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

NISTIR 4538

National PDES Testbed
Report Series

Configuration Management Concepts Document

NATIONAL
PDES
TESTBED

NIST

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

NISTIR 4538

National PDES Testbed Report Series

Configuration Management Concepts Document

Susan Katz

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

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

April 26, 1991

Disclaimer: Certain commercial products are identified in this paper in order to specify adequately the configuration management procedures. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products identified are necessarily the best available for the purpose.

Contents

1 Introduction 1
2 Background 1
2.1 Definition of CM 1
2.2 The STEP Development Effort 1
2.3 Mission of the CMS (Configuration Management System) 3
3 Configuration Management Concepts 4
3.1 How CM Solves Common Development Problems 4
3.1.1 Identification of Controlled Items 4
3.1.2 Centralized Control 5
3.1.3 Version Identification and Status Accounting 5
3.1.4 Promotion 6
3.1.5 Change Control 7
3.1.6 Logical Grouping of Controlled Items 7
3.1.7 Visibility: CM's support of Quality Assurance 7
3.2 Characteristics of a Good CM System 9
3.2.1 Ease of Use 9
3.2.2 Reliability 9
3.2.3 Security 9
3.2.4 Accessibility 9
4 Particular Implementations of CM on the STEP Project 10
4.1 Configuration Management of Documents 10
4.2 Configuration Management of Software 13
4.2.1 How the testers use the combination of Testbed tools 13
4.2.2 The software tools as functional components of the Toolkit 16
4.2.3 CM in the Testbed Environment 16
4.3 Configuration Management of STEP Data 17
5 Remote Access to the CMS 18
6 Implementation of the CMS 20
7 References 21
8 Acronyms 23
9 Definitions 24

Appendix A. An Express Conceptual Model and Its Instances 25

The work described in this document was funded by the U.S. Government's Department of Defense Computer-aided Acquisition and Logistic Support (CALS) program and is not subject to copyright.

This page intentionally left blank.

List of Figures

Figure 1 The Growth of a Software Problem 8
Figure 2 STEP Documents Configuration Management Flow (proposed) 12
Figure 3 Building and Testing STEP Part Data Tables Using the Toolkit Software 15
Figure 4 Functional View of the Configuration Management System 19

This page intentionally left blank.

1 Introduction

The purpose of this document is to establish Configuration Management (CM) concepts to be applied in support of the development of the Standard for the Exchange of Product Model Data (informally called STEP). Configuration Management services are to be provided for four organizations involved in the development of the international standard: the International Organization for Standardization (ISO), the IGES/PDES Organization (IPO), PDES, Inc. (a consortium of private-sector companies) and the National PDES Testbed.

Definitions of the acronyms used in this document can be found in Section 8, on page 23. Definitions of CM-specific terms can be found in Section 9, on page 24.

2 Background

"Configuration Management Systems and Services" is one of the major technical threads of the National PDES Testbed project [McLean]. This section explains what configuration management is, and why it is needed in the STEP development effort.

2.1 Definition of CM

Configuration management is the management of change. It is a formal discipline which provides methods and tools *a*) to identify components, versions and baselines of selected items, and *b*) to control changes to those items. CM provides a method for logically grouping related components throughout the various stages of product development. It also provides visibility and traceability for the evolving status of each item. An effective CM system thus identifies, controls, records, and reports on any functional, physical or status changes to the controlled items. A detailed discussion of CM issues is provided in section 3, on page 4 of this document.

2.2 The STEP Development Effort

The International Organization for Standardization (ISO) is currently developing a new international standard for the exchange of information related to automated manufacturing. The developing standard is informally called the Standard for the Exchange of Product Model Data (STEP).

STEP is divided into Parts (or volumes) which pertain to different technical areas. The development effort includes: writing the Parts, developing the software to support testing, testing the concepts contained in the Parts, and revising the Parts to make them technically correct and consistent with each other.

Several United States based organizations are also participating in various aspects of the development of this standard. These include the National PDES (Product Data Exchange using STEP) Testbed Project at the National Institute of Standards and Technology (NIST), the IGES/PDES Organization (IPO), and PDES, Inc. (a consortium of private sector companies). ISO is in charge of the development and publishing of the international standard itself. PDES, Inc. and the National PDES Testbed (NPT) are jointly involved in an effort to test and validate Parts of the standard before introducing technical changes into the ISO arena. They will have software that needs to be kept under configuration management. As part of the effort to support the development of STEP, PDES, Inc., NPT and IPO may have technical reports that need to be stored in and accessed from a central location. Also, they may submit documents that they wish to be included in the international standard; these must be tracked. Graphical information models need to be contributed to the configuration management system as well.

As the proposed STEP specification progresses through the formal standards development process, the documents representing the continuing work must be placed under configuration management. In addition, software developed to test the concepts put forth in the standard must also be placed under configuration management.

Configuration management provides the fundamental operational capability for tracking and maintaining versions of documents and software. The Information Services Center (ISC) within the National PDES Testbed Project at NIST has taken on the responsibility of providing a configuration management system (CMS) in support of the STEP development effort. The system requirements will be defined in coordination with the participating organizations, and the system itself will incorporate the procedures defined by each organization.

Configuration management is a difficult problem because it reflects organizational procedures as much as it is deals with technical problems. Providing CM services to the various organizations will require significant interaction with the management structure of each organization. It is important to note the backdrop against which the processes of CM will occur. All of the organizations have had and some still are undergoing various forms of internal reorganization. While everyone recognizes the importance of CM, clear statements as to the exact form that CM is to take are not easy to locate. The tasks are large and sometimes ill-defined, and the structure within which each of them must function is fluid. CM can provide some structure and serve as a focal point to aid in the management and technical development of tasks for each organization. CM requires a definition and breakdown of organizational procedures as well as technical systems development.

2.3 Mission of the CMS (Configuration Management System)

The Configuration Management System serves as a librarian to any portions of aspects of the STEP project that need its support. The primary mission of the CMS is to provide an orderly framework within which the development of the STEP standard can take place. This includes configuration management of the STEP specification itself, as well as items which track and document the technical history of the standard's development process, such as software used to test the concepts put forth in the standard, and technical reports written as part of the standards testing and development process.

