The American Society of Mechanical Engineers

Reprinted From
Engineering Data Management: Key To Integrated Product Development
Editor: T. R. Chase
Book No. H00766—1992

THE USE OF APPLICATION MODEL VALIDATION IN TESTING A PROPOSED STANDARD

Mary J. Mitchell
National Institute of Standards and Technology
Gaithersburg, Maryland

1 Introduction

From an information system development perspective, logical data modeling techniques have traditionally served in two roles. The first role is as a method of describing the information requirements of an application system. The second role is as a mechanism for integrating the requirements from a number of applications into a single logical and consistent schema so that data can be shared by multiple applications. The Standard for the Exchange of Product Model Data (STEP) project is developing an international standard[1] which uses data modeling as the basis for a multi-national and multi-enterprise integration effort. STEP is designed to provide a complete, unambiguous, computer-readable definition of the characteristics of a product throughout its life cycle. STEP product definition specifications are implementation independent, though implementation interface techniques provide the communication mechanisms for applications using file exchanges[2] or shared databases. Because of the diversity of applications that are within the scope of STEP, the integration generally causes extensive changes but the changes are justified if all of the information requirements are supported.

This paper describes a method for ensuring that all application information requirements are met.[3] While much of the terminology is specific to the development methods used for STEP [Danner92,Palmer91], any large scale, multi-enterprise integration activity should find this material useful. The method for validating a logical data model should be applicable even if the data modeling technique is other than one of those used by STEP.

Confidence in a standard by its user community is absolutely essential for a standard to gain acceptance. Validating that user needs are supported by the proposed standard is the foundation for any useful standard. Similarly, confidence that an integrated data model supports user needs is essential in any implementation of any large, complex database system. Significant investments will be required to implement STEP applications. Because of the integration, STEP will contain a number of untried solutions to technical problems. The STEP user community must not ask vendors to develop implementations based on specifications that have unknown levels of risk. Thorough, appropriate model validation testing before standardization can minimize this risk. In addition, this testing will reduce the need to continually 'patch' the standard to correct design flaws uncovered by implementing the standard. The use of proven integrated data models provide

[1] The International Organization for Standardization (ISO) is developing ISO 10303 -- Industrial Automation Systems: Product Data Representation and Exchange in Technical Committee 184 (TC 184) Subcommittee 4 (SC4) on Industrial Data and Global Manufacturing Programming Languages. For an overview of ISO 10303 refer to *Part 1: Overview and Fundamental Principles* [ISO1].

[2] The initial release of STEP supports only a file exchange interface but an interface for exchange through database transactions is under development.

the mechanism for controlling these risks. Developers of large integrated databases should want the same sort of proof that their integrated logical data model is correct prior to implementing numerous applications.

An Application Protocol (AP) is a specification for a portion of the product data described by STEP [Palmer91]. APs are the parts of STEP that are to be directly implemented by vendors of software products who adopt STEP. The entirety of STEP consists of all STEP APs along with supporting definitions for the elements that are common to many APs, the specification languages, implementation interfaces, and conformance testing requirements. AP specifications are derived from a set of integrated data models, called Integrated Resources[4], that contain product definitions in a broad context, e.g. across the product life cycle and across manufacturing disciplines. In the AP, these general descriptive definitions for products are interpreted to describe specific data requirements for a given application. Within an AP there is no redundancy and there is consistency with the integrated resources. Due to this consistency, the STEP AP specifications permit product information to be unambiguously exchanged or shared between implementations on dissimilar systems.

Procedures are needed to ensure that the technical solutions provided by STEP APs and integrated resources will work in a practical sense. The term application model[5] is used throughout this paper to refer to the component information models in an AP or any other domain-specific information model with similar properties. In 1991, a methodology was proposed for validating STEP AP models [Mitch91]. The National PDES Testbed[6] is used to test the validity of application models and the software from its Validation Testing System (VTS) supports this methodology. This document introduces the methodology and shows how the VTS applies the proposed methodology to support AP model validation. The methodology and software build on previous experience with testing STEP from an application's perspective. Another type of testing, the conformance testing of vendor AP implementations, should leverage the VTS as well as some of the outputs from application model testing.

