U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

NISTIR 4417

National PDES Testbed
Report Series

Development Plan

Validation Testing System

NATIONAL PDES TESTBED

NIST

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology

NISTIR 4417

National PDES Testbed
Report Series

NATIONAL PDES TESTBED

Development Plan

Validation Testing System

Mary Mitchell

U.S. DEPARTMENT OF
COMMERCE
Robert A. Mosbacher,
Secretary of Commerce

National Institute of
Standards and Technology
John W. Lyons, Director

September 1990

DEPARTMENT OF COMMERCE • UNITED STATES OF AMERICA •

Preface

This document describes a plan to construct validation software tools which will be used to ensure the quality of STEP. The Validation Testing System is an integral part of an overall project to establish the National PDES Testbed at the National Institute of Standards and Technology (NIST). The Testbed was initiated in 1988 under the sponsorship of the U.S. Department of Defense Computer-aided Acquisition and Logistic Support (CALS) program. A major goal of the Testbed is to provide technical leadership in a national effort to implement a complete and useful specification for the exchange of product data. This specification must be designed to meet the needs of American industry and the CALS program.

The National PDES Testbed supports and actively participates in the international effort to develop the Standard for the Exchange of Product Model Data (STEP). The STEP development effort is led by the International Organization for Standardization (ISO) TC184/SC4. This document describes a plan to develop a validation testing system to serve the needs of both international and national efforts.

This plan describes one of several technical project threads that have been established for the National PDES Testbed. Other threads address such areas as:

- specification and testing of application protocols,
- construction of a prototype STEP-based manufacturing cell,
- development of configuration management systems and services,
- establishment of a product data exchange network, and
- development of conformance testing systems.

The level of support provided for these technical threads and others will be determined by sponsor needs and a number of different priorities. As such, the development plan contained within this document outlines a reasonable schedule to accomplish the objectives of the thread. Changes in priorities and levels of support may either accelerate or delay the proposed schedule. This plan will be updated periodically to reflect technical changes in the project, current level of effort, and expected continued support.

Charles R. McLean
CALS PDES Project Manager
NIST

Contents

Executive Summary

The Standard for the Exchange of Product Model Data (STEP) is a developing and rapidly evolving standard. Unlike most standards activities which start from baselines with proven implementations, Product Data Exchange using STEP (PDES) is a development activity which contains a dynamic set of concepts that are intended to support future computer integrated design and manufacturing applications. Major portions of STEP are yet to be completed and in some cases have not even been started. Given this dynamic nature it becomes critical to create quality assurance mechanisms that provide timely, substantiated technical feedback to the standards activities. The major goal of the Standards Testing Center (STC) of the National PDES Testbed is to ensure that the quality of the specification is high and that STEP will indeed work. When this goal is achieved STEP will be accepted as a standard, and industry and government will have what they need to exchange product data.

Thus, the success of STEP as a standard is a function of our ability to ensure that the standard is useful and properly used. Once an initial specification has been created, it must be validated, i.e., tested to determine that it meets the needs of the user community. The results and recommendations generated by validation testing must be fed back to the standards organizations for review and action. Standards committee members may then amend the specifications, affected portions may be re-tested, and the specifications can be approved as part of STEP.

The Standards Testing Center will develop the Validation Testing System (VTS) to provide techniques and systems to permit engineers to validate and ensure the internal consistency and usability of the specification. The VTS will be an integrated computing environment for evaluating the quality of the specification. In addition, the system will act as a repository for proof of the qualities that the STEP specification exhibits. This proof, in the form of test results and real-world test product data, will push the standardization process past the impasse of conflicting technical opinion and encourage implementations of information systems which use STEP.

The need for an automated validation testing system can be justified from our initial experiences in STEP validation. Through a joint effort with PDES, Inc., an industry consortium, a validation approach was defined in May 1989 [PDE89]. The approach improved the measurability of results by linking the testing to application usability. Experience with this approach has demonstrated feasibility and clarified the necessary procedures. In addition, the need for software automation in the validation process has been unquestionably demonstrated. PDES, Inc. in their Phase I project invested approximately 30 staff-years in validating portions of the STEP specification. Still, they were able to test only a small percentage of the STEP draft. The validation process needs full software automation to make more efficient use of the limited personnel available.

The Validation Testing System will provide software which: 1) automates the evaluation of the computable qualities, such as whether the syntax of the specification language was followed and 2) assists validation teams with solving complex problems which are not feasible to automate. Two major delivery points are planned. In January 1991, a software library for STEP structures and operations will be available. The library will be used as a building block for the remaining VTS development and distributed to other STEP-based software developers, such as PDES, Inc., who can contribute additional functionality or transfer STEP technology to industry. In December 1991, a set of integrated tools will be available for use in the validation process.

Validation testing priorities are being set in response to industry needs. Industry can influence which portions of STEP are proven first through participation in validation testing activities. This effort needs application experts and test data from industry. The STC welcomes participation in the testing program through the NIST Industrial Research Associate Program, through PDES, Inc. membership, or through participation in the ISO or IPO.

1 Goals and Objectives

The Validation Testing System (VTS) will play a major role in ensuring that the STEP specification is usable and able to support the sharing of product data between organizations and across multiple vendor information systems. STEP is a developing standard [SMI88] with numerous technical issues yet to be resolved [DAY89, OWE90]. The STEP specification is primarily a collection of data models which represent the proposed design for the exchange of product data.

The STEP validation process seeks to ensure that STEP has qualities such as completeness, lack of ambiguity, consistency, lack of overlap, and that it addresses required functionality [DAN90-2][ISO90-1]. Such qualities are difficult to measure. The intent of STEP to support requirements from multiple industries, e.g. mechanical, electrical, and architectural, complicates this measurement. The process for evaluating these qualities is complex, error prone and labor intensive [PDE89]. The usefulness of STEP is being ensured by proving traceability of application requirements to the functionality in STEP via Application Protocols (AP's) [PAL90].

The VTS integrated computing environment will assist those responsible for ensuring quality in the AP's and the underlying STEP specification prior to standardization. In addition, the VTS will be a repository for the proof of STEP usability. This proof, in the form of test results with traceability to both the AP under test with its underlying STEP models and real-world test product data, will push the standardization process past the impasse of conflicting technical opinion. The resulting VTS software will, in addition, encourage implementations of STEP-based application systems by providing reference implementations that use fundamental STEP methods such as the Express language [ISO90-3] and STEP exchange file format.

The VTS will provide software tools which: 1) automate the evaluation of the computable qualities, such as syntactic correctness, and 2) assist validation teams with solving complex problems which are not feasible to automate.

In satisfying the primary objective of proving that STEP is usable, a set of secondary objectives becomes clear. First, there is a need for an effective model validation test methodology. Existing test methods are not sufficient. The validation testing system must support the functions defined in this validation test methodology. Second, there is a need for the VTS test environment which supports model validation to be predictable and consistent. The VTS software must assure quality. Clear guidelines and procedures must be put in place to ensure its proper use. Otherwise, the VTS could corrupt the test results and cause them to be misinterpreted. Third, there is a need for a neutral computing facility where industrial partners can collaborate without the risk of infringing on each other's proprietary environments. Finally, future conformance testing activities should be able to utilize the software, procedures and testing materials from the model validation process.