3 Configuration Management Concepts

The decision to use configuration management on a project must be weighed carefully with the particular needs of that project. Certain aspects of CM, if properly implemented, can solve certain problems that are otherwise commonly-encountered in the development process.This section will explore the use of configuration management in solving particular development problems, and will define the parameters by which a good CM system can be judged.

3.1 How CM Solves Common Development Problems

CM addresses a number of problems common to development efforts, and offers solutions to these problems. The following sections identify some of the most basic issues that arise during a document or software development effort, and an explanation of how standard CM functions can be used to avoid difficulties in the development process.

3.1.1 Identification of Controlled Items

During a development effort where CM is not implemented, much time is often spent in searching for particular items or trying to determine the proper version of a desired item. Items may be stored in diverse places and in different forms. Versions are compared manually to determine differences, to refresh one's memory as to the changes that had been made, or to determine which in fact is the latest version. In software development, time may be spent re-testing old modules to see which ones they were, or to determine their functionality relative to the new modules.

A CM system provides a unified method of identifying items and a convention for storing them, whether they are in a centralized location or in an organized, distributed system. The CMS also enforces the use of naming conventions, which provide a method of identifying both categories of information, and different versions of each item. Naming conventions may be implemented by storing like files in an appropriately named computer directory, or by attaching a prefix or suffix to the file names, or both. Naming conventions must be established by the user organizations, since they reflect organizational needs and procedures, but can be maintained by a CM system.

3.1.2 Centralized Control

Even on a project where storage of configured items is distributed, the CMS acts as a single point of control over and authoritative information about those items. Items maintained by and retrieved from the CMS are of known status. It is also known where they fit in the development cycle, and to what other items they are related.

The CMS maintains this control through the mechanisms of checkin and checkout.

Checkin is the act of formally submitting a document or software program to the CMS. When a user performs a checkin, the CMS will require that certain information be provided. This usually consists of the user identification, the status of the item, and the reason for checkin. The reason for checkin is either the initial submittal of the item to the CMS, or the submission of changes to form a new version of the item. If modifications have been made, the reasons for the specific changes should be recorded. The CMS records the date and time of each checkin.

Checkout is the mechanism by which controlled items are retrieved from the CMS. There are two types of checkout: one for retrieving read-only copies of configured items, and the other for making modifications to controlled items. The two types of checkout require different levels of security. Generally, any valid user in the system can check out any item for read-only purposes, but update access is restricted appropriately for each item or group of related items.

When any checkout is performed, the CMS makes a copy of the requested controlled version of the item from the CMS onto the user's home directory on the computer, or wherever the user is requesting that the copy be put, via the checkout command.

A checkout for update purposes requires user identification appropriate to the item being checked out, the status of that item, and the stage of the project. For example, access may become more limited during later phases of development, as tighter controls are needed. Alternatively, a software module may warrant wider distribution after it has been thoroughly tested. The requirements for progressive control throughout the development life cycle must be determined in conjunction with the users during the requirements definition phase of the project.

3.1.3 Version Identification and Status Accounting

Related to the issue of item identification is the issue of proper identification of different versions of the same item. In an uncontrolled environment there is no definite, unified way of determining which version of an item is the latest, in which order the versions were developed, what review process each has gone through, or even where all the versions of a particular item are located.

Each item in a CM system can change and develop over time. These changes are made for a number of reasons: improved understandability, response to reviews, response to testing, changing requirements, changing environment, or improved software performance and maintainability. A CMS stores the progressing versions in such a way that any version can be retrieved at any time, the reasons for the changes are recorded, and the actual changes from one version to the next can be listed.

A CM system must provide a standard method of labelling versions, and identifying the status of each version of each item in the system. This method may incorporate the use of naming conventions. For example, a directory called *jan31* may include the set of software module versions included in the release on that date. A suffix of ".exp" could indicate a status of "experimental." The users must be involved in the CM process, so that the specific names chosen are meaningful to them.

A particular version of each item that is of interest to CM is the baseline. The baseline is either the initial version of a controlled item, or the first version that is considered complete enough to serve as a model for future versions.

In addition to naming conventions, the CM system can also provide other, more informative methods of identifying versions. For instance, in addition to any identification information captured in a computer file name, the file itself may include either a field or a header which contains information provided by the user at the time the version was checked back in to the system. This information may consist of a valid status code (as defined by the user organization), or a comment explaining the reason for the updates, or both, depending on user requirements. A computerized system can be designed to ensure that these fields are filled in for every file.

3.1.4 Promotion

The concept of promotion in CM means that an item has passed a particular set of user-defined criteria, and is ready to be labelled as having passed them. Typically, each project will define a series of phases, or promotion levels, through which each developing item must pass on its way to completion or publication. At each promotion point, a specific user or user group has the authority to sign off on the completeness and quality of the item. The verifying user's identification and the date and time of sign-off are then stored as part of the history of the item. If the CM procedures call for audits, this information can support the auditing process. In any case, a computerized system can ensure that the desired promotion does not occur unless the user's authorization matches that required for the particular promotion in question. In this sense, by ensuring that proper procedures are followed, the CMS does some auditing along the way automatically.

3.1.5 Change Control

Change control involves avoiding conflicting changes being made to a document or piece of source code. That is, if two authorized users both have the same item checked out for update purposes at the same time, it becomes impossible to check back in either version without unknowingly omitting or undoing changes that were made in the other version.

CM avoids these potential problems by allowing only one authorized user at a time to have update access to any individual item. This is enforced through the checkout procedure.

3.1.6 Logical Grouping of Controlled Items

The ability to define different logical groupings of controlled items becomes necessary on complex projects. The problem of set definition arises when the application involves a set of different programs or documents, and each member of the set goes through changes and develops independently. The question then becomes: which version of each item should be included in each progressive functional set?

CM defines a release as a set of controlled items that together comprises the entire application, for instance, all the Parts of the STEP specification, or a matched set of software programs that function together as an application. Each defined release must specify which version of each of its constituents the release includes. A release may or may not include one version of each configured item stored on-line, but it cannot include any more than one version of each item. For instance, STEP Version 1 will contain only those Parts defined by ISO as ready to be published by the scheduled publication date.

