Electronic Component Information Exchange (ECIX)

Pinnacles Component Information Standard (PCIS)

Tutorial

Version 1.4-1-123197

Silicon Integration Initiative, Inc.
4030 West Braker Lane, Suite 550
Austin, TX 78759

Foreword

The publication of the Si2 PCIS Version 1.4 marks yet another step in several years of invention, discussion, argument, and consensus, followed by further questions. This document is the product of the effort of many individuals and the contribution of many dollars and person-years of effort. To that end, every person who was involved at any point along the way should take pride in the existence of this Si2 standard.

The original development of the PCIS standard was accomplished under the coordination of the Pinnacles Group, with members from five electronic component manufacturers who joined together to create a technical information exchange standard for use by the electronics industry. These original member companies have all been active members of the Si2 electronic databook working group, and it was in conjunction with Si2 (then the CAD Framework Initiative, Inc.) activities that the original concept of the PCIS was developed. The work of Hitachi America Limited, Intel, National Semiconductor, Philips Semiconductors, and Texas Instruments recently was extended when Hewlett-Packard and IBM Microelectronics joined the effort. Representatives of H-P, Hitachi, IBM, Intel, National, Philips, and TI now form the Si2 ECIX Project Technical Advisory Board.

The earlier Pinnacles Group PCIS 1.2 became a part of CENELEC TC 217 WG 4. Future Si2 standards for ECIX PCIS will continue to be submitted to CENELEC, en route to target IEC standards.

The publication of the ECIX PCIS Version 1.4 represents another in a series of milestones in the maturation of the electronic computer-aided design industry. The Si2 membership and the industry have demonstrated again that there are areas of common agreement in the architecture, construction, and delivery of electronic product design systems information. This standard will be extended into additional areas with the ongoing development of ECIX PCIS. Each new release of this standard will continue the proven progress exemplified in the PCIS Versions 1.0, 1.1, 1.2, and 1.3.

Si2 invites interested parties to provide input into the development of ECIX through active participation in the various ECIX Working Groups, and in other Si2 projects where appropriate. For membership information, contact Si2 Member Services, at http://www.si2.org.

Si2 continues as a leading contributor to standards for the EDA industry with actively participating companies and many individual contributors. They continue to work together to solve design system interoperability problems in areas where a common solution is recognized as being the most effective approach. Projects such as the ECIX Project are contributing important open standard solutions to key industry problems. As we move forward into new releases of this and other Si2 standards, the importance of the specifications will become registered in larger and larger circles within the industry.

Si2 and the ECIX team look forward to additional development and release of enhanced ECIX PCIS standards in the future.

Pinnacles Component Information Standard (PCIS) Tutorial Table of Contents

1 Introduction

This document, the PCIS SGML Tutorial, is intended to explain the principles behind the PCIS SGML application and to highlight the most important and more unusual aspects of that application. Parts of this tutorial will be of interest to anyone interested in the PCIS, while other parts are quite technical and will be of interest only to the creators and managers of SGML applications.

This tutorial begins with a description of the background of the project and the PCIS application standard. This is followed by a discussion of the general architectural principles on which the standard is based and a description of the SGML architecture. The next sections concern important information constructs within PCIS: Document and Element Identification; Product Identification; and Parametric Data. Each of these sections describes the approach to a particularly important type of component information. The tutorial then discusses some further document content considerations and suggests methods for using external files. Sections of this tutorial include:

- PCIS Glossary (which should help readers new to either SGML or electronic component information);
- Element Naming Conventions (which were used in constructing the SGML tags);
- The Initial Pinnacles Guidelines (which were written at the beginning of the project to guide development); and
- An explanation of the PCIS Formal Public Identifiers (which are the strings used to identify each formal part of the SGML application such as the Document Type Definitions (DTDs) , the DTD modules and every document created using the PCIS SGML application).

2 Background

2.1 The Decision to Use SGML for Document Interchange

From the initial investigations, the Pinnacles Group has been looking for a document modeling and interchange format that:

- Was platform and vendor non-specific;
- Allowed the inclusion and referencing of other applicable standards;
- Was supported by commercially available tools and services;
- Would ensure longevity of the electronic component data; and
- Could support current publishing processes (such as paper and CD-ROM).

Standard Generalized Markup Language (SGML) met these specifications. SGML, which is defined in international standard ISO 8879, is a widely accepted and supported standard for the encoding of structured information. SGML provides a vendor and platform independent way of identifying information structures.

SGML has been adopted by many organizations and industries whose needs for information exchange are similar to those of the electronic component industry, e.g., the airline, telecommunication, and automotive industries, the Department of Defense, and many major and minor publishers. These organizations have also chosen SGML because SGML can provide:

- A method of describing the structure of information and the relationships between data elements; and
- Content-driven document architectures which are focused on the purpose of the information rather than the physical (printed or displayed) appearance of information.

SGML works through the use of “tags” which are inserted in text to identify the beginning and end of each element (or type of information) in the text.