2 Validation Testing System Overview

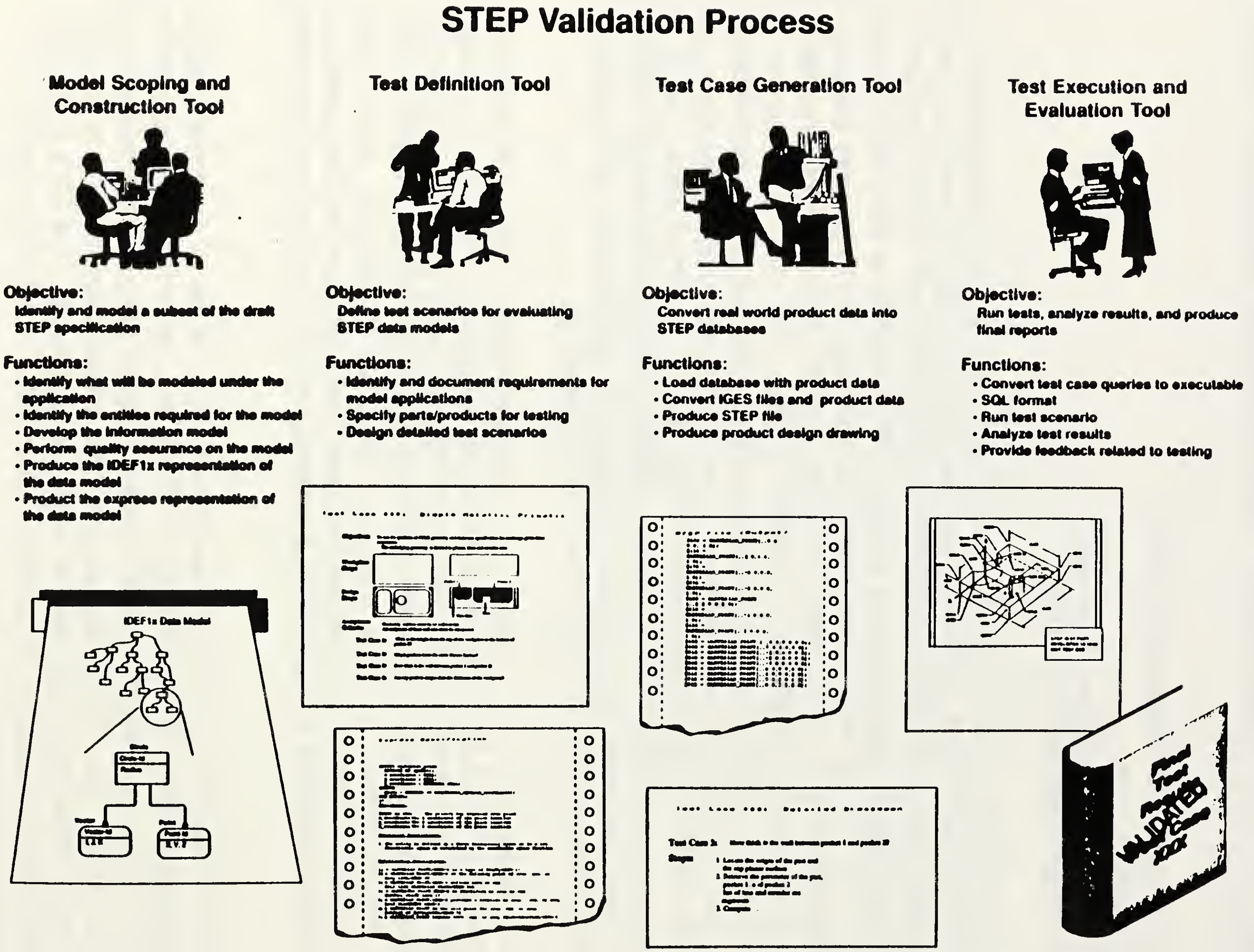
The design of the system is driven by the validation testing methodology and the opportunities for aiding the evaluation though software automation. The Validation Testing System (VTS) will consist of four integrated software tools which support STEP validation (Figure 1) via the development and evaluation of Application Protocols [PAL90]. The four tools are the:

- Model Scoping and Construction Tool,
- Test Definition Tool,
- Test Case Data Generation Tool, and
- Test Execution and Analysis Tool.

The system will provide a controlled environment which reduces the potential for misleading errors in the model validation process and reduces the need for trained technical personnel. Each tool will be designed to address the set of tasks required to accomplish a major stage in the validation process. The exception is the Model Scoping and Construction Tool which will service more than one step in the process because of overlapping requirements in the model scoping, model development and model improvement processes. The software modules in each tool will be integrated to eliminate manual intervention that can introduce errors into the testing environment. Each tool will have a narrow set of interfaces by which communication between other validation tools is achieved. All of the tools will rely on a generalized core of software to manage and manipulate STEP data structures. All tools which make up the validation system should have the same type of user interface to facilitate the interchange of validation testing personnel across major functional boundaries.

Some constraints impact the VTS design and implementation. First, the validation process evaluates conceptual data models. It cannot accomplish conformance testing of STEP-based application systems and it does not attempt to build prototype application systems. Existing software and conformance testing methods [HET89, ANS83, OSI88, ISO90-1] were not sufficient for this purpose. Improved techniques allow model validation to evaluate a larger set of usages but these methods must be fine tuned and documented. Model validation can feed testing materials to a future conformance testing program.

Second, the VTS will operate at multiple sites and on many types of computers. Model validation is being performed predominantly by PDES, Inc., an industry group working to accelerate STEP. Their test facility at SCRA could be used to reduce testbed resource contention. In the future, the VTS should support the broader community of all STEP Application Protocol developers. To ensure software portability, the VTS will utilize computer standards where possible. In particular, the POSIX, X Window system, C, C++ and SQL standards will be used.



STEP Validation Process

Model Scoping and Construction Tool

Objective:
Identify and model a subset of the draft STEP specification

Functions:
- Identify what will be modeled under the application
- Identify the entities required for the model
- Develop the information model
- Perform quality assurance on the model
- Produce the IDEF1x representation of the data model
- Product the express representation of the data model

Test Definition Tool

Objective:
Define test scenarios for evaluating STEP data models

Functions:
- Identify and document requirements for model applications
- Specify parts/products for testing
- Design detailed test scenarios

Test Case Generation Tool

Objective:
Convert real world product data into STEP databases

Functions:
- Load database with product data
- Convert IGES files and product data
- Produce STEP file
- Produce product design drawing

Test Execution and Evaluation Tool

Objective:
Run tests, analyze results, and produce final reports

Functions:
- Convert test case queries to executable
- SQL format
- Run test scenario
- Analyze test results
- Provide feedback related to testing

Figure 1. STEP Validation Process

Finally, there is a heavy reliance on input from application experts from industry during the modeling and testing definition stages of the validation process. The tools must be convenient for experts to use. A portable VTS micro computer is planned to support these processes.

2.1 Scenario: An Overview of the Validation Process

There are six activities involved in the validation process. These are: Scoping the Application Context, Model Construction, Test Definition, Test Case Data Generation, Test Execution and Evaluation, and Model Improvement. Process models which summarize these activities are found in Figures 2 and 3.

Scoping the Application Context

Activity Description - This activity identifies a formal technical boundary for the Application Protocol that will govern what portions of STEP will be validated and for what use. The application area is defined by modeling the general processes performed for the manner in which STEP will be used, e.g. exchange files or shared use. The information needs of the application processes are studied and categories are identified from the information used. These generalities about how classes of information are related to each other are formed into a planning data model. From this, it is clear what capabilities the application protocol is envisioned to support.

Process Outputs: 1) A process model, in IDEF0, for example, 2) a statement of the scope, 3) a planning data model, and 4) definitions of the processes and their interfaces.

Model Construction

Activity Description - In this activity, experts are interviewed to obtain detailed application information requirements and to understand any constraints on use. These requirements are used to drive the planning data model into a detailed conceptual data model called an Application Reference Model (ARM).

Segments of the STEP resource model specifications which meet these needs are identified along with any application specific restrictions and preferences. The result is a detailed conceptual data model called an Application Interpreted Model (AIM). (Figure 4) There are outstanding technical issues associated with AIM development. One serious issue is the inability of Express to support AIM requirements. Also, using STEP

Model Construction & Test Definition

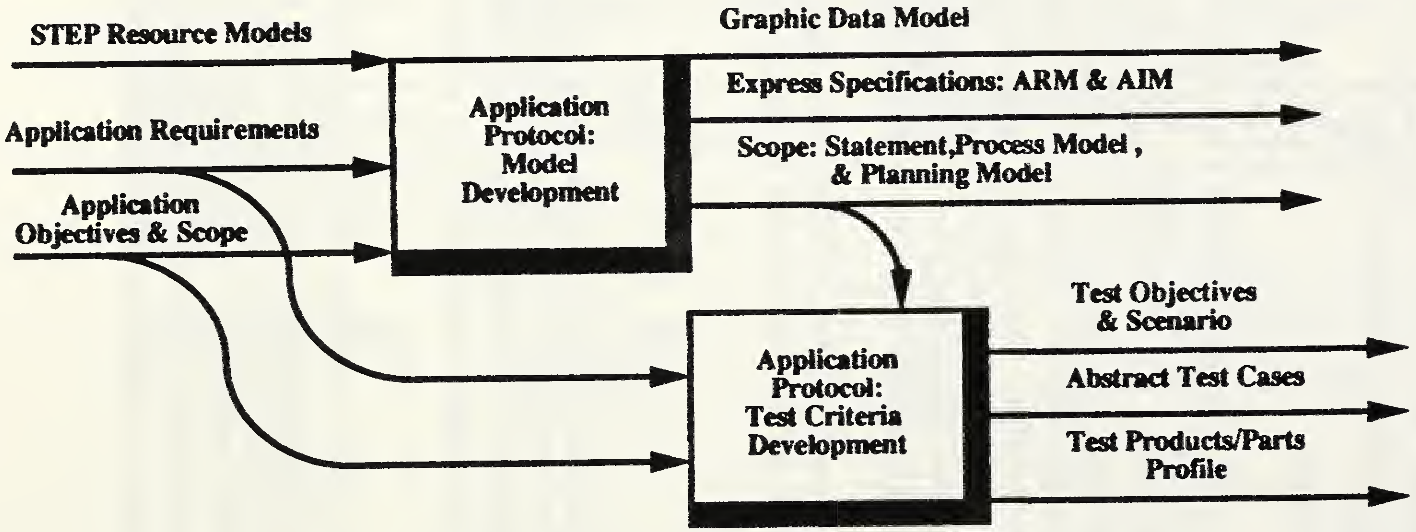

STEP Resource Models
Application Requirements
Application Objectives & Scope
Application Protocol: Model Development
Graphic Data Model
Express Specifications: ARM & AIM
Scope: Statement,Process Model , & Planning Model
Application Protocol: Test Criteria Development
Test Objectives & Scenario
Abstract Test Cases
Test Products/Parts Profile