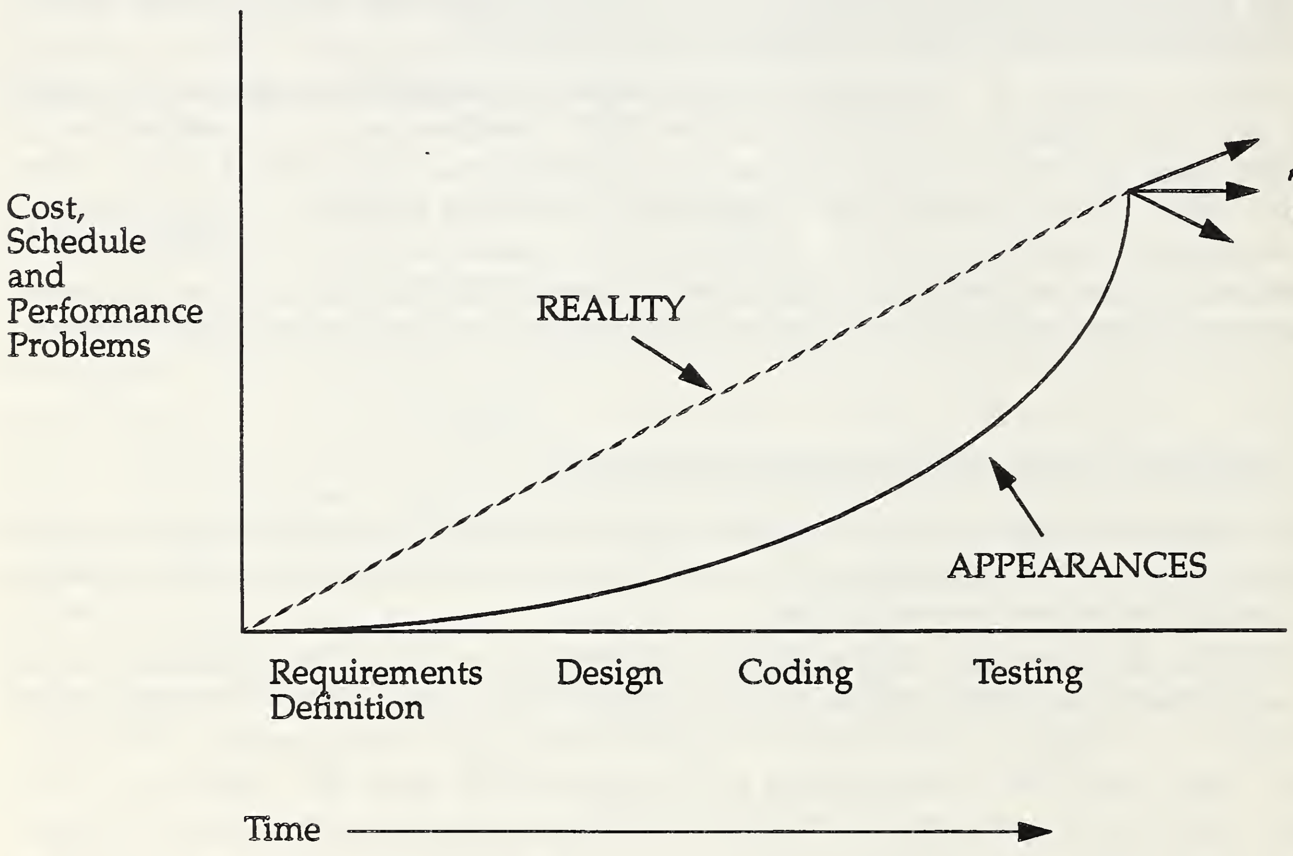
3.1.7 Visibility: CM's support of Quality Assurance

The development of documentation and software calls for the performance of certain quality assurance (QA) tests along the way. The nature and frequency of these checks must be determined by the application.

For software development, early visibility into module status can be important to the success of the project. Often problems are not discovered until late in the testing phase (see Figure 1, on page 8), when their cost and schedule impact is the greatest, although in reality, they have been accumulating and compounding since the beginning of the project. CM offers the structure of pre-defined checkpoints in the development process. These checks occur at promotion points, and document the fact that an authorized person has verified the completeness and quality of the module. CM also provides visibility by simply showing the progression in the development of each module and the reasons for changes. This allows specific changes to be easily "backed out" in the event of re-design, requirements changes, or problems uncovered during testing.

In the area of documentation development, ISO has an established set of procedures for the development and approval process for international standards, as defined in the IEC/ISO Directives [IEC/ISO]. There is a proposed set of CM checkpoints for the ISO procedures (see Figure 2, on page 12). CM itself does not ensure the quality of the configured items, but it provides a framework that can support quality assurance efforts. This support occurs in part simply by clearly delineating which documents are which, which versions represent the most recent iteration of a progression, and which versions of which documents should be grouped together to form a set. The CM provision for sign-offs as part of promotions also supports the QA efforts and serves to document them. The same is true for the updating of status fields and the addition of a "reason for change" each time a new version is checked in.

Figure 1 The Growth of a Software Problem



3.2 Characteristics of a Good CM System

The technical challenges involved in the development of a good CM system include ease of use, reliability, security, and accessibility.

3.2.1 Ease of Use

The CM system must be easy to learn and easy to use. It should use terminology and interface methods that are familiar to the users. Using the CM system should be preferable to not using it.

3.2.2 Reliability

It is important that the CM system clearly identify the documents and software modules under its control. The system must be reliable both in terms of on-line availability and in terms of accuracy of document and computer program storage and retrieval. CM procedures should plan for disaster recovery schemes, such as tape backup and storage.

3.2.3 Security

The CM system should provide certain access restrictions, but it is important to note that it does not provide all the features of a complete security system. Read and write access may be defined by the users on an file by file basis. Furthermore, promotion points (raising the status of a document or program to the next level) will require proper approval, as defined within the context of each organization.

3.2.4 Accessibility

A good CM system must provide easy access to all of its users. This includes being available for use through a computer that the users are familiar with, as well as providing a simple user interface. Cost, reliability, and security must be considered when choosing a remote access method to a centralized CMS.

4 Particular Implementations of CM on the STEP Project

Configuration management will be needed in different forms for different aspects of the STEP development effort. For instance, the documents that comprise the STEP standard itself have their own particular approval process. However, in the software development arena, there are different types of objects to be configured, and they have their own particular relationships. This section addresses the specific needs of the STEP project in configuration of three classes of items: documents, software, and test data files.