A detailed discusion of the VTS can be found in a series of reports from the National PDES Testbed [Mitch90, Mitch91, Morris91a, Morris91b, Morris91c]. Section 2 of this document gives an overview of the validation testing process for APs. Section 3 describes the activities which comprise the validation testing methodology [Mitch91]. The last section describes the future directions for the VTS software, based on experiences with an interim software system used at the National PDES Testbed. These experiences provide a basis for the VTS software architecture described by Morris [Morris91c]. Finally, the overlap of software requirements in the validation of application models and in the conformance testing of vendor AP implementations is discussed.

2 Overview of Validation Testing

Application protocol development and testing is a complex process. It involves the synthesis, analysis, and manipulation of large amounts of diverse information. Most of the process relies exclusively on human capabilities for analysis, judgment, and interaction; however, part of this process can and should be automated. The strategy for automation is based on an analysis of the information flow of the AP development and testing process and initial experiences with automation for validation testing at the National PDES Testbed. This section describes the validation process in general. For a more detailed presentation, refer to the proposed AP model validation methodology [Mitch91].

Validation testing of AP information models determines whether the AP does what it is intended to do, i.e., meets the functional requirements that led to its development. The integrated resources of STEP are also shown to be capable of supporting the application area. The proposed approach is to validate AP models by simulating the behavior of relevant applications using

[4] The initial Integrated Resources in STEP include: Integrated Resource Fundamental Concepts, General Shape Representation Concepts (which includes geometry and topology), Representation Structures, Product Structure Configuration Management, Presentation, and General Draughting Concepts. For a technical description refer to *Part 41: Integrated Generic Resources – Fundamentals of Product Description and Support* [ISO41].

[5] The information models that are components of an AP are called application interpreted models (AIMs) and application reference models (ARMs). Other domain specific information models include context-driven integrated models (CDIMs) which are used to evaluate STEP resource models [CDIM A1] and various AP precursor models existing outside of the standardization process for purposes such as vendor prototyping. The validation methodology is applicable to these other application models if STEP development methods are used. The first priority of the VTS project is to support the requirements for validation of ARMs, AIMs and CDIMs.

[6] The National PDES Testbed is located at the National Institute of Standards and Technology. Funding for the Testbed Project has been provided by the Department of Defense's Computer-Aided Acquisition and Logistic Support (CALS) Office. The work described in this document is funded by the United States Government and is not subject to copyright.

industry-contributed data. The validation tests are identified by examining the application processes. The types of data required to perform each activity in the application process are specified in detail. Realistic data from the application domain is associated with each of these tests. Multiple sets of data may be used to ensure that the expected range and variation of industry uses can be supported by the application model. The data needed to perform a specific process or generated by a specific process is then mapped into the structures defined by the application model. This approach to validation essentially simulates the behavior of an application system interacting with the file or database system that provides data storage. Since an AP is used for data sharing, its performance must be validated against the data access requirements for the application.

The development and validation of an application model can be decomposed into the following seven high-level activities[7]. The first two activities establish the application context and construct application data models that will be tested. The next few activities, three through five, focus on the application model testing evaluate the correctness of the application models. Activity 6 controls the identification and resolution of issues against the model. These issues must be resolved to assure confidence in the model. Once the application models have been validated, the remaining AP components can be developed. A seventh activity defines conformance testing requirements on implementations of the AP. All resulting outputs which are required components of an AP are listed in *italics*. Some activities may be performed by separate groups of people.