Figure 2. Process Model of Model Construction and Test Definition

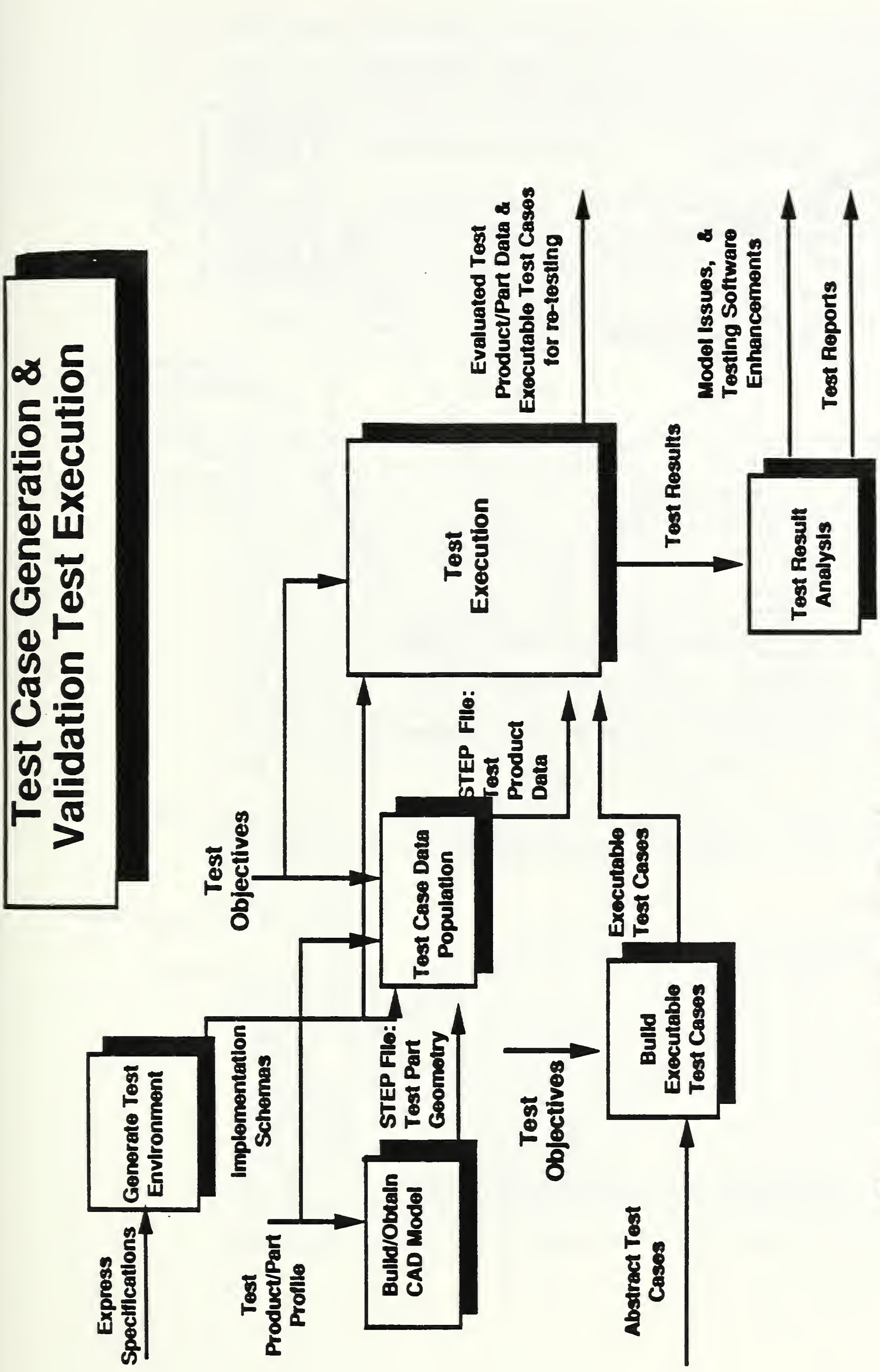

Test Case Generation &
Validation Test Execution
Express Specifications
Generate Test Environment
Implementation Schemas
Test Product/Part Profile
Build/Obtain CAD Model
STEP File: Test Part Geometry
Test Objectives
Test Case Data Population
STEP File: Test Product Data
Test Objectives
Abstract Test Cases
Build Executable Test Cases
Executable Test Cases
Test Execution
Evaluated Test Product/Part Data & Executable Test Cases for re-testing
Test Results
Test Result Analysis
Model Issues, & Testing Software Enhancements
Test Reports

Figure 3. Process Model of Test Case Generation and Test Execution

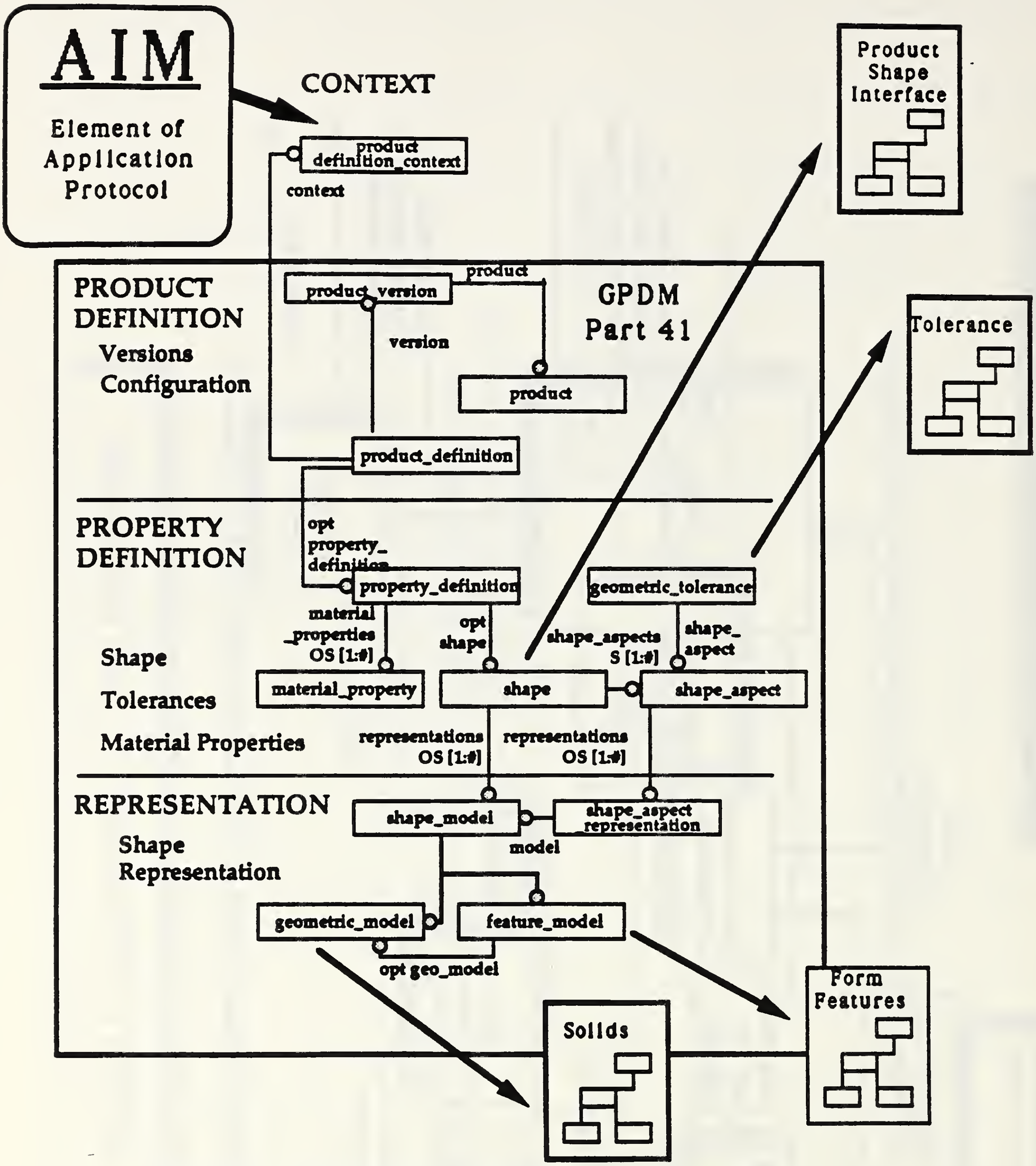


Figure 4. Application Interpreted Model (AIM)

for shared use instead of exchange (called a level 3 implementation), is likely to induce differences in the AP. Developing an AP which differs only by the intent to support shared use instead of file exchange would also prove or disprove that STEP is implementation independent [ISO90]. This is an unstudied open issue.