4.1 Configuration Management of Documents

The STEP Parts (volumes) [ISO] are the primary documents that need to be kept on-line and managed. At this level, the CMS stores each document, as it is received from the donating organization, as a text file. The configuration management of the *contents* of each Part, where that is relevant, is discussed in Section 4.3, on page 17.

To support the configuration management of documents, the ISC must first determine the ISO procedures governing the promotion of a document from one level of approval to the next [IEC/ISO].

After reviewing existing ISO procedures, the ISC can then recommend the configuration management practices that would be most helpful for the application [Katz1]. Then, in conjunction with the ISO-appointed representative for CM, the ISC can design a system that best meets the needs of the users.

The National PDES Testbed has presented a proposed documents approval process for STEP (see Figure 2, on page 12).

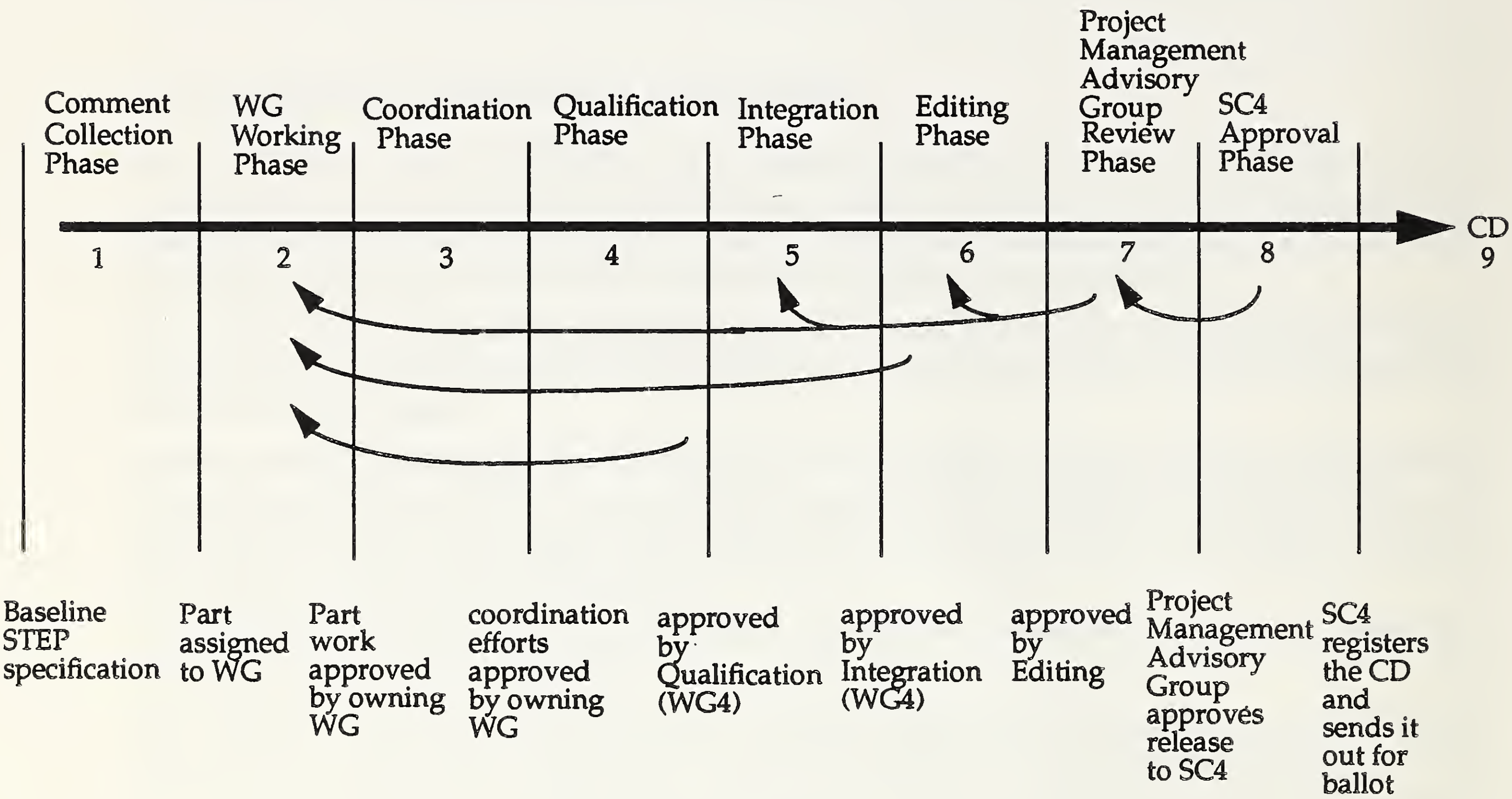
Key points of this proposal are: 1) that a distinct event must take place to promote each Part into the next phase, and 2) the requirement that only the owning Working Group (WG) may actually make changes to a Part (a WG is a collection of technical experts organized for one or more related technical areas). Reviewers make suggestions separately, but do not have update access to the actual Part. The first point is necessary to verify that the formal approval process is proceeding according to ISO rules. The second is recommended to ensure that changes which one reviewer sees as editorial-only do not inadvertently change the intended technical meaning of a Part, or reverse or conflict with earlier changes.

Within the context of an ISO Working Group, each user will access the CMS to view the list of controlled items, and to check out and check back in the controlled version of each Part.

This proposal serves as an example of the type of requirements analysis that must be done for the other user groups as well. First, the items to be controlled must be identified and classified. Logical groupings of the controlled items must be defined. The phases of the project, or levels of completeness, must be defined and appropriately labelled. The action(s) necessary to promote an item from each phase to the next must be decided upon, and the individuals or user level authorized to perform each promotion must be defined. Status codes must be agreed upon. Finally, any naming conventions, or guidelines for "reasons for change" fields must be developed by the user organizations.

Figure 2 STEP Documents Configuration Management Flow (proposed)

STEP Documents Configuration Management Flow



Notes:

- Each numbered Phase represents one to many versions of a STEP Part that are stored on-line under the configuration management system.
- Each vertical bar represents the action necessary to promote the Part from one phase to the next.
- During each phase, the files may be checked in and out many times; however, the promotion points (vertical bars) pertain to the entire Part, and require appropriate signature authority.
- At any time during the life cycle of a Part, that Part may be returned to the owning WG for re-work (e.g. if it is not approved by the Editing Committee).