I. Scoping the Application Context

The activity of "Scoping the Application Context" identifies a formal technical boundary for the application model by examining the functions of the application. The boundary is defined by analyzing the general processes the application performs using an activity model. The activity model, which illustrates the scope of the application area, is reviewed by experts in the area to ensure that it reflects common business practices. The scope and requirements guide the determination of what needs to be tested. The results of the activity follow:

- an *activity model represented in IDEF0*[8] which defines the application processes in the AP, and
- a *statement of scope and overall requirements*.

II. Model Construction

During the "Model Construction" activity detailed information models are constructed. Interviews with experts in the application area and reviews of comparable automated systems provide detailed information requirements and usage. These requirements are driven into a detailed information model which is called an Application Reference Model (ARM). Appropriate segments of the existing STEP integrated resource models are identified and interpreted to satisfy the application requirements as specified in the ARM [Danner92]. This interpretation process results in an Application Interpreted Model (AIM). The AIM supports the requirements of the ARM but is based on the information structures from the integrated resource models. The ARM is documented in one of the accepted information modeling formats and the AIM is to be provided in two formats, Express and Express-G [Palmer91, ISO11]. The following outputs are produced from this activity:

- the *ARM*, documented in terms familiar to an application domain expert,
- a formal and documented specification of the *AIM in Express*, and
- a graphical representation of the *AIM in Express-G.*

The interpretation process produces an application model that is specific to an application area and also consistent with other phases in a product life cycle. Both the ARM and the AIM are validated. The ARM validation ensures that requirements are valid and that the model can support them. The AIM validation ensures that the interpretation of the integrated resources is correct and that the interpreted resources can support the requirements.

III. Test Definition

The result of the "Test Definition" activity is a plan for validating the application model. This information is informative and guides the testing process but it is not computer-processible. The

[7] For a discussion which focuses on AP development and AP project planning, refer to Development Plan: Application protocols for Mechanical Parts Production [Stark91].

[8] A description of this technique can be found in Functional Modeling Manual (IDEF0), ICAM Architecture, Part II, Volume IV, Air Force Systems Command, Wright-Patterson Air Force Base, OH, February 1982.

test plan describes how typical users and systems within an application area use information to perform the activities described in the application activity model. Results from expert interviews and automated system reviews are synthesized into significant combinations of information that identify non-redundant and realistic test conditions, called test purposes, which are based on the usage requirements. Each test purpose is a data access request that needs to be satisfied using representative data during the validation process. Included in this activity is an identification of the types of and sources for data needed to conduct the tests. The test plan provides the organization to manage the complexity of the required tests.

There are three steps involved in the "Test Definition" activity. In the test planning step, what needs to be tested is decided. A test plan with test purposes for data usage is produced along with a product profile which describes unique characteristics of the product information. The product profile is used for gathering representative test data. In the next two steps, "create cross reference map" and "coverage analysis," additional test details are defined and industry-contributed product data is gathered and organized. Separate activities are not defined for these two steps because they do not generate new requirements for software tools, but they are critical steps in validating an application model.

The Test Definition activity produces two additional outputs which are used by the next activity: 1) a cross reference map, and 2) a coverage feedback report. The cross reference map indicates the correspondence between the application model and the representative test data. The creation of the cross reference map frequently uncovers major structural flaws in the application model. The coverage analysis of the representative test data reveals unused segments of the application model. If the AP project cannot identify data which corresponds to these segments, then the application model needs to change. Either the model was misunderstood and requires clarifying documentation; or additional searches for corresponding data are necessary. Ultimately, the unused segments are removed from the application model when their information requirements cannot be verified.

This activity produces four outputs:

- an overall test plan with usage test purposes,
- a product profile and the identification of representative test data meeting these criteria,
- a cross reference map to correlate the test data with the application model, and
- a report which describes issues and needed improvements to the application model.

IV. Test Case Data Generation

During the "Test Case Data Generation" activity, test case data is assembled or built from the contributed product data which has been selected by using the characteristics identified in the product profile. The objective is to identify where in the application model the representative product data will reside and if the information structures provided in the application model can accommodate it. Each piece of industry-contributed data should have a single, logical place in the model. In initial testing experiences [PDES90] much of the data was not available in electronic form so the test case data was prepared by hand. This was the most labor-intensive and error-prone activity of the entire process, but potentially the most reusable for conformance testing. Many deficiencies in the application model are uncovered by associating industry-contributed product data with the model's information structures.