Process Outputs: 1) The ARM and AIM in the form of a graphic representation, such as IDEF1X, NIAM or Express G, 2) AIM and ARM formal specification in Express, 3) an example STEP file if the method of using STEP is exchange (no mapping exists for shared use), and 4) supporting documentation such as definitions and diagrams.

Test Definition

Activity Description - This activity defines testing requirements and designs the tests to evaluate the functionality of the AIM. There are generally many ways to accomplish an application function and the type of product may determine which techniques are used. The usage, the information required and the criteria used to judge if the information is sufficient are all affected. This stage consolidates expert interview results, identifies significant combinations, and organizes the details of what information is used, how it is used, and the expected results into realistic sequences called test scenarios. These scenarios are then compared and non-redundant, realistic test conditions are selected for development into test cases. Subject matter experts are critical to the process. The significant characteristics for test products or parts are also identified. (Figure 5)

Process Outputs: Test objectives and test scenarios, abstract test cases, usage constraints, expected results, and test product/part profiles.

Test Case Data Generation

Activity Description - This activity builds the product data for the product/part profile for a given test scenario. In the best case, a donated CAD description can be obtained from which an IGES file is extracted. This can provide up to 25% of the data but covers only a handful of STEP geometric representation entities. Our initial experience was that half of the test product data definitions were built from scratch. This is the most labor intensive, error prone but potentially most reusable activity in the entire process. The current approach is to: 1) use a CAD system to produce a geometric model, relying upon the CAD system to deliver geometrically consistent data, 2) extract into a STEP file format through

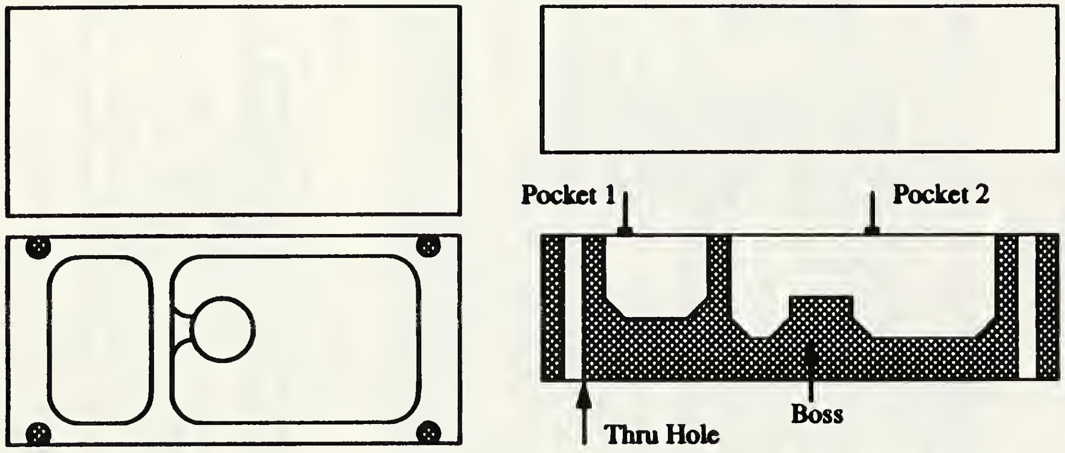
Test Scenario 1: Simple Metallic Prismatic

Example Objectives: To test the qualities of PDES geometry and materials specification for exchange given these restrictions:

The underlying geometry is limited to planes, lines and circular arcs.

Workpiece Shape

Design Shape



Example Acceptance Criteria:

Geometry entities must be an ordered list.
All endpoints of lines and arcs must be connected.

Abstract Test 1: What is the height from the top of the workpiece to the bottom of pocket 1?

Abstract Test 2: Find all pockets which have the same Corner Radius.

Abstract Test 3: How thick is the wall between pocket 1 and pocket 2?

Abstract Test 4: Are any pockets deeper than the thickness of the workpiece?

Figure 5. Sample Test Definition

an IGES translator (no use of solid geometry), 3) manually enter the remaining entities following the Express specification syntax and structure to produce complete Test Case Data in a STEP file. This last step is actually closely related to test execution. The most serious flaws in the conceptual data models are usually discovered and documented in this step.

Process Outputs: Test Product/Part Data in the form of STEP files developed to a specific AIM and for the purpose of satisfying the test objectives. Also, model issues and recommended VTS software enhancements will be documented.

Test Execution and Analysis

Activity Description - This activity executes the test cases, captures and analyzes the results, and develops issues against the AIM. In doing this, a test environment must be built for a specific AIM. Test case product data must be brought into this environment. The test case usage requirements must be converted into an executable form. What is actually tested needs to be compared to the model to determine if the model was actually validated adequately.

Process Outputs: 1) The executable test cases, 2) improved test product/part data for reproducing test results, 3) test reports, and 4) model issues which describe the needed improvements to the conceptual data models (ARM or AIM).

Model Refinement and Improvement

Activity Description - This activity receives issues on the ARM or the AIM that were developed during the test case generation or test execution and analysis activities. Alternative solutions are developed and a solution is selected. Once these issues have been resolved, the model is modified and a new model is released for testing purposes. The test case product data and executable tests must be modified to conform to the newly released model.

Process Outputs: 1) The refined AIM and ARM, 2) issue resolution statement describing the alternative solution selected and supporting rationale, 3) revisions to the example STEP file, and 4) refinements to test product/part data and executable test cases.

2.2 Relationship to Application Protocols

The products resulting from the validation process are major elements of an Application Protocol (AP). An AP describes a set of related processes that use product data. An AP consists of the following: 1) an Application Reference Model (ARM) that states explicitly the information needs for an application, 2) an Application Interpreted Model (AIM) which defines how these requirements are met by the STEP integrated data model, 3) a scope that describes what is covered by the application, 4) a usage guide which describes, unambiguously, how the information is to be exchanged, and 5) abstract test cases which include test objectives, abstract test assertions, acceptance criteria, test product definition data, and expected results.

AP's are critical because it is the AP which governs how the elements of STEP are used. Conformance criteria will be tied directly to AP's. The specifics of what is STEP compliant has not achieved consensus [ISO90-2]. The current model validation process is equivalent to what would be needed for an AP development process with two exceptions: 1) the model validation requirements which demonstrate usability would need to be accepted as conformance requirements, 2) while there is agreement that testing improves the quality, AP developers have not been required to perform even fitness testing or to develop executable test cases. The first AP will be completed before the AP development guidelines are completed. U.S. researchers recommend that the guidelines expand to include the model testing aspect of the validation process. Limited guidance is currently available to AP developers [PAL90, PDE89].

There is, however, acceptance that AP's have the responsibility for ensuring that STEP is workable. The STEP standards body, ISO TC184/SC4, has agreed that AP's are necessary and has dictated that the initial version of STEP will contain one or more AP's.

Further definition is needed on the model validation process and how it relates to AP development, on the outstanding technical issues, and on model validation process refinement opportunities. From the VTS perspective, software developers need this information to understand how their software will be used. Other organizations are developing software to accelerate the development of STEP; additional definition could focus these groups into collaborative development on software which could be used by the VTS.

Critical characteristics of AP's which must be resolved by the validation process include: 1) whether or not an AP defines the degree of functionality which all application systems must achieve and useful increments of functionality, 2) can or should AP's be nested or allowed to overlap, and 3) are there STEP architectural requirements by which all AP's must abide [DAN90, ISO90-4]. A precedent exists from another standard for the first issue only. The Graphical Kernel System (GKS), has defined levels of compliance with the levels based upon functionality [ANS85]. These compliance levels allow vendors to target application systems to the functionality desired by users and consumers to understand how different systems would fit together within or across application contexts. The GKS compliance structure is worthy of study for STEP AP's. Defining a set of AP compliance levels must occur after the AP is defined and requires the consideration of real-world application systems. The experience and the software developed or integrated into the VTS is needed and will contribute to solving these issues.