2.2 PCIS: An SGML Application Standard

Calling for the use of SGML is not enough to provide the electronic components industry with a viable interchange standard. Consistent encoding of the information set forth in databooks required the creation of an SGML Application Standard.

2.2.1 SGML Application Standards

An application standard is a specific subset of the standard. For example, a general standard might say:

- Standard spoons may be 6, 7, 8, or 10 inches long;
- Standard spoons shall be made of a non-toxic wood, or of a food-safe non-reactive metal such as stainless-steel or sterling silver; and
- The length of the standard spoon may be indicated either on the back of the handle of the spoon or on a removable sticky label attached to the inside bowl of the spoon.

An application specification for a specific manufacturer’s spoons might say:

- We will make all spoons out of pewter;
- Our teaspoons will be 6 inches long;
- Our serving-spoons will be 10 inches long; and

- Raised lettering on the back of the handle of the spoon will indicate the date of manufacture, the length of the spoon, and the bowl capacity in fluid ounces.

The application standard establishes strict guidelines within the context of a general specification and selects among the options available in the general standard.

An SGML application standard specifies how a group, organization, or industry will implement SGML. An SGML application standard conforms to ISO 8879, the international standard for SGML, but is far more specific. An SGML application standard typically includes:

- Selection of SGML features;
- A standard tag set;
- Standard documentation and definitions;
- A standard approach to structure; and
- Standard table coding.

An application standard may be described as enabling or prescriptive. An enabling standard offers many options for content models; a prescriptive standard has fewer options and more rigid models.

2.2.2 Selection of SGML Features and Capacities

ISO 8879, the International Standard for SGML, defines a basic set of capabilities that all SGML applications must have. In addition, it defines several optional features. The optional features include several ways to reduce the number of tags in documents and to process complex documents. In general, the more features used, the more computer resources required to support an application.

In SGML, each application must specify the maximum length of several SGML components, including tags, attribute values, and the entities which hold standard text. These “capacities” can be used to estimate how much computer memory will be needed to process documents. The larger the number assigned to the capacities, the more computing power needed to process documents. Small capacities may be selected if it is more important that documents be processable on small computers, while larger capacities allow more user-friendly tags, larger boiler-plate text blocks, and larger, more complex documents.

An application standard should specify which SGML features and capacities must be supported by all systems, sub-systems, hardware, and software that support the standard. For example, while some SGML applications have a maximum tag name length of eight characters, the US Department of Defense's application standard specifies thirty-two characters. This allows them to use tags with long, mnemonic names (such as SHORTTITLE, which identifies a smaller version of the document title to be used in running heads and indexes). Similarly, PCIS specifies a maximum tag name length of thirty-two characters.

2.2.3 Standard Tag Set

The international standard for SGML provides the means to name tags and specify their relationships to each other. However, the SGML standard does not provide any tag names nor does it recommend any. There is nothing in the SGML standard that prohibits users from making up different tag names for each document. For example, the title of a research paper could be tagged <TITLE>, the title of a bibliography could be <TI>, and the title of a Memorandum <TTL>. An SGML application specifies a set of tags to be used by a group of people and provides definitions of those tags so they can be used consistently.

2.2.4 Standard Documentation and Definitions

An SGML application, like any other computer application, must be documented in order to be useful. It is essential that users be taught to use a new system, and that they have appropriate reference materials at hand to optimize their use of the system. One part of that SGML application documentation is the Tag Library: documentation including the name of each element, the tag used to represent it, a definition of it, an example, and usage instructions. Each of the tags in the standard tag set must be defined clearly, with samples familiar to the users who will be working with the tags.

2.2.5 The PCIS Application Standard

The PCIS SGML Application Standard includes:

- An electronic-component-specific set of elements and tags – The PCIS elements identify types of information concerning electronic components, such as Bus Modes and Pinout Diagrams and Parameter Symbols. The tag names are shortened forms of the element names to be used within the text to mark information types.
- Document Type Definitions (DTDs) for common document types – An SGML DTD is a representation of an information structure or database design that is readable by machines and humans, defines hierarchical relationships, allows recursion and complex models, and can be validated for ambiguity and completeness. The PCIS DTDs provide information models for specific electronic component documents: databooks, datasheets, application notes, and miscellaneous documents.
- A standard method for constructing formal public identifiers – SGML formal public identifiers provide a mechanism for giving a unique name to every PCIS document, such as a datasheet or databook.
- An SGML Declaration – The PCIS SGML Declaration names which features of SGML can be used in PCIS documents.

2.3 Goals of the Pinnacles Group

During the development of the PCIS, the Pinnacles Group worked under a set of informal rules called “Goals/Non-Goals.” This arrangement allowed the group to decide, on a case-by-case basis, whether a given item under discussion was “in scope” for the overall effort. This, in turn, provided a continual focus and allowed the group to accomplish a greater amount of work during the first year of the project than would otherwise have been possible.

These goals and the associated non-goals are described in detail in the general information section of the project’s site on the worldwide web at **www.si2.org/ecix**. As the PCIS is extended and becomes interoperable with other standards in the same and related domains, the information on this web site will be updated accordingly.