The computer-processible output of this activity is the test data. This data directly supports the next activity, "Test Execution and Analysis." The process that makes that data available for test execution should not need further human intervention. Also in this activity, the test purposes for usage are fully detailed and documented as abstract test cases.[9] This activity results in the following output:

- detailed test data in a format suitable for processing by the VTS software, e.g. STEP exchange file format [ISO21],
- usage abstract test cases, and
- a report which describes issues and needed improvements to the application model.

V. Test Execution and Analysis

The "Test Execution and Analysis" activity involves the development, execution, and analysis

[9] The abstract test case and test purpose in conformance testing are related concepts but the intent is different. In validating an application model, the intent is to evaluate how well the model functions for supporting its intended scope. In conformance testing, the intent is to evaluate if an implementation supports all of the required features of a standard.

of the test cases against the application model. In order to execute the test cases, a computerized testing environment needs to be established and the test cases need to be formally specified with respect to the testing environment. Analysis of the executed test cases involves comparing the test results to the expected results to determine the validity of the application model. In addition, general statements about what any implementation of the AP must support are documented. This activity produces the following results:

- *validation test reports*,
- additional usage abstract test cases,
- improved test case data for reproducing test results,
- executable test cases for reproducing test results, and
- a report which describes issues and needed improvements to the application model.

VI. Model Refinement

The "Model Refinement" activity resolves issues uncovered during the testing process. Alternative solutions are proposed and the best solution is selected. Once there is agreement on how to resolve an issue, the model is modified and a new model is released for validation testing. The process of resolving an issue may replicate many of the preceding steps, e.g. the addition of an entity for resolving an issue might cause additional industry-contributed data to be gathered and new abstract test cases to be built. This activity results in the following information:

- the refined application model (*ARM or AIM*),
- an issue resolution statement describing the selected solution and supporting rationale, and
- refined test purposes, abstract test cases, and executable test cases.

Model validation testing is an iterative process. The end result of the process is an application model suitable for inclusion in a STEP AP. The model must be both useful and usable to be part of the standard. The involvement of a variety of application experts in the validation process helps to ensure that the model is useful. There should also be reviews by application experts who were not members of the AP project. To ensure that the model is usable, validation testing should be repeated until the model satisfactorily supports the information needs identified in the test plan.

The validation of an application model is dependent on the application area under consideration, but the validation process itself is constant and many aspects of it can be automated or supported by automation. Due to the nature of the standard being developed, it is mandatory that some parts of the process are automated. The standard will enable the automatic sharing of data. Therefore, the ability to automatically access data using the application model needs to be verified. Section 3 below discusses how this can be accomplished.

VII. Specification of Conformance Requirements

The "Specification of Conformance Requirements" activity is performed when there is confidence that the application model provides the functional capabilities that were specified in its scope. Conformance requirements specifiy all characteristics that must be satisfied by a conforming implementation of the AP. The three required components for an AP which fall within "Specification of Conformance Requirements" are:

- a *conformance clause* which specifies overall requirements for completeness and conformance,
- *test purposes* for conformance testing of vendor implementations,
- a *Protocol Inplementation Conformance Statement (PICS) proforma* is a checklist for identifying any optional characteristics of the AP which a vendor may claim to implement.

No further detail is provided since there are many unresolved issues relating to this topic (see Annex C of the "Guidelines for Development and Approval of STEP Application Protocols" [Palmer91] for further enumeration).

3 Automation for the Validation Testing Methodology

This section describes the automated dataflow within the VTS at the National PDES Testbed. The software, which supports the validation testing process, simulates the information access requirements for the application area being tested. The VTS software will provide a controlled environment for model validation testing, thereby reducing the potential for introducing errors into the process. The VTS software and the control of the supporting environment will also reduce the level of computer sophistication and interaction needed so that the users of the system will be able to concentrate on validating the application models.

