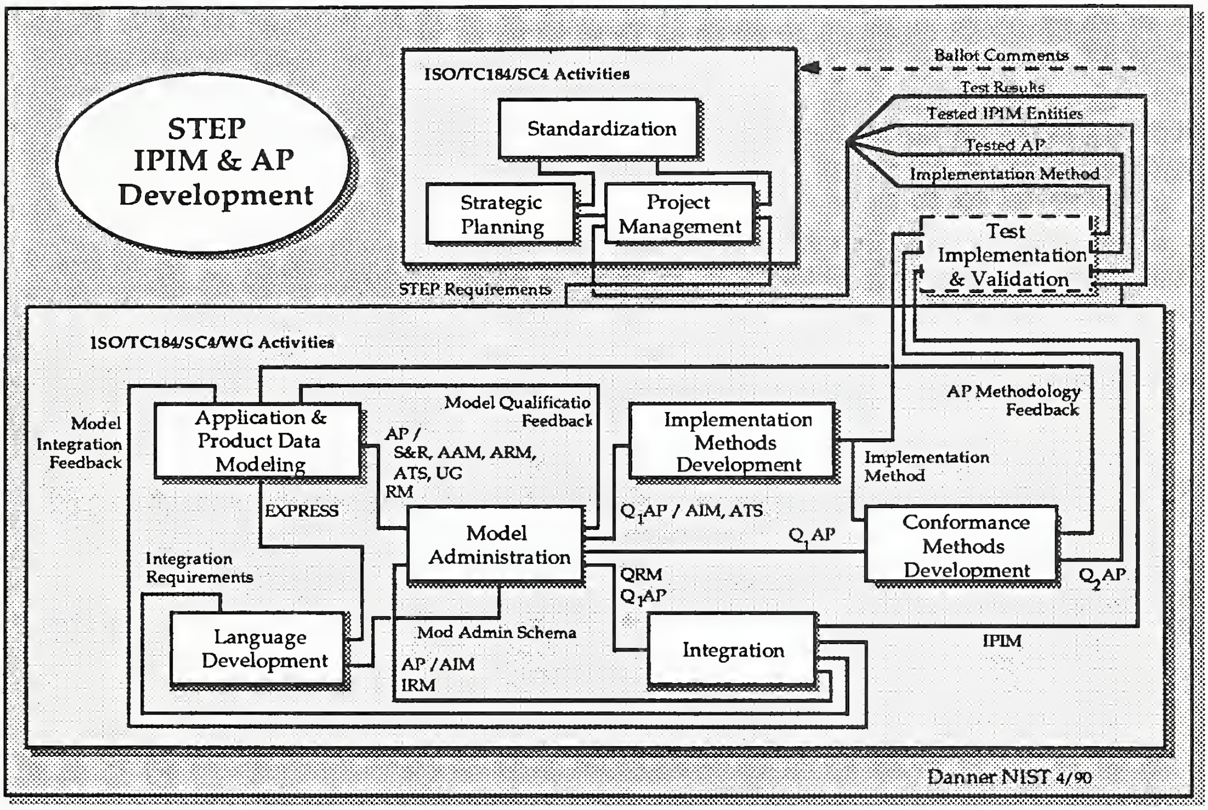

Figure 10. STEP IPIM & AP Development

Abbreviations:

AAM	Application Activity Model	QAP	Qualified Application Protocol
AIM	Application Interpreted Model	QRM	Qualified Resource Model
AP	Application Protocol	RM	Resource Model
ARM	Application Reference Model	SR	Scope & Requirements
ATS	Abstract Test Suite	IPIM	Integrated Product Information Model
IRM	Integrated Resource Model	UG	Usage Guide

5 Establishing and coordinating the organizational units within the PDES and STEP projects and the working relationships between these units are ongoing activities. See the Appendix for current and proposed approaches to dealing organizationally with the required functional activities.

11. Summary & Conclusions

The PDES Initiation Effort established the original framework for product data modeling within the PDES and STEP projects. It established a distinction between generic information models of the logical layer and application layer models that used the generic models as resources. This distinction was abandoned for the modular framework embodied in the Integrated Product Information Model (IPIM). The IPIM presented every model as a potential resource for every other. Integration was limited to the maintenance of modularity.

Evidence began to accumulate, however, that the original framework was still relevant within the context of a modular approach.

> A criterion for distinguishing product data models at the logical layer from those at the application layer was established. Product data models at the logical layer did not refer to a particular product or a particular use of the product data they specified (i.e., they were independent of application context).
>
> The current scope of PDES and STEP information models was described as encompassing product definition, representation, and presentation data. Product definition data was further categorized as having functional, physical, and programmatic aspects. Physical definition data included material properties and shape.
>
> Four classes of models within the logical layer were identified during the deep structure integration of models concerned with context independent product definition. Models that provided generic product definition, property definition, representation & presentation, and mathematical resources.
>
> The application protocol methodology described context dependent application models as an interpretation of context independent models.