In brief, however, the overall goal of the PCIS is to provide a standard for the interchange of technical information about electronic components which is:

- Extensible;
- Hardware, software, platform, vendor, format, and media-non-specific;
- Interoperable with the Silicon Integration Initiative (Si2) Electronic Component Information Exchange (ECIX) architecture, including component terms dictionaries; and
- “[A] seamless flow of electronic component information efficiently from producer to consumer.”

2.4 Information Included within the PCIS

The scope of the Pinnacles Component Information Standard effort was set before the first document analysis workshops began. Information analyzed included:

- Everything in component databooks (this includes datasheets, application notes, front and back matter such as design consideration sections and waveforms, the Table of Contents, indexes, package outlines, instruction sets, packaging, etc.); and
- Anything in stand-alone parts of databooks (such as datasheets or application notes) not included when such stand-alones were collected into databooks.

Initially, Component Datasheets were viewed as very different from other types of datasheets. During the course of the analysis, the decision was made that all types of datasheets, such as Family Datasheets, Process Datasheets, and Software Datasheets, are so similar that it is often difficult to determine which category a particular document fits. All datasheets contain two basic types of information: data concerning the definition of a product and application material concerning the use of that product. The two types of information may be intermixed in different ways within a single databook volume.

2.4.1 Metadata

There are two types of metadata associated with databooks: general metadata and company-specific metadata. General metadata comprises information about the databook or datasheet and includes such data as document title, publication date, and what parts of the information have been changed. Company-specific metadata may include such information as who wrote or changed portions of the information, what has been reviewed and by whom, and which process, machine, or corporate division produced various pieces of information. General metadata was declared to be within the PCIS scope, company-specific metadata to be outside the scope.

PCIS Version 1.4 adds several metadata-related attributes to the models for PCIS Application Note, PCIS Databook, PCIS Datasheet, and PCIS Document. These attributes give corporations the ability to consistently encode information about the confidentiality of the document within an individual creator company or enterprise (“confidential”), the name or names of any companies external to the creator company or enterprise to which the document has been disclosed (“disclosed.to”), and any nondisclosure agreement connected with the document that the creator enterprise wishes (“nda.info”).

2.4.2 Computer-Sensible Models

In the electronic component industry, computer-sensible models (such as SPICE models) are rarely included in print datasheets but often supplied as an associated information product. Such a model might replicate information in a datasheet or supplement it. Modeling information is most appropriately supplied in electronic form; when it is supplied in print form it must be rekeyed to be used. In an electronic environment, the user may want to view the raw data for a model, feed the model to another application, or launch a modeling application directly.

The preliminary PCIS decision was that modeling information could be encapsulated in an SGML file or stored as an external file referenced from within the SGML file. In either case, the model would be stored in its native format, not in SGML. Therefore, models were declared to be within PCIS scope. For the purposes of PCIS, a model was defined as a model supplied by the provider of the PCIS document. Potential models that third parties, customers, or users of the document may develop from the data provided were not included in the scope.

2.5 Document Types Not Included in the PCIS

The electronic component industry produces many other types of documents. Those listed below were specifically excluded from the analysis effort of PCIS:

- Publicity materials, leaflets, advertisements, and promotional materials such as Product Briefs;
- Journal articles;
- Catalogues;
- Master selection guides;

- Manuals, including user manuals, programmer reference manuals, software manuals, and hardware reference manuals;
- Training manuals, tutorials, and other training materials; and
- Manufacturing instructions such as Build Sheets and Process Descriptions (recipes).

The initial PCIS was developed for datasheets and data books only, but could be extensible to other electronic component information forms. The initial PCIS was developed for English language documents only but the design does not prohibit support for other natural languages in the future.

2.6 The Document Analysis Process

The Pinnacles Group used a variant of Joint Application Development (JAD) methodology to gather the information necessary to produce the PCIS. In JAD, the traditional “serial interview” approach to gathering requirements is replaced by one-time, one-place facilitated decision-making, with all interested parties represented. Facilitated JAD workshops are often used in software and hardware development projects and in the initial stages of TQM (Total Quality Management) implementations for the rapid and accurate specification of requirements. The PCIS document analysis was accomplished during a series of JAD-like Rapid DTD Development (RDD) workshops. (RDD is a proprietary methodology of the ATLIS Consulting Group). JAD and RDD are “team design” techniques, in which systems professionals and users cooperate in a highly structured environment to reach consensus on requirements.

Intel, National Semiconductor, Philips Semiconductors, and Texas Instruments each hosted a week-long RDD analysis workshop. During each of those intense, facilitated sessions, engineers and publications experts from one company determined the structure, content and relationship of data elements contained in component documents, concentrating on that company's product literature. The workshops were cumulative as each successive company started with and built upon the work of prior analysis sessions. This progression of discovery and consensus-building culminated in a reconciliation workshop, attended by representatives from each member company.