The primary requirement of this process is the capability to manipulate and represent an application model and associated data for a variety of purposes. Therefore, many functional requirements [Morris91a] such as the ability to display and access the contents of models written in Express [ISO11] and the ability to manage the versions of documents and other files, are common among various activities. Some of these requirements such as word processing for preparation of documents, database access and persistent storage, and computer-aided design analysis of geometry, are not unique to STEP and are available in commercial systems. For these requirements, commercial systems will be used and integrated with the VTS software. Other requirements that are unique to STEP are either available from related projects or will be developed for the VTS.

Each activity of the validation testing process consumes and produces specifications or data. A subset of this material is directly processible and can be used to automate the activities. This automation parallels the flow of information between the activities described in Section 2. **Table 1** illustrates information inputs and outputs; the entries in **bold** represent the computer-processible portion of the information flow between the activities. The remainder of this section focuses on only those portions which are currently computer-processible. For a more general discussion of the information flow for model validation, see Mitchell [Mitch91].

The *Model Construction* activity produces application models in both human and computer-interpretable formats. Currently only the Express version of the application model is directly used as a basis for the software. The STEP integrated resource models are also represented in Express

	INPUT	**OUTPUT**
Scoping the Application Context	Application Requirements	Scope & Requirements Statement **Application Activity Model**
Model Construction	Scope & Requirements Statement **Application Activity Model** **Integrated Resource Models**	**Application Models (i.e. ARM, AIM)** Graphic Information Model
Test Definition	Application Requirements Scope & Requirements Statement **Application Activity Model** **Application Model in Express**	Test Plan with Test Purposes Product Profile Contributed Product Data Cross Reference Map Model Issues
Test Case Data Generation	Scope & Requirements Statement **Application Activity Model** **Application Model in Express** Product Profile Test Plan with Test Purposes Contributed Product Data (i.e. **IGES files**) Cross-Reference Map	**Test Case Data (i.e. STEP files)** Abstract Test Cases Model Issues
Test Execution and Analysis	Scope & Requirements Statement **Application Activity Model** **Application Model in Express** Test Plan with Test Purposes Abstract Test Cases **Test Case Data (i.e. STEP files)**	**Executable Test Cases** Validation Report Model Issues Refined Abstract Test Cases & **Test Case Data** Tool Enhancement Requirements
Model Refinement	Model Issues **Application Model in Express**	**Application Model in Express** Model Log with Resolutions
Conformance Requirements	**Application Model in Express** Abstract Test Cases **Test Case Data**	Conformance Clause Abstract Test Suite Model Issues

Table 1 Information Flow Between Model Development and Validation Activities

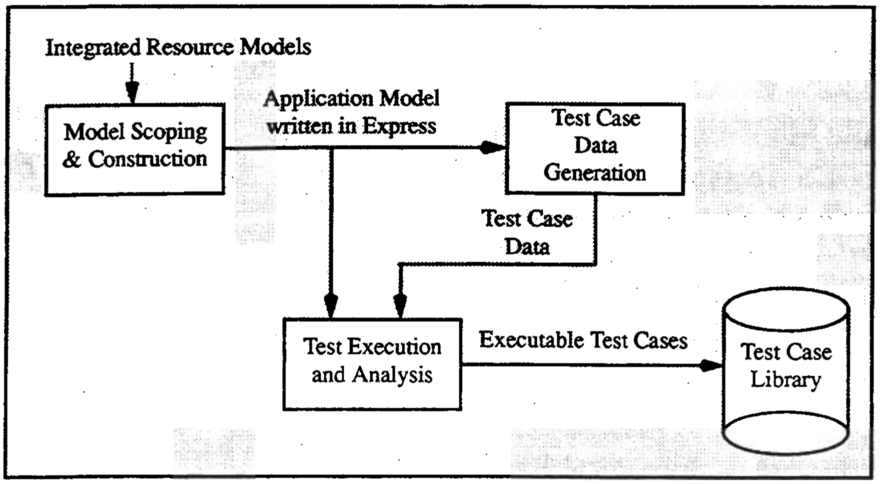

Figure 1 Interim System computer-processible data flow

and supply a basis for the application model being developed and validated. If additional computer-processible representations for these outputs were available, such as a test notation for abstract test cases, the amount of automation could be increased.