The STEP community recognizes and understands the model development activity for Application Protocols. There are, however, a number of technical issues which need resolution. The lack of testing experience has implications on AP development. AP developers need to be provided with the experiences from the validation teams in PDES, Inc. to properly estimate the time and effort required to produce a quality AP.

2.3 Proposed VTS Software Architecture

What can the VTS software realistically accomplish? The process of constructing and testing an AP is a thought intensive, detailed, and, currently, labor intensive process. Software tools have real potential to improve upon two classes of tasks: 1) error prone, complex tasks, and 2) repetitive, searching, sorting, and combining operations. The experience from PDES, Inc.'s model validation activities has produced an understanding of which aspects of the validation process are most able to be improved.

The strategy is to concentrate on the first class of tasks and then add functionality to reduce the effort spent on the second class of tasks. The greatest opportunity for improvement exists in the area of test case data generation and next in the area of test execution and analysis. The reality is that there are developments, particularly within PDES, Inc., but also within other research projects on foundation technologies [COL88, HAR89, ROB89] which are likely to deliver elements of the solution that are compatible with the VTS architecture and can be integrated into the VTS tools. Because of this, the plan for producing the VTS allocates time for assessing these developments and for the software integration. Many functional elements of the tools will not come from direct development by the VTS staff. The majority of the direct software development will be spent on core utilities and functions which will be used by all of the tools and shared with other STEP development activities at the earliest possible date.

Even though each stage of the validation process has different end products, there are many overlapping requirements for STEP-based software which may be shared by all. The plan is to re-use the common elements by customizing the user interface, organizing and packaging the software along validation process boundaries, and placing the software on the best platform for the operating conditions found during AP development. Elements of this overall architecture are shared by all VTS tools (Figure 6).

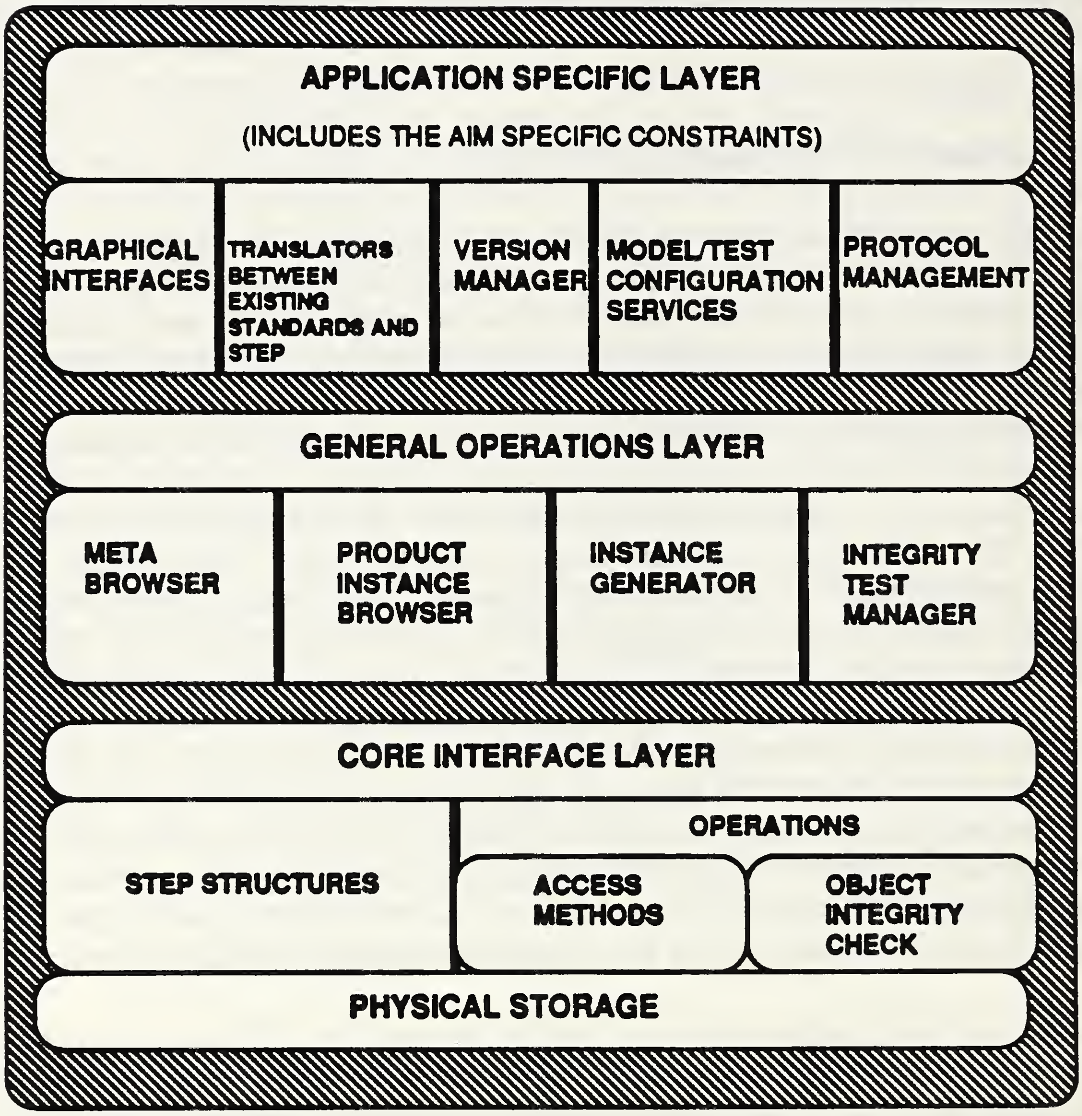


Figure 6. VTS Software Architecture

The VTS architecture is split into layers to insulate changes in the STEP schema from the software used to test the schema. The Core Interface Layer (CIL) supports STEP data structures and the most general access and storage operations. The General Operations Layer (GOL) provides for browsing and instance generating operations. Changes to the Express language, the STEP file mapping or encoding rules, and any new mapping for shared use should only affect the CIL. Changes to an Express schema should not affect either the core or the General Operations Layers. The Application Specific Layer (ASL) is the layer at which specific STEP utility applications exist.

The bottom layer of the architecture, the Core Interface Layer (CIL), consists of a library of C++ class definitions that are directly derived from a data model represented in the Express language [ISO90-2]. Each Express entity will be converted into a C++ class structure. Basic operations are defined to create, destroy, and modify instances of each type of structure. Data storage management tools manage the physical storage [MCL90].

In the middle layer, the General Operations Layer (GOL), are the general purpose browsers, data creation tools, and an integrity test manager which are designed to manipulate any of the objects from the CIL and to be independent of the contents of the objects. The STEP Data Access Interface Specification (SDAIS) is being defined in the voluntary standards community. Once this interface specification is available, it will be implemented in the GOL. The SDAIS will provide the basis for a single standardized interface for all STEP application system development. The general utilities of the GOL are used by tools from the Application Specific Layer through use of this interface specification.

The Application Specific Layer (ASL), the top layer, provides higher level functions such as graphical interfaces, version management, translators, and model/test configuration services. There are a number of specific STEP-based applications planned for development by the Application Prototype Center (APC) of the National PDES Testbed [FOW90] as well as external projects which can fit into the ASL. Two such tools which have utility are the IGES-to-PDES geometry translator and STEP geometry visualizer by PDES, Inc. The APC will be providing a solid geometry visualizer capability. The GOL provides a base on which application testing tools and prototype applications will be constructed.

Specific constraints that must be validated to prove the usefulness of an AP schema are conceptually located in protocol management modules of the ASL. These rules are not general to all instances of STEP product definitions, so they must be imposed locally for a specific AP. With the exception of the functionality needed to support these application protocol specific constraints, the predominant effort on the ASL will be software integration tasks.

2.4 Summary of VTS Tool Requirements

The CIL is the current priority. This will be the basis for the VTS and will be used in some of the PDES, Inc. software development. The Test Case Data Generation Tool is the top priority tool. Within the tool, the primary function supported is the creation and modification of STEP product definition data. This tool assists validation personnel in populating the standardized STEP data structures with realistic test data. The tool enforces the test data's consistency with the AIM which is undergoing validation. The major functional component of this tool is being called the STEP Data Probe. The requirements for it, as currently understood, will drive the GOL development of the instance generator, the meta-browser and the product instance browser. At a minimum, it must be capable of physical storage for STEP files. But the ability to support upward conversion of STEP files is highly desirable.

The Test Case Execution and Evaluation Tool is the next priority. It will drive further development of the CIL to support data access and the object integrity checking. The access method to the physical storage of STEP data must be able to be traversed to support evaluation of the usability of ARM or AIM to satisfy the AP test purposes. Persistent physical storage is required to eliminate the need to reload the test case data for every session. At the GOL, the integrity test manager which manages the consistency of entire STEP instances will be developed. It will rely upon the ASL for graphical interfaces, translator utilities and model/test configuration services.