This PCIS documentation describes the analysis results from that reconciliation workshop as well as the decisions made during architecture and planning sessions held by representatives from all five founding companies. A formal Review Session was held in April 1994, with input solicited from organizations outside the Pinnacles Group. Comments and suggestions from the April review were incorporated into PCIS 1.1. Early implementations in 1995 detected some errors and obvious inadequacies of PCIS 1.1. Corrections of these bugs and changes to make Pinnacles data more compatible with the World Wide Web were incorporated in PCIS 1.2. PCIS 1.3 included a revised table model to facilitate interchange and a few changes based on experience in implementing PCIS in several ECIX member companies. PCIS Version 1.4 contains further changes and simplifications based again on implementation experience.

3 General SGML Architectural Principles of the PCIS

The Pinnacles Component Information Standard (PCIS) defines an interchange format, which is neither a document development format nor a presentation format. Thus, some of the capabilities and functionality that may be desired for authoring or presentation are not included in the PCIS by design. However, enough information has been included in the interchange format to create the form(s) of those documents desired for development and/or presentation. The PCIS standard design has been driven primarily by the nature of the component information; such considerations as data volume and ease of implementation have been considered only secondarily.

An SGML Document Type Definition (DTD) is a set of rules that can be used to ensure that all documents within a class governed by that DTD are constructed in the same way. A class of documents is the set of documents that is defined as having the same rules for markup, including the same elements and element relationships. In general, a class of documents can be considered to be documents that are similar in structure and content. DTDs range from “completely enforcing” (where a specific and highly detailed set of rules is enforced throughout the document structure) to “enabling” (where as many existing documents as possible are legal under the rules of the DTD). The PCIS DTDs are, by design, enabling DTDs. Each relevant information element has been named and given an identifier, so that different companies can name information of the same type consistently if they choose to identify that information in a document, but very little information is actually required to be identified.

The decision to design PCIS DTDs as enabling DTDs was motivated by the requirement that the interchange standard must work for a large number of enterprises and users throughout the industry. It is not expected that any company will use the currently-defined PCIS DTDs as authoring DTDs but that more prescriptive company-specific models will be designed and used for authoring and presentation.

3.1 Required Document Elements and Element Sequence

The Pinnacles Component Information Standard provides a mechanism to identify the information currently printed in documents that pertain to electronic components. The PCIS does not require that particular information be present in any given document. Since it is a stated goal of the PCIS that member companies be able to use the standard to describe existing documents such as component datasheets, this standard also cannot require or enforce the sequence in which any current product information is to appear.

Therefore, element sequence has been enforced only in those elements that are not currently printed in the datasheets or that are part of the document metadata, that is, information concerning the document as opposed to information within the document. When possible, such metadata has been positioned as the first element within an element. Element sequence has been imposed within metadata as well as within non-printing supplementary information, such as models that have little or no relationship to current printed products.

Any PCIS-conformant document must identify the company that created it (Enterprise Information) and the name of the document (Document Title). No other elements must be identified.

3.2 Generated Text

Generated text is text such as automatically generated headings and the numbers or bullets that precede list items that could be created from SGML tagging by a formatting or display engine. Although SGML-encoding permits and encourages the use of generated text for standard structures, including numbered elements, this feature is not used with PCIS conforming documents. Because the PCIS is an interchange and not an authoring standard, generated text is created by the authoring application, and that text is included in the PCIS exchange version of the document. Thus, while the authoring set may include elements that are intended to generate text, no elements are expected to generate text in the PCIS interchange version of the documents. Generated text such as automatically named headings or titles are included in the title element content. Numbered structures (such as lists, numbered sections, or

footnotes) in an interchange document instance carry their numbering with them. The attribute named “label” has been placed on all numbered structures to hold this information. Generated text such as bullets on lists is also carried in PCIS documents as attributes. Such attributes indicate the preferred rendering of the authoring enterprise, but the attribute values can be overridden by the receiving enterprise.

It is highly probable that the authoring-set DTDs used to create PCIS documents will often automatically generate numbers and other text, and their generation will be required before a document is moved to the PCIS interchange format to be exchanged.

3.3 Information Classes (the “div.class” attribute)

Many kinds of technical information exist within a single document such as a datasheet. The Pinnacles workshops concluded that there are six distinct useful categories (types or classes) of technical information present in electronic components documents. No single document need have all six classes, although many datasheets do. If the category of each piece of information were clearly identified, the categories could be used to retrieve related groups of information. Such groupings might be used by an engineer to limit data searches to the most specific product information, or by marketing to create spin-off publications, or by a researcher to determine the environmental impact of a process. Use of the information class (in the “div.class” attribute) is key to the creation of high-quality PCIS documents useful in electronic retrieval.

The six types/classes of information identified in electronic component documents are:

- Product Summary Information;
- Detailed Product Specification Information;
- Application Information;
- Safety and Environmental Compatibility Information;
- Information Concerning Support Tools; and
- Reliability Data.

There is no requirement that the data in each class be grouped together within a document. In fact, in many documents the information in any given class may be scattered throughout. As long as information can be identified as belonging to a particular class, the physical arrangement does not matter. (The PCIS calls this the “ice cream model” of information arrangement and it is described in the following pages.)