The *Test Definition* activity involves a great deal of human interaction, synthesis, and analysis. It is the least automatable activity in the process. The automation is limited to assistance in referencing the application and activity models and in the preparation of documentation. The work being done on conformance testing by the developers of STEP includes a formal test notation language that may increase the ability to provide additional automated assistance.

The primary automation for the *Test Case Data Generation* activity assists in preparing test data. The industry-contributed data is represented in many formats, but to be usable by validation testing, the test case data must be formed into a STEP exchange file format [ISO21] or loaded into a database system which has been built to manage STEP data structures. A reliable and efficient way to receive a limited portion of this data, principally geometric entities, is in the form of an Initial Graphics Exchange Specification (IGES) [IPO91] file extracted from a CAD system. IGES files can be translated to STEP exchange files [Breese91]. Additional contributed data needs to be prepared manually to complete the information required for the application model.

In the *Test Execution and Analysis* activity, executable test cases are generated, executed, and the results analyzed. This activity allows for a high degree of automation. The typical testing scenario follows:

1. the application model and test data are represented in a database;
2. executable test cases are specified in the database system's query language;
3. the queries which simulate typical application data access requirements are executed; and
4. the results are analyzed, issues against the application model are documented, and a validation report is generated.

The *Model Refinement* activity leads to a new application model and may contribute to refinements to the STEP integrated resource models. Throughout the validation testing process, any deficiencies in the application model, the test cases, or the test environment are documented, and appropriate enhancements are made. Appropriate steps in the validation process are repeated using the refined application model and test case data for any test purpose that was affected by these refinements.

The Conformance Requirements activity is still evolving. The current understanding of what should be accomplished in this activity is described in Palmer [Palmer91]. There is potential for reusing test case data, abstract test cases and VTS tools for these purposes. The intended purpose of this activity is to specify all of the requirements that a vendor implementation of an AP must satisfy. The VTS project efforts will evaluate these requirements when they become available.

4 Model Validation Testing at the National PDES Testbed

The National PDES Testbed has been used for the validation testing of STEP application models since 1989. The software currently in place at the National PDES Testbed [Breese91] provides some of the automation desired. An important point to make is that all of the software is being developed to generate a test enviroment for the schema under test. While the current software will support any data model developed in Express, knowledge of the other data modeling techniques used within STEP has contributed to the design. This section summarizes the direction for future improvements and additions to the validation testing software at the Testbed. This direction is based on past experiences with the software and STEP development methods, which helped clarify the needs for the validation testing software.

4.1 Experiences with the Interim System

The interim software employed in the validation testing of application models consists of a set of independent tools which operate in a variety of computer environments. The current method of sharing data in the testing process is by exchanging data files between these tools. This requires data translation and manual intervention, which introduces the potential for errors and inconsistencies, every time data is processed in the testing activities. Moreover, the process of importing and exporting between tools is time-consuming.

The automation for this testing process is currently provided by software tools which translate the application model and the test data among a number of formats [Clark90a,PDES91]. This software includes:

- an Express compiler and translators for representing the application model;
- an editor for STEP data [Clark90b] which structures the information as specified in Express for the application model;
- a relational database [Date90] which provides data access and storage management along with a query capability;
- a STEP exchange file parser and loaders for populating the STEP editor and database;
- Export facilities for extracting data from the editor or database into STEP exchange files;
- an IGES to STEP translator for converting digital product data from a Computer-Aided Design system to STEP; and
- a visualizer of geometric model data for display of a very limited set of STEP data.