4.2 Configuration Management of Software

Configuration management of software encompasses computer programs, executable object modules, "include" files, "header" files, and support files such as makefiles.

The NPT and PDES, Inc., in a cooperative effort, build many software tools which are used in various stages of product model testing. For instance, the STEP information models (contained in some of the STEP Parts) are written in a language called EXPRESS (defined in STEP Part 11 [ISO]). Each EXPRESS file is checked for syntax and compiled into an EXPRESS Working Form, using a compiler built in the Testbed. To populate a particular model with the data for a specific manufactured product, another tool must be used. A third tool is used to edit the resulting STEP file.

It is important that the software tools used are consistent with each other. If a particular tool is being changed by a developer, then it must be tested in a known environment. The configuration management system must allow for distinct environments for model testing as well as for tool development. In addition, during the development process, the CM system stores the progressing versions of each tool, and also defines matched sets of the tools. The developers can then easily identify existing matched sets (software of the correct version), and define new sets as they are tested.

4.2.1 How the testers use the combination of Testbed tools

A set of Testbed tools called "the PDES Toolkit" is used to test the EXPRESS models contained in the STEP specification (see Figure 3, on page 15). These tools will continue to be adapted to the testers' needs for the duration of the STEP development effort. However, the concepts depicted in Figure 3, on page 15, will remain valid, even though the details are likely to change.

There are five steps needed to produce a populated SQL database (see Figure 3, on page 15, and Appendix A., on page 25), starting with Express versions of the schema and the particular product data. Once this database is created, the testers run SQL queries against it to view the data, to test how easy it is to extract desired combinations of data, and, ultimately, to determine whether the input Express schema actually produced the desired results. The goal is to produce a database that contains the desired information in useful form, and behaves well (good response time and reasonable output) when queried. The Express schema, once refined enough to produce the desired results, is the item that will eventually be incorporated into the STEP standard.

The Express code file (Step 2) contains the definition of a conceptual model used to represent a manufactured product. Later in the process, the Express code file is mapped into a database schema. The same Express code file used in Step 2 must also be used in Steps 3, 4 and 5. The STEP physical file (Step 4) describes a particular product, in STEP physical file format, and must be built according to the schema described in the Express code file (Step 2). As such, the STEP physical file describes instances of the object class defined in the Express code file.

Step 2 serves to translate the Express code file into SQL commands, which, when run, create the schema tables in the SQL database. Steps 3 through 5 build the STEP physical file and use it to populate that database.

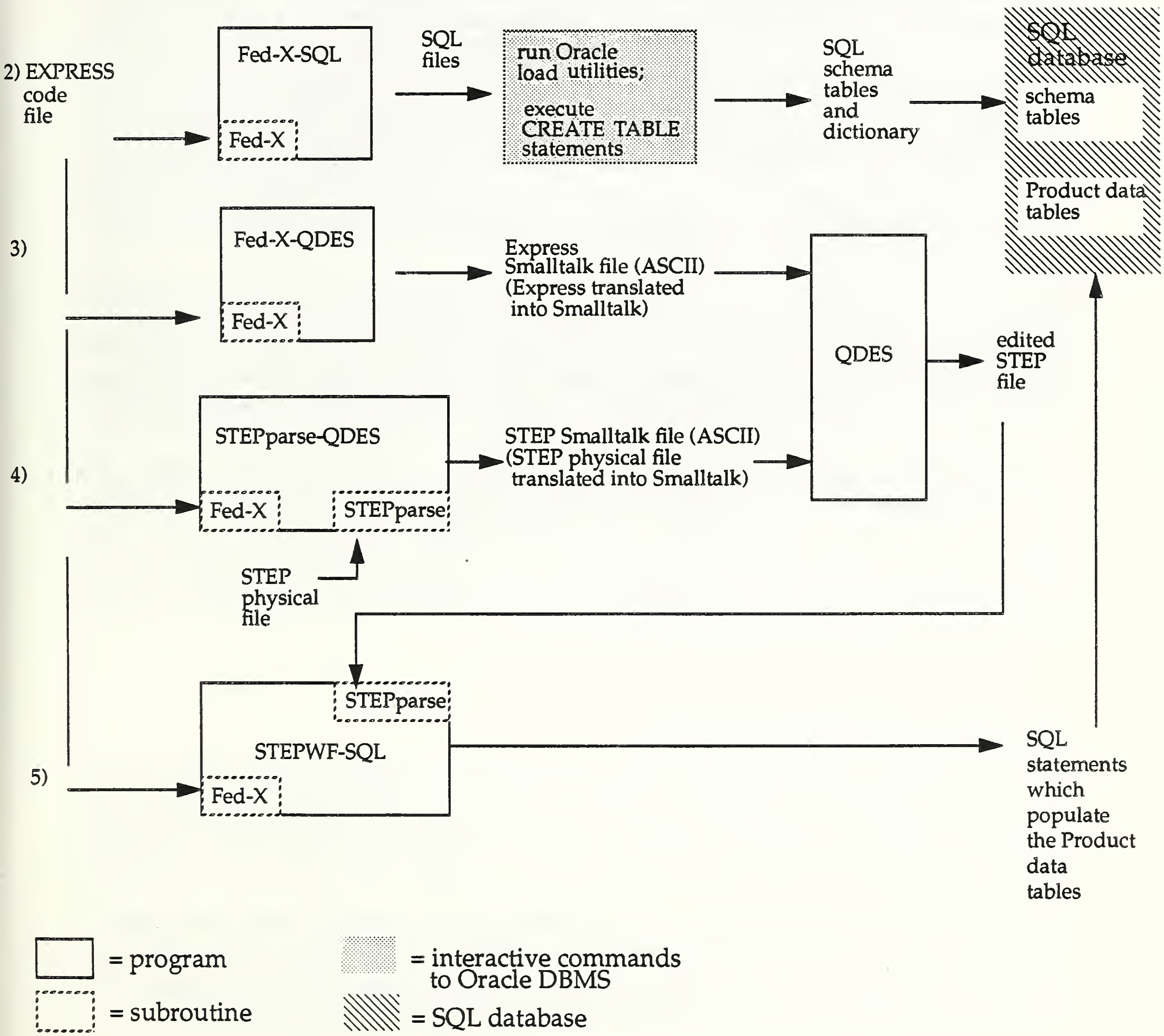
There are three conceptual levels involved here. The most general is the STEP standard itself, which establishes rules for how manufacturing information is to be expressed and structured. A particular implementation may be said to "conform to PDES" or may be referred to as "a PDES implementation", meaning that it follows the rules laid out in the STEP specification.