All divisions of a PCIS document may be identified as containing information of a particular class.

3.3.1 The Ice Cream Model

It seemed to the developers of the PCIS that component documentation in general, and datasheets in particular, could be usefully compared to ice cream. The ice cream analogy is explained here to demonstrate why the PCIS is so complex and why it enforces so few rules regarding the sequence in which information will be found.

The basic structure of information within a PCIS document may be thought of as analogous to the way ice cream is packed into containers.

Table 1. The Ice Cream Analogy

Ice Cream	Datasheet
Flavors. Ice cream comes in many flavors (chocolate, vanilla, strawberry) which are sometimes packaged alone and other times packaged in combination.	Information Classes. Datasheets contain several distinct types of information (Product Summary, Detailed Product Specification, Applications Information, etc.) which are sometimes present alone and are sometimes combined in a single datasheet.
Combinations. Some packages contain only one flavor of ice cream (vanilla). In some containers, the chocolate is at one end, followed by vanilla, with strawberry at the other end (Neapolitan). In other containers, the flavors swirl together (strawberry-ripple).	***Combinations.*** In some datasheets all of the summary information appears before the product information, which is then followed by the applications information. In other datasheets the technical information about the product may be intermingled with information about the application of the product.
Goodies mixed into the ice cream. There may be flavor "chunks" mixed into the ice cream, such as pieces of cookies or candy bars. Some of the chunks can appear in many flavors, while others are restricted to only one flavor. For example, strawberries may be expected only in strawberry ice cream, and chocolate chips only in chocolate or vanilla. Other chunks, such as perhaps marshmallows, may occur in many flavors.	***Floating Elements.***There are "chunks" of highly detailed information intermingled in the documents, such as characteristics tables, environmental impact statements, and references to models. Some of the chunks can appear in any type of information (characteristics may appear in the summary, in the technical product description, and in application notes), while others may be restricted to only one section.
Some of the chunks have quite a detailed structure, for example sandwich cookies or multi-layer candy bars. Other chunks may be quite simple, such as chocolate chips or marshmallows.	***Some of the chunks have quite a detailed structure,*** for example, tables and characteristics. Other chunks may be quite simple in structure, for example cross-references or sub-divisions.

3.3.2 The Purpose of the "div.class" Attribute

The "div.class" attribute, when properly used, can be the most flexible and powerful subject-access tool in PCIS. The division classes identify the major types of information in PCIS documents.

Although the division of information into these six categories is the most important logical organizing principle in PCIS, the information is rarely if ever arranged by category in datasheets. The information concerning any one category (such as Application Information) can be scattered throughout the datasheet. This tends to be true regardless of whether there also exists, for example, a section explicitly labeled Application Information, or the like, in the same document. Therefore, a user searching for all the information on how to design-in a product may need to look in many places in the document. In order to make retrieval by category possible, without requiring that datasheets be completely redesigned, the "div.class" attribute was invented.

An alternative method for organizing datasheets could have been to use the named classes as the top level elements:

```
<!ELEMENT datasheet - -     (prod.summary | detail.spec |
                             application.info | safety.envir.info |
                              support.tools |reliability.info)* >
```

This technique was discussed as a possible organizing principle early in the PCIS design. This model was rejected in favor of a division-based model, and it was decided to embed this information in the division model using a “division class” attribute. The primary advantages of the “div.class” attribute is the flexibility it allows in document organization and the enabling of retrieval regardless of the organization of the document.

The attribute allows divisions AT ANY LEVEL in the document hierarchy to be designated as any information class. In particular, low level divisions (perhaps nested three or four levels below the “top”) can be identified as belonging to a class that differs from the class of their containing elements. For example, inside a division of class “detail.spec” it is likely that most of the information in the class will be detailed specifications; but a few subsections may in fact be of class “application.info”. By explicitly specifying the “div.class” information, the data becomes retrievable by searching for these categories without requiring that the documents be reorganized.

3.3.3 Use of “div.class” in Search and Retrieval

Use of the “div.class” attribute empowers searching in both the positive and negative sense of search. A user wanting to see all application-specific information in the document would not only find everything that was put in the top level Application Division, but also any information in the other divisions that was designated as being of the “application.info” class. Similarly, a user could find all the environmental data (and only environmental data) concerning a product.

Information classes can also be useful in limiting searches. A user could search for any mention of a particular product, limiting the search to detailed technical specifications and avoiding the more general or application-oriented references.

Many users of online information choose not to view information in the sequence in which it is supplied. Using sophisticated electronic viewers, users can move through the information in a non-linear sequence. In order to make this possible, users move from one identified piece of information to another. By implication, if a piece of information is not identified, the reader will rarely see it.

3.3.4 Implications for PCIS Authoring Sets

By using the “div.class” attribute as described, datasheets can be organized by subject or habit, and the classes of information can be intermingled at any level. This should not create much extra work for authors. Most of the named technical divisions in PCIS have a default value for “div.class”, and we don't expect this class to be changed often, if ever, on these named divisions. In order to make class information available for generic divisions, “div.class” values will have to be provided by the author.