The interim system has some of the needed functionality but provides unacceptable performance for some of the functions. The most significant improvements to the current system can be made by replacing the STEP editor and database management system with improved and integrated components. Both tools have suffered from significant performance problems with the large sets of complex data typical of engineering uses [PDES90].

In addition the STEP editor and database system currently use different data representation paradigms to represent the application model and its associated data. This places a burden on the users to understand the different representational formats and the relationship between these formats.

4.2 Future Directions for the Model Validation Software

The software needs for the STEP AP model validation process can be divided into two categories:

1. automation of the validation process by simulating the data access requirements of the application; and

2. automation to support the validation process, through assistance for preparing documentation and for referencing and browsing the application or integrated resource models.

The first category is a mandatory requirement for effective validation testing. The tests resulting from the process must be computer-processible and repeatable. These tests reflect the intended usage of the application model. Software for this purpose is the first priority for future implementation efforts.

The second category is partially met in the current system by word processing and drawing packages. These solutions provide limited support for these functions and leading to inefficient use of human resources. However, these automation needs will not be addressed until the first category is supported.

Two tools are most important for supporting the first category of automation, the simulation of data access based upon application information requirements:

1. a STEP data editor, and
2. a database system with a query capability.

Current efforts for the VTS software focus on developing an improved and integrated STEP data editor. This editor is being developed so that it will integrate with a database system. However, the editor will not depend on having a database system. The VTS software will provide an integrated set of functions which will provide a more effective and efficient environment and one more capable of simulating the data access needs of the application area. In summary the VTS software will make improvements over the interim software in the following areas:

- performance, in terms of both computation time and reliability;
- workflow automation to eliminate manual intervention where possible;
- more sophisticated support for editing of STEP data to reduce inconsistences in the data;
- error checking to reduce the potential for errors and improve error detection;
- expansion of functionality to address the needs of model scoping, model construction, and model refinement;
- provision of a single interface to the software, which will reduce the effort needed to learn the tools; and
- adoption of a data representation paradigm that resembles Express more closely than the relational paradigm.

When completed and tested, the VTS software and supporting documentation will be made available through the National PDES Testbed STEP On-line Information System [Katz91], as are the NIST developed portions of the interim system.[10]

[10] NIST provides absolutely no warranty. The NIST PDES Toolkit is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

4.3 Future Directions for AP Validation Methods

There are two areas where additional effort in defining validation methods are needed. These are:

1. validation techniques for the AP Conformance Testing requirements; and
2. definition of the relationship between AP model validation testing and the specification of AP conformance tests.

Summary

The validation testing of data models is needed to ensure technical solutions provided by the integrated model will work in a practical sense. An Application Protocol (AP) within STEP is a specification of data sharing requirements for a particular application area. Application Protocols are designed to permit practical implementations of STEP. The model validation focuses on the principal mechanism for specifying the data sharing requirements, an application specific data model. The body of the paper describes the process by which an application model is validated.

Application model development and validation are complex processes that rely extensively on human capabilities for analysis, judgement and synthesis of large amounts of diverse information. These activities require the support from automation to produce a technically complete AP. The model validation process and the STEP development methods place unique requirements on the software that will be needed to support the effective testing of STEP. The National PDES Testbed at the National Institute of Standards and Technology has undertaken a software project to support these activities. The current direction of this project has been formulated from our initial experiences in exercising this process and with software automation for the model validation testing. In addition, this paper introduces the potential contribution that application model validation and validation tools could make to the conformance testing of AP implementations.

References

[Breese91] Breese, J. N., McLay, M., and Silvernale, G., Validation Testing Laboratory User's Guide, NISTIR 4683, National Institute of Standards and Technology, Gaithersburg, MD, October 1991.

[Clark90a] Clark, S. N., An Introduction to the NIST PDES Toolkit, NISTIR 4336, National Institute of Standards and Technology, Gaithersburg, MD, May 1990.