The next level is the Express schema. At this level, the objects are the building blocks needed to define manufacturing information. As such, each object itself really represents a class of lower level objects. These lower level objects constitute the third conceptual level. For example, at the schema level (the middle conceptual level), the objects are STEP schemas, and the instances may be points, edges, faces, etc. (characteristics which define a manufactured part). At the lowest conceptual level, the instances would be *particular* points, edges and faces that define actual products to be manufactured. An example of a STEP schema and its instances can be found in Appendix A., on page 25.

Step 1 in Figure 3 only needs to be performed when there is a change in the Express language itself (for example, going from the version of Express documented in ISO document number N362 to that of N496). Steps 2 and 3 are performed once for each schema being tested. Steps 4 and 5 are repeated until the testing of the schema (loaded in Step 2) is complete.

Figure 3 Building and Testing STEP Part Data Tables Using the Toolkit Software

1) compile the tools (Fed-X-SQL, STEPWF-SQL, STEPparse-QDES, Fed-X-QDES) using the libraries corresponding to the version of EXPRESS (ISO document N-number) used in the EXPRESS file to be tested. (note: the Oracle load utilities and QDES itself are independent of the Express version.)



4.2.2 The software tools as functional components of the Toolkit

Five of the tools of the PDES Toolkit [Clark1] [Clark2] are shown in Figure 3: Fed-X-SQL, Fed-X-QDES, QDES, STEPparse-QDES, and STEPWF-SQL. In addition, two of the more important subroutines, Fed-X and STEPparse, are used in many of the testbed tools, and are also shown in Figure 3. Only the programs are executed directly by the STEP testers; the subroutines are controlled by the Testbed programmers.

The subroutines, Fed-X and STEPparse, are both parsers. They check their input files for syntax, and load a version of the input information, called the working form (WF), into memory on the computer. The WF remains in memory only during the running time of the parent program (e.g., Fed-X-SQL). In the case of Fed-X, the input is the Express schema file. The syntax is checked according to the rules of the Express language, and the working form created is the ExpressWF. In the case of STEPparse, the input is the STEP physical file. The syntax is checked against the STEP physical file format, as defined in the STEP standard, Part 21 [ISO]. The working form created is called the STEPWF. In each case, when a working form is created, it is used by other subroutines of the parent program to produce the final output.

QDES stands for "Quick 'n' Dirty Editor for STEP," and is an editor for manipulating STEP product models [Clark3][Clark4]. QDES is written in Smalltalk-80, and its input files must be in a format readable by Smalltalk. The translators Fed-X-QDES and STEPparse-QDES, translate the Express schema file, and STEP physical file, respectively, into Smalltalk format. QDES allows the user to browse and edit an instantiated model, and outputs the revised STEP physical file.

The Fed-X-SQL program translates the Express model into SQL statements. When run, these SQL statements produce a database schema that represents the Express model. To load the part data into the database, the STEPWF-SQL program must be run. This program first loads the STEP physical file into memory as the STEPWF. It then builds and runs the appropriate SQL statements to load the physical file information into the Oracle database created by the Fed-X-SQL program.

4.2.3 CM in the Testbed Environment

For the software, two logical levels need to be controlled. First, the *users* (those testing STEP) must be able to identify the version of each tool needed for each test run. For example, a particular executable (binary) version of the translator, Fed-X-SQL, is needed to translate version N496 of Express to version 5 of SQL. This is true for each of the executable tools.

Second, the Testbed *programmers* (those developing the Toolkit) need to be able to identify which versions of each subroutine, in which combination, will provide the desired functionality for each version of each tool (for example, which version of the Fed-X parser needs to be used in the Fed-X-SQL version in the example above).

The PDES Toolkit software is now being managed under the Revision Control System (RCS). "A User's Guide to PDES Software Development Under RCS" [Bodarky] explains the use of the current system.

4.3 Configuration Management of STEP Data

In addition to the STEP Parts and the Toolkit programs, there are several types of data which are important to the STEP development process. Depending on project schedule and resources, the CMS may be expanded to support these items as well.

These include EXPRESS entities of all levels of complexity, data dictionaries which map to those entities, STEP physical file data (instances of the STEP entities; see Appendix A., on page 25), Application Protocols (AP's) [Stark], test data suites, test results, and other types of support information.

For the Express files, the CMS must keep track of which version of the Express language each file is written in, and which STEP physical files are built to which Express schema files.

There may be dozens of files associated with a single manufactured product. These files contain different types of data, but all relate to some phase of the manufacturing of a particular product. Some of these files may contain numeric control (NC) programs, some are STEP physical files (instances of a STEP EXPRESS model), and some may be process plan files or set-up information files.

The CMS can provide a hierarchical directory structure whose configuration is controlled, as well as providing configuration management of the individual files.

5 Remote Access to the CMS

Users of the CMS are distributed all over the world. Access for users throughout the U.S., Europe and elsewhere must be considered. Mechanisms must be put into place which meets these users needs. In addition, the cost of usage must be minimal as the CMS will support a collection of volunteer technical experts.

The configuration management system can be accessed through electronic mail, through the Internet network, or by modem (see Figure 4, on page 19).

Electronic mail (e-mail) is a low cost access mechanism which is enhanced by the use of an "archive server." Users can send e-mail messages to the NPT archive server [Ressler2]. The archive server can interpret commands sent in the mail messages, and interface with the CM system to carry out those commands. The archive server then sends responses or entire files back to the user via e-mail.

Direct modem access is also provided. To use direct modem access, remote users simply dial in to the NIST modem pool, and submit commands directly to the configuration management system as if they were local users.