In creating PCIS authoring sets, companies are creating many new named “divisions” for use internally. These are used for enforcing authoring rules such as “this type of characteristic table must appear before that one”. On export from the authoring application to the PCIS interchange format, these named divisions are “mapped” to the PCIS Division element. It is likely that most of these new named divisions can be assigned a default“ div.class” in their authoring models so that authors will not need to think about “div.class” for these elements but searchers will be provided with the class information. Thus, even if PCIS does not name the division explicitly, a searcher can determine if this division is product summary information or detailed specification information, during a search and before reading the material. Of course, authors should be asked to supply the “div.class” on divisions that are not the same class as their enclosing divisions and where the “div.class” cannot be inferred from the elements used.

3.3.5 Using the “div.class” Attribute in Converting Legacy Data

In converting existing documents to PCIS SGML, users are finding that the existing documents do not contain much of the information that could ideally be in PCIS documents. For example, it is impossible to create a complete Sources from existing print documents; much of the information that should be in them does not appear in print.

However, the information necessary to assign “div.class”, at least at the top level or two, is available in most print documents. The information class can be inferred from the section or chapter headings and from the type of information contained in the document.

We suggest that, even for documents being converted to PCIS at a fairly superficial level, the effort to provide “div.class” will be repaid by increased usability of the documents.

3.3.6 The Values for “div.class”

3.3.6.1 Product Summary Information

Product Summary Information provides a product capabilities overview, a marketing consolidation, or a brief description or summary of the more detailed technical specifications of a product. Such summary information is more general than Detailed Product Specifications and is also highly variable, depending on the product and the marketing message to be delivered. Typical Product Summary Information may include a list of features, the description of the product's family, a statement of benefits, a list of typical applications, or a marketing position statement.

3.3.6.2 Detailed Product Specification Information

In contrast to the Product Summary, Detailed Product Specification Information describes the operation of a product, its measurable characteristics, and the appropriate limitations and conditions on its use. Detailed Product Specification Information may include ratings; mechanical, architectural, and performance characteristics; specific description and usage; process information; physical dimensions; etc.

3.3.6.3 Application Information

Application Information describes possible uses of the product, ranging from general use to specific end-applications such as an actual engineered solution. General use applications can encompass information concerning the product family (such as CMOS or Op Amps) or areas of application (such as amplifiers or video).

3.3.6.4 Safety and Environmental Compatibility Information

Safety and Environmental Compatibility Information relates to the safe handling and environmental impact of the manufacture, shipping, use, and disposal of the product and its package and packaging.

3.3.6.5 Information Concerning Support Tools

Support Tools describes hardware or software not directly part of the product but a useful or necessary supplement to the product. Information Concerning Support Tools may include descriptions of products that are necessary for proper part operation, models, evaluation kits, supporting software, or related hardware such as alignment tools, heat sinks, and sockets.

3.3.6.6 Reliability Data

Reliability Data demonstrates the ability of a product to perform a required function or specifies the change in performance-level of a function under stated conditions for a stated time period. Reliability as used here is a measure of some quality of a product remaining after the product has been exposed to some operating stress for a stated time period. Reliability Data includes information concerning product durability where failure is not predicted.

3.4 How the Pinnacles Component Information Standard may be Used

The PCIS DTDs are not particularly appropriate for the work of document creation. To create and display documents, data creators may and probably should, at their discretion, create authoring-sets based on the PCIS DTDs. These authoring-sets may:

- Require the presence of specific data elements;
- Specify the sequence of elements; and/or
- Include additional elements for internal use (any such elements must be removed before interchange). It is expected that such information will include elements for tracking the authoring and validation process and for information that is company confidential.

These authoring-sets are used to create documents that are, at least as exported, fully conformant to the Pinnacles Component Information Standard DTD(s). There is no need to transfer the specific authoring-set DTD with the document(s) created by it.

Similarly, viewing, retrieval, and search and display applications create display-sets based on the PCIS. In order to load display tools, the PCIS form of the data must be automatically transformed into a form that is optimized for the particular search and retrieval or display system. Such a display-set might:

- Sequence parts of the data in consistent formats;
- Add additional tagging (such as “container” tags around structures to be considered one element for formatting purposes);
- Transform the form of some elements (such as making a different element to correspond to each value of the type attribute of highlight);
- Remove parts of the information that are not relevant to a particular application.

4 The PCIS SGML Architecture

This section of the tutorial makes extensive use of SGML terminology and assumes the reader's familiarity with these SGML terms. Although the glossary in Section 11 includes a few SGML terms, the reader unfamiliar with SGML may wish to read only Collections of Document Elements.

4.1 Modularity

Until the early 1990s, most SGML applications have described each publication type with a single monolithic DTD. The DTDs for similar publications would contain similar or even identical element declarations. The PCIS is using a more modern, modular approach to DTD construction. There are four PCIS DTDs; each actually contains a few element definitions and a collection of references to smaller entities that the PCIS calls “modules.” Each module has been assigned a formal public identifier and is described in an SGML PUBLIC entity declaration.