The Model Scoping and Construction Tool and the Test Definition Tool will benefit from the development of the other systems. They are also technically less challenging. In addition, there are commercial products which achieve some of the known tool requirements and will be integrated into these tools. These tools will be implemented during the last part of the VTS development.

3 Development Plan

The Validation Testing System provides software tools to automate and assist in the model validation process. The overall plan and individual tasks respond to requirements derived from working with PDES, Inc. to apply this process. Some software was developed or borrowed from earlier projects for this trial. The usability of this software was adversely affected by the lack of a common software architecture, the lack of defined requirements, and scheduling demands. The VTS will support two groups: the PDES, Inc. teams responsible for model validation and ISO AP development projects. There is more risk associated with the latter group, because the AP methodology is not stable and these projects lack the testing experience that the PDES, Inc. teams have developed.

The Validation Testing System seeks to provide two types of capabilities: 1) software tools which automate evaluating the computable qualities, such as whether the syntax of the specification language was followed and 2) software tools which assist validation teams with solving complex problems which heretofore were not feasible to automate. Two major delivery points are planned. In January 1991, a library for STEP structures and operations, the STEP Core Class Library, will be available. The library will be used as a building block for the remaining VTS development and distributed to other STEP-based software developers, such as PDES, Inc., who can contribute additional functionality or transfer STEP technology to industry. In December 1991, the set of integrated tools will be available for use in the validation process. At this time, another distribution will occur which contains the Core Interface Layer and General Operations Layer software along with any non-proprietary software from the Application Specific Layer and installation instructions for the PDES Testbed platforms.

The software will only be alpha-release quality but with good user documentation. Expected users of these tools will be experienced computer users, but they are not expected to have a software development background.

Task Descriptions

VTS 1 **Develop VTS Technical Development Plan**

Develop the overall project plan for the Validation Testing System. This plan accomplishes this task.

VTS 2 **Formulate Validation Testing Methodology**

Document the existing Validation Testing Methodology. Develop a process model of the validation process and identify any improvement opportunities. Document open issues against the methodology and describe the potential impact, if any, on the Validation Testing System. Describe the outstanding AP related issues including those related to AIM development. Identify the risk factors due to the AP Development Methodology and their impact on the VTS design.

VTS 3 Develop Integration Architecture

Develop a high-level system description for the Validation Testing System. The system will have an open architecture, relying on open systems framework and standards to allow the major tools to fit together. The interfaces between each of the tools will be defined. The document describes how the elements of the VTS software architecture are organized into the four tools which comprise the system. It further describes any functional capabilities specific to a particular tool. This lays the foundation for negotiating compatibility between the Validation Testing System, the STEP Based Production Cell, future PDES, Inc. Block Point Release software, and potentially other STEP software developers.

VTS 4 Implement Core Functions and Utilities

Implement or integrate externally developed software which support the services and functions which will be used by multiple components of the Validation Testing System. These utilities will be responsible for the direct manipulations of the STEP implementation specifications, such as the STEP file structures, the Express language, etc. The end product will be a STEP C++ library and a software installation package.

VTS 5 Implement Model Scoping and Construction Tool

Select, acquire, customize and integrate commercial software with selected STEP Core Interface Layer (CIL) functions and utilities, and the General Operations Layer (GOL) Meta-Browser to support the Application Scoping, Model Construction, and Model Improvement activities of the validation process.

VTS 6 Implement Test Definition Tool

Select, acquire, customize and integrate commercial software with the GOL Meta-Browser and possibly some of the CIL functions and utilities to support the Test Definition activity of the validation process.

VTS 7 Implement Test Case Data Generation Tool

Customize CIL and GOL elements, implement and integrate externally developed software which supports the needs for Test Case Data Generation. Assess and acquire X Window development and user environments as well as libraries for engineering data operations. Within the tool, the primary function supported is the creation and modification of STEP product definition data. This tool assists validation personnel in populating the standardized STEP data structures with realistic test data. The tool enforces the test data's consistency with the AIM which is undergoing validation. The major functional component of this tool is called the STEP Data Probe.

VTS 8 Implement Test Case Execution and Evaluation Tool

Customize CIL and GOL elements, implement and integrate externally developed software which supports the needs for the Test Execution and Evaluation Tool. The access method to the physical storage of STEP data must be able to be traversed to support evaluating usability of ARM or AIM to satisfy the AP test purposes. This tool must also support the display of the geometric model representation.

VTS 9 Integrate Tools and Test Interfaces

Follow the validation process to exercise each of the tool interfaces. Develop test materials for this. Produce a final integration report describing any deficiencies, any lessons learned and a plan of action to correct these problems.

VTS 10 Port and Distribute Production-level VTS Software

In this task, the VTS software will be tested for portability to other computer types; packaged, distributed and installed at other testing sites; and delivered to a technology transfer agent.

VTS 10.1 Test portability

Test portability of software for the types of computers in the National PDES Testbed. Build installation procedures which support portable platform concepts. Develop installation instructions. Test portability and installation instructions by building the systems from scratch on the platforms supported.

VTS 10.2 Package and distribute

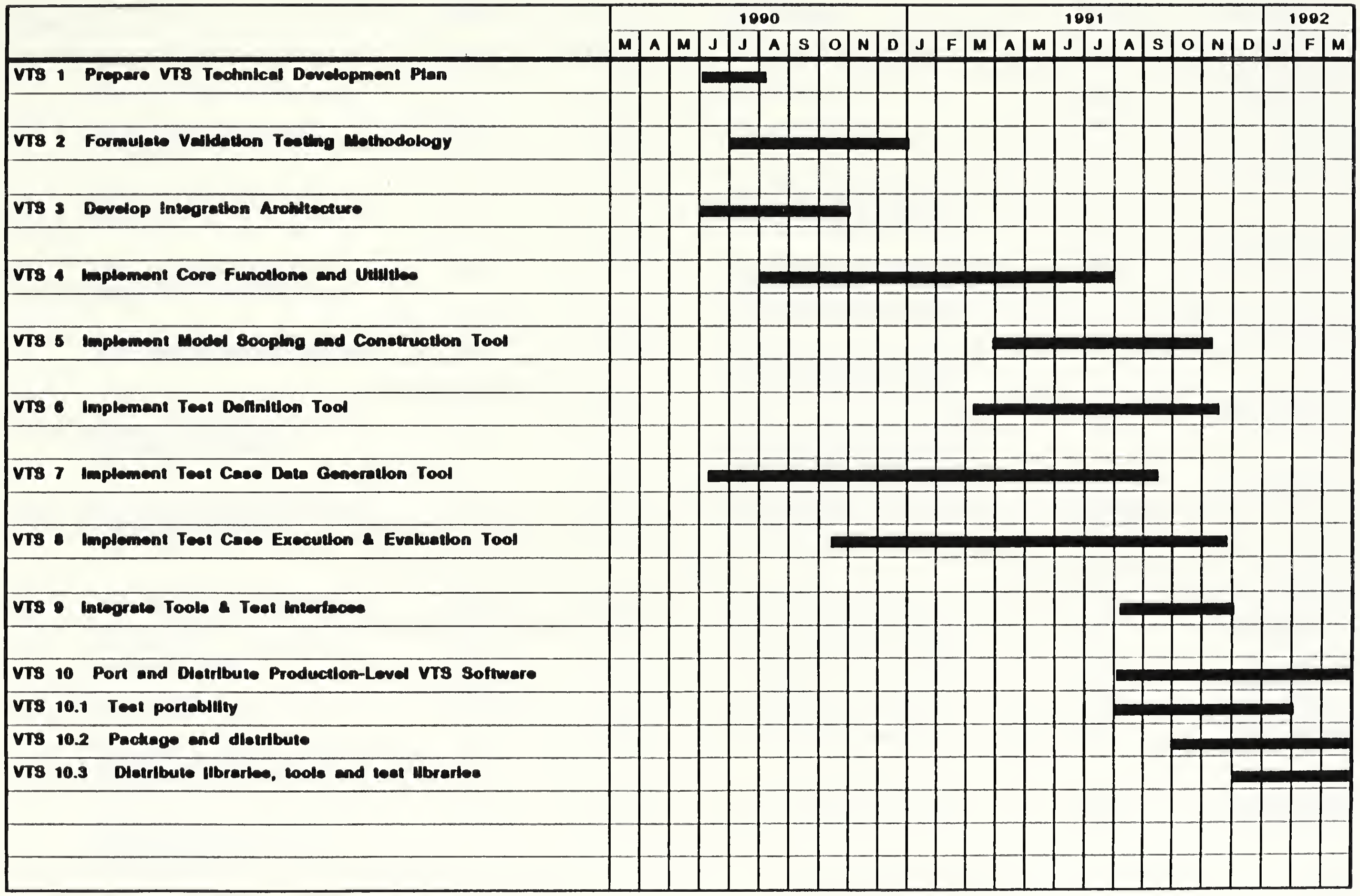
Package and distribute STEP libraries and integrated tools with supporting documentation to PDES, Inc., other testing sites, and external STEP developers.