Each of these developments led to the same conclusion. An integration framework is required that contains a context independent IPIM and application protocol models. The IPIM is composed of four classes of models; a Generic Product Data Model, property definition models, representation & presentation models, and mathematical resource models. The deep structure integration of these models produces an IPIM with an explicit well formed architecture. The IPIM then serves as a resource to application protocols.

Application protocols involve three classes of models. Application reference models define specific application contexts. General context models can be used as a resource by several application reference models. Product definition facts of the Generic Product Data Model are interpreted by both general and application contexts through mappings. The formal statement of the mappings are the application interpreted models.

Each class of model in the integration framework needs to be represented in the first version of the standard. Mechanisms for implementation and conformance testing should also be demonstrated. Finally both the Express language and the integration framework should be rigorously evaluated to ensure that all of the requirements necessary for the exchange of product data are in place.

References

[1] Kelly, J.C.; Final Report of the PDES Logical Layer Initiation Task, April, 1986.

[2] Smith, B. and Rinaudot, G.; Product Data Exchange Specification: First Working Draft, NISTIR 88-4004, National Institute of Standards and Technology, U.S. Department of Commerce, December, 1988.

[3] Express Information Modeling Language Reference Manual, N386, ISO TC184/SC4/WG1, October, 1989.

[4] Harrison, R.J. and Palmer, M.E.; Guidelines for the Specification and Validation of IGES Application Protocols, NISTIR 88-3846, National Institute of Standards and Technology, U.S. Department of Commerce, January, 1989.

[5] Woodall, A., Pearson, M., Owen, J., Shaw, N., and Palmer, M.; Critical Issues for Application Protocols, ISO TC184/SC4/WG1 N431, 1989.

[6] Tolman, F. and Geilingh, W.; STEP Planning Model, ISO N328, April 1989.

[7] Schuldt, R.L.; Universal Data Classification: The Key to Enhanced Communications in a TQM Environment, American Institute of Aeronautics and Astronautics, 1989.

[8] Winograd, T.; Language as a Cognitive Process: Syntax, Addison-Wesley Publishing Company, Reading, MA, 1983.

[9] Brodie, M.L., Mylopoulos, J. and Schmidt, J.W.; On Conceptual Modelling, Springer-Verlag, NY, 1984.

[10] Danner, W.F.; A Global Model for Building-Project Information: Analysis of Conceptual Structures, NBSIR 88-3754, National Bureau of Standards, U.S. Department of Commerce, April, 1988.

[11] Tyler, J.; A Planning Model for Unifying Information Modeling, National Institute of Standards and Technology, U.S. Department of Commerce, in progress.

[12] Eastman, C., Bond, A., and Chase, S.; A Formal Approach for Product Model Information, UCLA School of Architecture and Urban Planning, November, 1989.

[13] Mitchell, M.J., Yang, Y., Ryan, S., and Martin, B.; Data Model Development and Validation for Product Data Exchange, Proceedings of the Autofac '89 Conference, Detroit, October, 1989.

[14] Danner, W.F., Mitchell, M.J., and Yang, Y.; STEP Integration Framework and Phased Integration Plan, National Institute of Standards and Technology, U.S. Department of Commerce, in progress.

NIST-114A (REV. 3-89)	U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY	1. PUBLICATION OR REPORT NUMBER NISTIR 90-4295
		2. PERFORMING ORGANIZATION REPORT NUMBER
	BIBLIOGRAPHIC DATA SHEET	3. PUBLICATION DATE APRIL 1990

4. TITLE AND SUBTITLE

A Proposed Integration Framework for STEP
(Standard for the Exchange of Product Model Data)

5. AUTHOR(S)

William F. Danner

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)	7. CONTRACT/GRANT NUMBER
U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY GAITHERSBURG, MD 20899	8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This paper presents a proposed integration framework for product data modeling. The framework provides for the representation of functional, programmatic, and physical product data across all phases of a product's life cycle. It provides a single coherent approach to product data modeling for the specification of application views. Most importantly, it creates an open system that encourages the innovative use of information.

The framework has as its major feature an integrated product information model with four conceptual levels. These include generic product description, property description, representation & presentation, and mathematical resources. A generic product data model is the key element of the framework. It is composed of application-independent facts common to all products. The generic model is a distillation from the models currently under consideration by the STEP[1] and PDES[2] projects.[3] The generic product data model meets the requirements of multiple application areas by providing for the interpretation of generic facts in specific contexts. It also provides a logical structure for the integrated product information model which is used by application models to fulfill user requirements.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Framework, Information Modeling, Integration Framework, IPO, PDES, Product Data, Product Data Modeling, Product Modeling, Standard for the Exchange of Product Model Data, STEP

13. AVAILABILITY	14. NUMBER OF PRINTED PAGES
[X] UNLIMITED	28
[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).	15. PRICE
[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.	A03
[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.	

ELECTRONIC FORM