There are many advantages to a modular system. The smaller units can be written once and maintained in a single place. Lower level data structures can be kept consistent across document types unless analysis identifies real differences between them. New PCIS DTDs can be built quickly, since many of the necessary components will already be defined within the DTD library.

There are four PCIS DTDs:

- PCIS Databook – The definition of a databook;
- PCIS Datasheet – The definition of all types of datasheets;
- PCIS Application Note – The definition of an application note; and
- PCIS Document or Document Fragment – The definition of a miscellaneous document, such as an introduction to a databook, a selection guide, or other documents that may be included in a databook.

Each DTD has been given a formal public identifier, prepared according to ISO 8879. Section 11 of this Tutorial contains a list of the PCIS formal public identifiers and an explanation of how they are constructed and what their segments mean.

There are also five entities which are referenced by each other and by the DTDs. Each module contains a logical subset of the PCIS data elements; for example, one module describes the elements necessary to define a table (a structure of rows and columns). Each module has also been given a formal public identifier prepared according to ISO 8879, so that the modules can be easily incorporated into DTDs by using SGML entity references.

PCIS-compliant applications use both the assembled DTDs and the modules from which they are assembled. For example, the Sources module is incorporated by reference into the Datasheet DTD. A person developing an input system for characteristic tables may want to use the Sources entity, or even part of the Sources entity, without the rest of the structures. Similarly, a developer constructing a company-specific PCIS authoring-set DTD may want to create a different superstructure than that defined in the PCIS Datasheet DTD, replacing some modules and incorporating others.

The PCIS includes the following modules:

Name	Module Name	Module Description
Administrative Information	PADMIN.ENT	Information concerning a document (meta-information), such as the Title, Marketing Date, Literature Order Number, Author, etc. All DTDs use at least a subset of this information.
Basic Elements	PINNBASE.ENT	Basic text structures shared by all the PCIS DTDs, including, for example, Paragraph, Ordered List, Footnote, etc. All the common SGML parameter entities and attribute lists are defined and identified here.
Character Sets	PCHARSET.ENT	The PCIS DTDs include by reference many special characters such as letters of the Greek alphabet, special math characters, and publishing symbols. Where possible, PCIS references these special characters as defined in the ISO PUBLIC entity sets defined in ISO 8879, the SGML standard. Where the ISO sets are incomplete, this module includes PCIS-specific SDATA entities to define the characters. Individual applications must make explicit the correspondence between these entities and actual character set printable characters.
Pinnacles Sources	PSOURCES.ENT	This module contains the elements for information added to the basic document text, such as computer-sensible models, the full structure of parametric data such as characteristics and conditions, information on support tools, detailed connection and terminus information, and detailed product identification information.
Tables	EXCHANGE.ENT	This module, published by SGML Open, contains the elements that define a table (a structure of information presented in rows and columns). (The decision to use an existing table model rather than create a new one is discussed in Section 4.12).

4.2 SGML Parameter Entities

SGML parameter entities (defined in the glossary of this tutorial) are used extensively in the PCIS DTD modules. Parameter entities have been defined for the most common element content models and attribute Declared Value lists, both of which are likely to be changed in authoring-set DTDs. Using parameter entities for these models allows the modules to be used in authoring-sets or display-sets without modification to the original PCIS module, since the creators of authoring or display sets can simply redeclare the value of a particular parameter entity as necessary.

Parameter entities have also been used to name attribute list subsets and content models that have been used repeatedly throughout the DTDs, to keep similar declarations synchronized. Parameter entities have also been used to group elements that are treated in the same way within content models.

4.3 Collections of Document Elements

It is useful to group elements into categories by the way they are used and the places in which they can occur. These groupings are used in the DTDs, and are discussed in the Tag Library. There are many elements that may contain any of the elements in a grouping, or that may appear in any of the elements in a grouping.

PCIS calls these element groupings "pools" (after IBM and OSF). Pools are a way of grouping elements that may occur at the same hierarchical level, such as all the division-level elements. PCIS recognizes five hierarchical levels, listed here from highest to lowest:

- Division-level Elements;
- Paragraph-level Elements;
- Intermediate-level Elements;
- Inside-a-paragraph-level Elements; and
- Phrase-level Elements.

It is most useful to explain these hierarchical levels in reverse order: Phrases are small elements (for example, a Cross-Reference) that may occur anywhere in text: inside a Paragraph, inside aTitle, or inside a person'sName. Inside-a-paragraph level elements are, at least theoretically, larger and more complicated than simple phrases; they may have internal structure. Such an element (for example, an Address) could easily occur inside a Paragraph but could not occur inside a person's Name or inside a Title. Intermediate elements may occur within a Paragraph or at the same level as a Paragraph (for example, a Figure or Table). Paragraph-level elements are hierarchically at the same level as a Paragraph (such as a Caution, Ordered List, and Paragraph). Division-level elements include the General (or Generic) Division element and other elements that form major sections of document.