VTS 10.3 Distribute libraries, tools and test libraries

Distribute public domain utilities, development libraries and test libraries to National Technical Information Service (NTIS) or another technology transfer agent.

Validation Testing System Plan

Project Schedule



DELIVERABLES: VALIDATION TESTING SYSTEM

TASK		START	STOP	DELIVERABLE	SUPPLIER	CUSTOMER
1.	Prepare technical development plan	Jun 90	Aug 90	Technical Development Plan	STC	CALS/NIST/Ex STEP
2.	Define validation test methodology	Jul 90	Dec 90	Validation Method Doc	STC	CALS/NIST/PDES,Inc.
3.	Develop integration validation architecture	Jun 90	Oct 90	Integration Arch Doc	STC	NIST/PDES,Inc./Ex STEP
4.	Implement Core Functions and Utilities	Aug 90	Jul 91	Core Software Installation Package Requirements & Design Doc System & User Documentation	STC	NIST/PDES,Inc./APC/PCO
5.	Implement Model Scoping and Construction Tool (MSC)	Apr 91	Nov 91	MSC Software Installation Package Requirements & Design Document System & User Documentation	STC	NIST/PDES, Inc./CALS
6.	Implement Test Definition Tool (TD)	Mar91	Nov 91	TD Software Installation Package Requirements & Design Document System & User Documentation	STC	NIST/PDES,Inc./CALS
7.	Implement Test Case Data Generation Tool (TCDG)	Jun 90	Sep 91	TCDG Software Installation Pkg Requirements & Design Document System & User Documentation	STC	NIST/PDES,Inc.
8.	Implement Test Case Execution and Evaluation Tool (TCEES)	Oct 90	Nov 91	TCEE Software Installation Pkg Requirements & Design Document System & User Documentation	STC (APC,ISC & PDES, Inc.)	NIST/PDES,Inc.
9.	Integration tools and test interfaces	Aug 91	Dec 91	VTS Interface Test Report	STC	NIST/PDES,Inc.
10.	Production-level validation testing capability available	Aug 91	Mar 92	Software Distribution Package	STC	NIST/PDES,Inc.
10.1	Test portability	Aug 91	Nov 91	Installation & Test Procedures	STC	NIST/PDES,Inc.
10.2	Install in National PDES Testbed and Distribute to PDES, Inc. & other AP developers	Oct 91	Nov 91	VTS Installation Test Report	STC	NIST/PDES,Inc./PCO
10.3	Distribute of public domain utilities, development libraries and test libraries to NTIS	Dec 91	Mar 92	Software Distribution Package	STC	NIST/PDES,Inc./PCO

4 Resources

4.1 Personnel

The following combination of skills is required for the software development team. This projection was used in developing the schedule. Of this, approximately three and one-half staff have been identified from existing resources; currently, a number of the most critical positions need staffing.

- Project Manager
- Analyst, Lead Designer
- System Integrator, POSIX environment
- Applications Programmers for C, C++, Database, CASE (3)
- Software Test Specialist, Configuration Librarian
- POSIX, Utilities, System Programmer
- X Window, User Interface Programmer (half-time)
- System Integrator, DOS environment (half-time)
- CAD/Mathematical Programmer (half-time)

In addition to these positions, some time will need to be negotiated for services from a Technical Editor and a Validation Testing System User on an as needed basis.

Standards Testing Center Manager

Responsible for the overall structure and planning for the STC. Develops work plans, defines tasks and builds and monitor schedules. Defines functions, roles, and skills required for tasks. Negotiates for staff resources and system support. Identifies priorities requirements with users. Acquires equipment and commercially available software for the VTS. Coordinates with PDES, Inc. and the other National PDES Testbed Centers for new or enhanced tools. Develops procedures for development, use and support of the VTS. Directs the VTS software architecture development. Experience in conceptual modeling, the validation and application protocol methodologies, software testing, the STEP framework. Possesses interpersonal, verbal and written communication skills, as well as project planning, staff management, software integration and system design experience.

Analyst, Lead Designer

Performs analysis of the functional requirements of the model validation and AP development process, and designs system architecture, software modules and software interfaces for the VTS. Experience in conceptual modeling, Express, IDEF, validation testing process, structured design techniques and object oriented design techniques. Possesses interpersonal, written and verbal communication skills.

System Integrator, POSIX environment

Coordinates VTS software releases and develops installation procedures. Develops utilities and procedures to ensure portability. Ports software to VTS platform environments. Develops system interface specifications. Experience in (1) the validation testing process, Unix Operating System, Unix utilities, X Windows, and C; and (2) merging software from several sources into a reliable end product. Possesses software integration, software engineering, written and verbal communication skills.

Applications Programmers - C, C++, Database, CASE

Develops procedures and implements utilities in C, C++ and database languages for automating validation testing. Integrates software developed by other project team members into the deliverable tools. Maintains existing code on several computer platforms (updates software to work with new operating system versions, to work with new compilers, and to work with new STEP methods versions). Experience with (1) C, C++, debuggers, X Windows, Unix, and SQL; and (2) higher level programming concepts such as object oriented design. Experience in merging software from several sources into a reliable end product.

Software Testing Specialist, Configuration Librarian

Develops software test plans and procedures. Prepares test data. Develops test reports and model issues. Reviews system and user documentation. Performs software testing, captures test results, classifies and directs problems to the appropriate development staff. Experience in CAD systems, software testing, configuration control concepts, and UNIX tools. Application expertise in CAD systems and relational DBMS experience.

POSIX, Utilities, Systems Programmer

Builds STEP generic user and developer environments. Installs and coordinates the introduction or upgrade of equipment and software packages. Develops procedures and implements utilities for automating validation testing, establishes network connectivity and transfers between VTS Tools, and enhancing the STEP developer's programming environment. Performs VTS facility planning. Experience with (1) Unix, Unix shell, C, Unix utilities, networking protocols, and computer equipment; and (2) systems programming and systems administration experience for applicable systems.

X Windows, User Interface Programmer

Develops utilities in the X Window environment. Builds tools on top of existing public domain class libraries. Develops X code for a general user interface to edit and browse STEP data. Supplies application programmers with general code to implement X-based tools. Extends the X protocol to directly include STEP specific data constructs. Installs and customizes X protocol for remote access to tools. Experience with X Windows, C, and C++, Unix, debuggers. Experience with higher level programming concepts such as object oriented design. Experience in merging software from several sources into a reliable end product.

System Integrator, DOS environment

Coordinates VTS software releases and develops installation procedures. Customizes commercial products and develops interfaces to integrate products and software. Ports already developed code into a DOS environment. Experience with the validation testing process, DOS OS, PC utilities, X Windows, and C. Experience in merging software from several sources into a reliable end product, software engineering, relational database, written and verbal communication skills.

CAD/Mathematical Programmer

Defines methods for testing the mathematical correctness of rules defined in STEP, particularly for geometric representation. Develops software for application programmers to use in solving detailed mathematical and geometric problems. Examines the mathematical basis for STEP entity representations in an object oriented programming environment. Develops test data and examples for validation purposes. Experience in mathematics, CAD/CAM systems, C, and C++. Experience with higher level programming concepts such as object oriented design.

Validation User

Provides requirements for functional capabilities. Identifies priorities needs and communicates scheduling requirements. Reviews software designs, user documentation, and the software test plan for adequacy. Experience in performing STEP validation, a knowledge of STEP development methods, and a knowledge of the model validation and AP development methodology.

4.2 Commercial Software

The project has been successful at obtaining software donations and loans. Relatively small expenditures are expected. Training and manual costs are expected to equal direct software expenditures.

- X Window (OSF compliant)
- C++ and C compilers (ANSI Standard compliant)
- CAD package with solids, feature-based, with IGES (loaned)
- Relational and object oriented database packages (donated)
- CASE Tools (PC based, anticipate a donation)
- Document preparation

- Electronic mail system
- Software for configuration management

4.3 Equipment

A considerable amount of equipment exists in the National PDES Testbed, both from purchases and from equipment donations and loans. Most of the additional equipment described below fits specialized needs and would greatly improve the effectiveness of the Validation Testing System:

- Compute Server, dedicated to remote computing, two needed, high-end POSIX Workstation with 3 gigabytes storage each for validation users.