File transfers or simple information requests can be accomplished with either e-mail or modem access method. File transfers can also be done across the Internet. The "STEP On-Line Information Service User's Guide" [Katz2] provides instructions for using those services which are already in place. "The National PDES Testbed Mail Server User's Guide" [Ressler2] provides specific instructions for e-mail access.

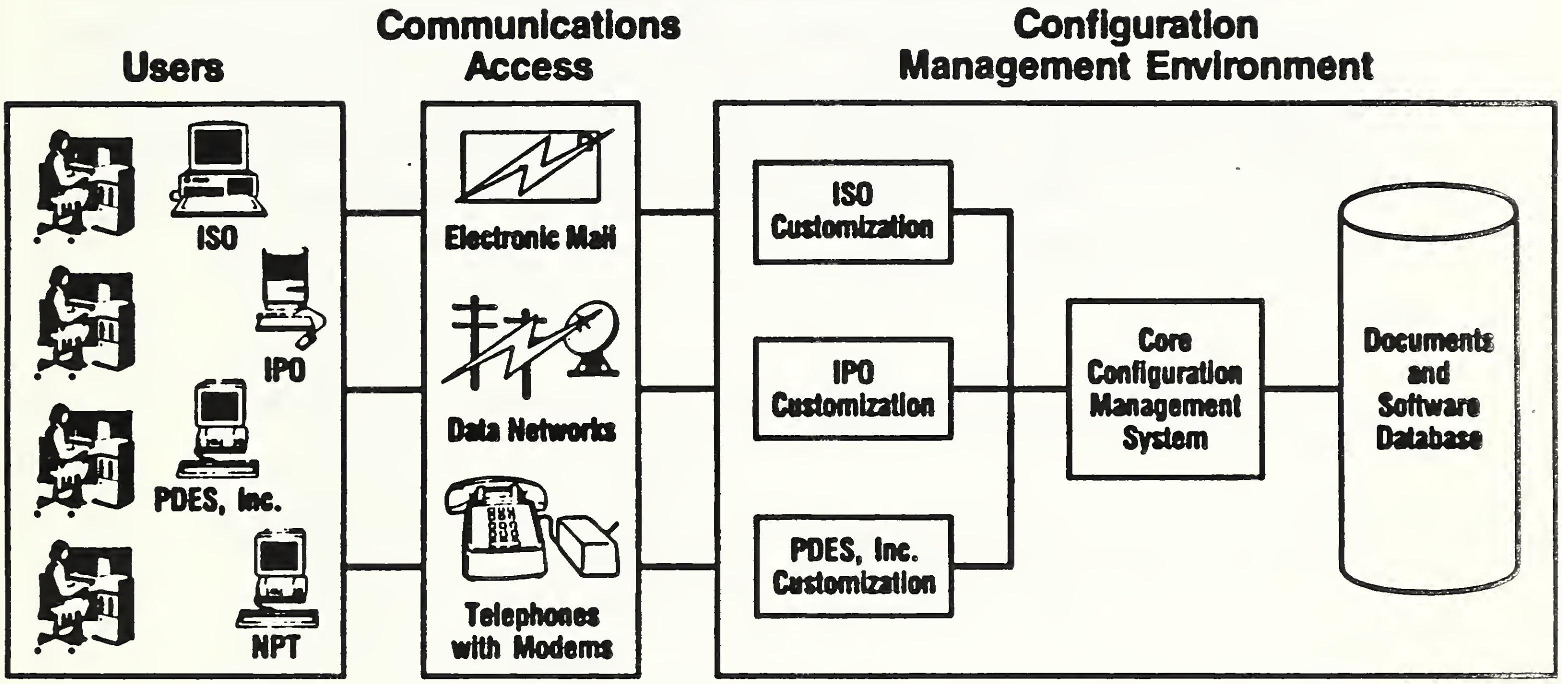


Figure 4. Functional view of the configuration management system

6 Implementation of the CMS

The implementation of a configuration management system to support the STEP development effort will proceed as described in the "Configuration Management Systems and Services Development Plan" [Ressler1]. The implementation will provide the basic functions described in this Concepts Document. The specifics of the implementation process will be further detailed in progression of deliverables called for in the Development Plan, particularly the "Comprehensive Requirements Document" and the "CMS Design Document."

In order for the CMS to aid in the STEP development process, it must be implemented and viewed as a set of practical, convenient tools, not as an impediment. Early user involvement, including participation in requirements definition and review of user's guides, is planned. A representative from each of the four user organizations will be consulted for input on the system requirements. These four "Configuration Managers" will also be instrumental in ensuring that agreed upon CM procedures are followed within their organizations.

7 References

[Bodarky] Bodarky, Scott, A User's Guide to PDES Software Development Under RCS, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, (to be published).

[Breese] Breese, J. Newton, Validation Testing Laboratory User's Guide, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, (to be published).

[Clark1] Clark, Stephen Nowland, An Introduction to the NIST PDES Toolkit, NISTIR 4336, National Institute of Standards and Technology, Gaithersburg, MD, May, 1990.

[Clark2] Clark, Stephen Nowland, The NIST PDES Toolkit: Technical Fundamentals, NISTIR 4335, National Institute of Standards and Technology, Gaithersburg, MD, May, 1990.

[Clark3] Clark, Stephen Nowland, QDES User's Guide, NISTIR 4361, National Institute of Standards and Technology, Gaithersburg, MD, June, 1990.

[Clark4] Clark, Stephen Nowland, QDES Administrative Guide, NISTIR 4334, National Institute of Standards and Technology, Gaithersburg, MD, May, 1990.

[IEC/ISO] IEC/ISO Directives, International Electrotechnical Commission, Geneva, Switzerland, and International Organization for Standardization, Geneva, Switzerland, 1989.

[ISO] ISO 10303-n, Product Data Representation and Exchange[1], ISO TC184/SC4 Secretariat, Gaithersburg, MD, (to be published).

[Katz1] Katz, Susan, Configuration Management of the STEP Documents: Procedures and System Requirements, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, (to be published).

[Katz2] Katz, Susan, STEP On-Line Information Service User's Guide, NISTIR 4491, National Institute of Standards and Technology, Gaithersburg, MD, January, 1991.

[McLean] McLean, Charles R., National PDES Testbed Strategic Plan 1990, NISTIR 4438, National Institute of Standards and Technology, Gaithersburg, MD, October, 1990.