[Clark90b] Clark, S. N., QDES User's Guide, NISTIR 4361, National Institute of Standards and Technology, Gaithersburg, MD, June 1990.

[Danner92] Danner, W. F. and Yang, Y., STEP Development Methods: Resource Integration and Application Interpretation, Working Draft, ISO TC184/SC4/WG5/N31, January 1992.

[Date90] Date, C. J., An Introduction to Database Systems: Volume 1, Fifth Edition, Addison-Wesley, 1990.

[IPO91] The Initial Graphics Exchange Specification (IGES), Version 5.1, IGES/PDES Organization, NCGA, Fairfax, VA, September 1991.

[ISO1] ISO 10303 Industrial Automation Systems -- Product Data Representation and Exchange -- Part 1: Overview and Fundamental Principles, Working Draft, Version 9, ISO TC184/SC4/WGPMAG/N50, Mason, H., ed., December 29, 1991.

[ISO11] ISO 10303 Industrial Automation Systems -- Product Data Representation and Exchange -- Part 11: Description Methods: The EXPRESS Language Reference Manual, Committee Draft, ISO TC184/SC4/WG5/N14, Spiby, P., ed., April 29, 1991.

[ISO21] ISO 10303 Industrial Automation Systems -- Product Data Representation and Exchange -- Part 21: Clear Text Encoding of the Exchange Structure, Committee Draft ISO TC184/SC4, Van Maanen, Jan, ed., March 12, 1991.

[ISO31] ISO 10303 Industrial Automation Systems -- Part 31: Conformance Testing Methodology and Framework: General Concepts . Working Draft, ISO TC184/SC4/WG6/N29, Owen J., ed., January 2, 1992.

[ISO41] ISO 10303 Industrial Automation Systems -- Product Data Representation and Exchange -- Part 41: Integrated Generic Resources: Fundamentals of Product Description and Support, Committee Draft, ISO TC184/SC4/WG4/N16, McKay, A., ed., September 6, 1991.

[Katz91] Katz, S., STEP On-Line Information Service User's Guide, NISTIR 4491, National Institute of Standards and Technology, January, 1991.

[Mitch90] Mitchell, M., Development Plan: Validation Testing System, NISTIR 4417, National Institute of Standards and Technology, September 1990.

[Mitch91] Mitchell, M., A Proposed Testing Methodology for STEP Application Protocol Validation, NISTIR 4684, National Institute of Standards and Technology, Gaithersburg, MD, September 1991.

[Morris91a] Morris, K. C., McLay, M., and Carr, P., Validation Testing System Requirements, NISTIR 4636, National Institute of Standards and Technology, Gaithersburg, MD, September 1991.

[Morris91b] Morris, K. C., Mitchell, M., and Sauder, D., Validating STEP Application Models at the National PDES Testbed, NISTIR 4735, National Institute of Standards and Technology, Gaithersburg, MD, December 1991.

[Morris91c] Morris, K. C., Architecture for the Validation Testing System Software, NISTIR 4742, National Institute of Standards and Technology, Gaithersburg, MD, December 1991.

[Palmer91] Palmer, M., Gilbert, M., and Foster,J., Guidelines for the Development and Approval of STEP Application Protocols, ISO TC184/SC4/WG4/ N15 Version 0.9 working draft, September 17, 1991.

[PDES90] Test Report for Context-Driven Integrated Model (CDIM) Application A1, Skeels, J., ed., PDES, Inc. internal report PMG012.01.00, SCRA, Charleston, SC, April 1990.

[PDES91] Block Point Release 2.3 System Manual, PDES, Inc. internal report PTI018.03.00, SCRA, Charleston, SC, August 1991.

[Stark91] Stark, C., and Mitchell, M., Development Plan: Application Protocol for Mechanical Parts Production, NISTIR 4628, National Institute of Standards and Technology, Gaithersburg, MD, July 1991.