- Development Server, dedicated to remote computing, one exists, probable upgrade to high-end POSIX Workstation.

- Development System, upgrade existing system to high-end POSIX Workstation with local disk

- Developer Systems, low-end POSIX Workstations, 5 exist

- Test System, low-end POSIX Workstation with local disk, 1 exists

- CAD/Graphical Workstations, 3 exist (loaned)

- Validation Workstations, high-end POSIX Workstation, 2 exist (loaned)

- SLIP protocol device, one for every three validation teams, for remote X Window access.

- X Window terminals, 3 for peak validation/development usage, (could use low-end POSIX Workstations).

- Development System, high-end PC.

- High speed modem bank, 8 exist in-bound, 2 exist out-bound, (shared resource, negotiated use).

- High speed modems, 4 external for dial-in, off-site use and to be pressed into service if capacity of modem bank becomes saturated.

- Display system to attach to a PC or X Window terminal for training, demonstrations, and validation team working sessions.

4.4 Facilities

The space for guest research staff from PDES, Inc. and other sources has been allocated for Validation Testing purposes. It has been outfitted over the past year. There are only a few items which are still required:

- Validation testing user laboratory space for two testing teams (twelve seats), exists and is in use.

- Validation meeting space for modeling and testing teams (twelve seats), exists and is in use.

- Asynchronous communication lines for modem connections, 5 lines, exists and are in use.

- Photocopier and FAX machine - necessary due to the distance between the laboratory facility and division support services.

5 References

[ANS83] "Software Engineering Standards: ANSI/IEEE Standard 729-1983, Glossary of Software Engineering Terminology," The Institute of Electrical and Electronics Engineers, Inc., 1984.

[ANS85] "Computer Graphics: Graphical Kernel System (GKS) Functional Description," ANSI X3.124-1985, American National Standards Institute, 1985.

[COL88] Colborn, K., "OSF Determines User Interface; Choices Could Affect the Development of Applications Software," EDN, December, 1988.

[DAN90-1] Danner, W.,"A Proposed Integration Framework for STEP (Standard for the Exchange of Product Model Data)," National Institute of Standards and Technology, NISTIR 90-4295, 1990.

[DAN90-2] Danner, W., Nieva, A., Shaw, N. and Yang, Y. "Proposed Revision of the Requirements Document for STEP," Working Draft, National Institute of Standards and Technology, submitted to TC184/SC4/PMAG, August 1990.

[DAY88] Day, T. and Lopatka, R., "Barriers to PDES Approval," United Technology Research Center, Internal Report, Hartford, CN, March, 1988.

[FIPS88] "Conformance Testing Policy and Procedures," Draft Federal Information Processing Standard, National Institute of Standards and Technology, 1988.

[FOW90] Fowler, J., "STEP Production Cell," National PDES Testbed Development Plan, National Institute of Standards and Technology, NISTIR 4421, 1990.

[HAR89] Harrison, W., Shilling, J. and Sweeney, "Good News, Bad News: Experience Building a Software Development Environment Using the Object-Oriented Paradigm," OOPSLA'89 Proceedings, New Orleans, LA, October 1989, pp. 85-94.

[HEN89] Henghold, W., Schumaker, J., and Baker, L., "Considerations for the Development and Implementation of PDES within a Government Environment," Manufacturing Technology Directorate, Wright Aeronautical Laboratory, Air Force Systems Command, Wright-Patterson AFB, OH, AFWAL-TR-89-8009, 1989.

[HET88] Hetzel, B., "The Complete Guide to Software Testing," Second Edition, QED Information Sciences, Inc., Wellesley, MA, 1988.

[ISO90-1] International Organization for Standardization, Working Draft 10303-31, "Conformance Testing Methodology and Framework; General Concepts," ISO TC184/SC4/WG6 Document Number N5, July 1990.

[ISO90-2] International Organization for Standardization, "Overview and Fundamental Principles," Working Draft 10303-1, ISO TC184/SC4/WG6 N6, August 1990.

[ISO90-3] International Organization for Standardization, Working Draft 10303-11, "Express Language Reference Manual," ISO TC184/SC4/WG1, Document Number N496, May 1990.

[ISO90-4] International Organization for Standardization, Working Draft 10303-41, "Generic Product Data Model," ISO TC184/SC4/WG4, Document Number N1, Sept 1990.

[MCL90] McLay, M., and Morris, K., "The NIST STEP Class Library," National Institute of Standards and Technology, NISTIR 4411, 1990.

[OSI88] "OSI Conformance Testing Methodology and Framework; Part 1: General Concepts," ISO Draft International Standard 9646-1, 1988.

[OWE90] Owen, J., and Woodall, A., "Application Protocols: current issues raised by the prototype AP exercise," ISO TC184/SC4/WG1, Internal Technical Report, Document Number N495, May 1990.

[PAL90] Palmer, M., "Guidelines for the Development and Approval of Application Protocols," Working Draft, Version 0.3 ISO TC184/SC4/WG4, August 1990.

[PDE89] "Test Criteria Requirements Analysis Project Final Report," PDES, Inc. Internal Report, South Carolina Research Authority, Charleston, SC, May 1989.

[ROB89] Robb, T., "A Relational Algebra Package for C++," Proceedings of C++ at Work'89, Tungsboro, MA, November 1989.

[SMI88] Smith, B., and Rinaudot, G., "Product Data Exchange Specification: First Working Draft," NISTIR 88-4004, National Technical Information Service (NTIS), Order PB 89-144794, 1988.

6 Glossary

Abstract Test Assertion

The portion of an abstract test case which is a logical expression specifying a state that must exist to satisfy an application purpose.

Abstract Test Case

A logical expression specifying a state that must exist to satisfy an application purpose in human readable form, which provides basis to develop machine executable tests.

Acceptance Criteria

The part of an abstract test case which specifies the set of conditions that must be satisfied in the test result to achieve the predicted test outcome.

Application

A specific function or work area that contributes to creating product definition data and/or finished product deliverables to meet an industrial need. The nature of an application depends on several factors, one of which is discipline, such as electrical, mechanical, etc.

Application Interpreted Model (AIM)

A conceptual data model that describes the STEP standardized data constructs required for functional equivalence to the application protocol's application reference model.

Application Protocol (AP)

A method which defines the context for the use of product data for the purpose of achieving consistent and reliable exchange and specifies this use of the standard in a application context to satisfy an industrial need.

Application Reference Model (ARM)

An conceptual data model that formally describes the information requirements, including structural requirements, and constraints for an application area. The model uses application specific terminology and rules familiar to an expert from the application area. The model should be independent of any physical implementation and must be validated by experts from the application area.

Conceptual Data Model

A description of the information requirements, relationships between informational objects, information structure, and constraints for a subject area.

Conformance Requirements

A description of the conformance criteria and test purposes designed to verify if an implemented system complies with a standard.

Executable Test Case

A machine executable test which results from transforming an abstract test case into a computable form so that reproducible test results can be produced.

Expected Result

A foreseen test outcome.

Fitness Test

The review and walk-through by technical experts of an application reference model which demonstrates that the model is useful in the context of a particular application area.

Integrated Data Model

A conceptual data model which represents the assembly of multiple data models into a coherent, non-redundant, and internally consistent model. An integrated model is characterized by an level of uniform detail and abstraction in addition to addressing the specified scope.

Scope and Requirements

A statement of application domain information needs that must be accommodated to achieve meaning and useful information exchange.

Semantics

The meaning that is assigned to an item of information.

Usage Guide

Information on employing an application protocol for a particular manner of using STEP, such as exchange or shared use.

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4417

2. PERFORMING ORGANIZATION REPORT NUMBER
734.02

3. PUBLICATION DATE
09/90

4. TITLE AND SUBTITLE
Testbed Development Plan
Validation and Testing System Development

5. AUTHOR(S)
Mitchell, Mary J.

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

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

10. SUPPLEMENTARY NOTES

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 describes the plan for the development of the Validation Testing System (VTS) which will provide techniques and systems to engineers for validating and ensuring the internal consistency of the STEP specification. Unlike most standards activities which start from baselines with proven feasibility, Product Data Exchange using STEP (PDES) is a development activity, containing a dynamic set of concepts which have not stabilized. Major portions of STEP are yet to be completed and in some cases haven't been started. Given this dynamic nature it becomes critical to create quality assurance mechanisms that can provide technical feedback to standards activities. The major goal of the Standards Testing Center (STC) of the National PDES Testbed is to ensure that the quality of the specification is high and that PDES will indeed work. The VTS will be an integrated computing environment for evaluating the quality of the specification.

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

Application Protocol; Quality of Data Models; PDES; STEP; Testing Software; Validation

13. AVAILABILITY

☒ UNLIMITED

□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
37

15. PRICE
A03

ELECTRONIC FORM