[Mitchell] Mitchell, Mary, Development Plan, Validation Testing System, NISTIR 4417,National Institute of Standards and Technology, Gaithersburg, MD, September, 1990.

1. Informally known as STEP. Published as a set of Parts, each of which is a separate International Standard (IS).

[Ressler1] Ressler, Sandy and Katz, Susan, Development Plan, Configuration Management Systems and Services, NISTIR 4413, National Institute of Standards and Technology, Gaithersburg, MD, September, 1990.

[Ressler2] Ressler, Sandy, The National PDES Testbed Mail Server User's Guide, NISTIR 4508, National Institute of Standards and Technology, Gaithersburg, MD, January, 1991.

[Stark] Stark, Charles, and Mitchell, Mary, Development Plan, Application Protocols for Mechanical Parts Production, NISTIR ,National Institute of Standards and Technology, Gaithersburg, MD, (to be published).

8 Acronyms

BBS:	Bulletin Board System
BPR:	Block Point Release (a software configuration management term)
CALS:	Computer-aided Acquisition and Logistic Support
CCB:	Configuration Control Board (a PDES, Inc. organizational unit)
CD:	Committee Draft (formerly DP - Draft Proposal)
CM:	Configuration Management
CMS:	Configuration Management System
DP:	Draft Proposal (now called CD - Committee Draft)
E-mail:	Electronic Mail
IEC:	International Electrotechnical Commission
IPO:	IGES/PDES Organization
IS:	International Standard
ISC:	Information Services Center of the NPT
ISO:	International Organization for Standardization
NIST:	National Institute of Standards and Technology
NISTIR:	National Institute of Standards and Technology Internal Report
NPT:	National PDES Testbed
PDES:	Product Data Exchange using STEP
PDES, Inc.:	A consortium of private sector companies involved in the development of the STEP standard.
PMAG:	Project Management Advisory Group (an ISO TC184/SC4 organizational unit)
RCS:	Revison Control System (used for software configuration management)
SG:	SubGroup (a former ISO organizational unit)
STEP:	Standard for The Exchange of Product model data
WG:	Working Group (an ISO organizational unit)

9 Definitions

archive server: generically speaking, an archive server interprets and executes commands. As used here, the archive server will read e-mail from remote CM users, interface with the CM system and/or the local file system to fulfill the request, and send the response back to the user via e-mail.

baseline: can be used to identify either the initial version of a controlled item, or the first version that is considered complete enough to be put under control. A baseline is a particular instance of "version." There can also be a baseline release, which would be the set of all baseline versions of the items defined by the release.

checkin: the act of formally submitting a document or software program to the CMS. A checkin requires certain standard information, such as user identification and reason for submittal. The date and time of checkin are also recorded.

checkout: the mechanism by which controlled items are retrieved from the CMS. When a checkout is performed, the CMS makes a copy of the requested controlled version of the item from the CMS onto the user's current directory on the computer. There are two types of checkout: read-only and update. The user must be authorized for the type of checkout being performed. An update checkout locks out other users from having update access to the CMS version of the item, until the item is checked back in. The CMS always records the user identification, date and time of each checkout.

configuration management system (CMS): The software which enables users to access files on a version by version basis in a controlled manner.

document: an item in either electronic or paper form.

makefile: a file which directs the compilation of a set of source files to create an executable program.

promotion: the official process of verifying and recording that an item has passed a particular set of user-defined criteria. Typically, each project will define a series of phases, or promotion levels, through which each developing item must pass on its way to completion or publication.

release: a set of controlled items that together comprises the entire application, for instance, all the Parts of the STEP standard, or a matched set of software programs. Each defined release must specify which version of each of its constituents the release includes. A release may or may not include one version of each configured item stored on-line, but it cannot include any more than one version of each item. For instance, STEP Version 1 will contain only those Parts defined by ISO as ready to be published at that time.

revision: the process of making changes to a checked-out version of a controlled item, resulting in a new version when the revised item is checked back in to the CMS. Revision is also used as a noun, referring to a particular version.

sign-off: the formal, recorded verification, by an authorized user, that a particular version of a configured item meets user requirements for the particular development phase in question.

version: a particular instance of a controlled item. Each version is stored in a separate computer file. Note, however, that the user term "STEP Version 1" really refers to a release, as used in this paper.

Appendix A. An Express Conceptual Model and Its Instances

A STEP Information Model, portion describing a cartesian point in EXPRESS:

```
ENTITY cartesian_point
 SUBTYPE OF (point);
 x_coordinate : REAL;
 y_coordinate : REAL;
 z_coordinate : OPTIONAL REAL;
DERIVE
 space : INTEGER := coordinate_space (z_coordinate) ;
END_ENTITY ;
```

Examples of Instances of cartesian point, defined in a STEP physical file:

```
STEP;
DATA;
@1 = CARTESIAN_POINT (,0.0, 0.0, 0.0);
@2 = CARTESIAN_POINT (,0.0, 0.0, 1.5);
@3 = CARTESIAN_POINT (,0.0, 1.5, 0.0);
@4 = CARTESIAN_POINT (,0.0, 1.5, 1.5);
ENDSEC;
ENDSTEP;
```

NIST-114A
(REV. 3-89)

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

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4538

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
APRIL 1991

4. TITLE AND SUBTITLE

Configuration Management Concepts Document

5. AUTHOR(S)

Susan B. Katz

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

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

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

The purpose of this document is to establish Configuration Management (CM) concepts to be applied in support of the STEP Standard for The Exchange of Product model data) development effort. Configuration management is the management of change. It is a formal discipline which provides methods and tools a) to identify components, versions and baselines of selected items and b) to control changes to those items. CM provides a method for logically grouping related components throughout the various stages of product development. It also provides visibility and traceability for the evolving status of each item. An effective CM system thus identifies, controls, records, and reports on any functional, physical or status changes to controlled items. A detailed discussion of CM issues is provided in section 3 of this document. A comparison of CM and other types of support services used in software development (verification, testing, validation, and quality assurance) can be found in Appendix A.

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

CM; Configuration Management; International Organization for Standardization; ISO; STEP

13. AVAILABILITY

[X] 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.

[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
32

15. PRICE
A03

ELECTRONIC FORM