Within these five levels, PCIS makes additional distinctions and creates additional groupings, such as "All the Generic Phrase-level Elements" and "All the Technical Phrase-level Elements". "All the Generic Phrase-level Elements" is a collection of non-technical, general-purpose phrase-level elements (for example, Cross-Reference and Superscript) which may be used anywhere in text. "All the Technical Phrase-level Elements" is a collection of specific technical phrase-level elements (for example, Package Name and Inline Math) that are more appropriately used in technical sections, not in general text.

The purpose of this fairly complex grouping structure is to limit which elements can be used in which locations, for example inside other elements. The major PCIS pools are defined below; a more complete discussion of the pools is given in the Introduction to the PCIS Tag Library.

The major PCIS element groupings are:

Pool Name	Pool Content	Pool Description
div.elems	Division-level elements	Divisions are hierarchical structures that usually begin with a title and which may nest. General divisions will contain any content desired. Some specifically named divisions are predefined to contain specific types of content to aid in retrieval. Examples include Architectural/Functional Description and Product Mechanical Information.

Pool Name	Pool Content	Pool Description
all.para	Paragraph-level elements	These are elements that may occur at the same structural level as a paragraph, i.e., that may be used anywhere a paragraph could be used and which are not subdivided hierarchically into divisions. Examples include Paragraphs, Acknowledgments, Code Listings, Figures, Note in Texts, Unordered Lists, and Warnings.
gen.para	Generic paragraph-level elements	These are elements that may occur at the same structural level as a paragraph in contexts in which technical and administrative paragraph-level elements are not allowed. This is often the case inside relatively small structural elements such as function tables and acknowledgments. Examples of generic paragraph-level elements include Paragraphs, Figures, and Unordered Lists.
all.phrase	All phrase-level elements	These are text or phrase-level elements that occur only within other structures such as Paragraphs or List Items. This includes such elements as hypertext links and Cross-References which float and are not tied to a particular structure, as well as such technical phrases such as Inline Math and Package Designators.
gen.phrase	Generic phrase-level elements	These are text or phrase-level elements that can occur anywhere text can occur, and include such elements as hypertext links and Cross-References.

4.4 Recurring Content Models

There are two division content models that are used in many places in the PCIS DTDs, and that are often referred to by name:

tech.div.model	Technical Division Model	This is the model for all divisions that may contain technical information. These divisions may begin with a Title, Table of Contents, Author Information, and Abstract. This may be followed by anything in the paragraph-level elements pool, followed by anything in the division-level elements pool.
gen.div.model	Generic Division Model	This is the model for all divisions that contain only generic information, not named technical information elements. These divisions may begin with a Title, Table of Contents, Author Information, and Abstract. This may be followed by anything in the generic paragraph-level elements pool, followed by Generic Divisions.

4.5 Monastic DTDs

“Monastic DTDs” is an idea proposed by Wayne Wohler and Eliot Kimber (et al.) at IBM and adopted as part of the PCIS SGML Architecture. A monastic DTD eliminates certain features of SGML to facilitate re-use of document fragments and to help remove sequential processing bias from the SGML data stream. The idea is to ensure that the content of a document fragment can be verified without

knowing every context in which it may occur, by making every element fragment a proper subtree of the document. Thus, SGML constructs that change the content of an element, or modify the recognition of delimiters depending on an element's parent, are prohibited.

The rules for monastic DTDs in the PCIS DTDs, modules, document instances, and SGML Declaration include:

- No exceptions in the DTDs (this rule does not affect the table model, since EXCHANGE.ENT is an adopted DTD module);
- Short Reference Maps not permitted;
- Declared value #CURRENT not used; and
- Marked Sections (with IGNORE and INCLUDE) not written in the PCIS DTDs. All marked sections in document instances must be resolved before interchange.

Exceptions are prohibited to enable the use of partial documents and document fragments. In a DTD, inclusions and exclusions are inherited down the element hierarchy. This means that if an element is included at a high level, it is permitted at all lower levels unless excluded. Similarly, if an element is excluded at a high level it is not permitted at any lower level. This causes problems when considering portions of documents without their full context. In an application that uses inclusions and exclusions, it is impossible to fully validate partial documents without full context; for example, a structure that looks valid may in fact have been excluded at a higher level. This is rarely a problem in documents that are always created in rigidly controlled environments and that are published and used as whole documents. In environments where partial documents may be created, or cut and pasted, and in which data recipients may be expected to disassemble the documents, the use of exceptions could cause difficulties. Therefore, within the PCIS, inclusions are prohibited, and exclusions are used only by SGML Open's version of the CALS Table model.

4.6 Mixed Content versus Inclusions

The PCIS DTDs make extensive use of mixed content models but only when all elements or groups in the content model are connected with "OR" and when the entire model is repeatable. The following paragraphs provide the rationale for, and alternatives to, the mixed content decision.

SGML defines three distinct types of element content:

Character Content An element contains data characters (#PCDATA);

Element Content An element contains nothing except other elements;

Mixed Content An element contains both data characters and elements content, in any order.

Mixed content models in which the entire grouping is not repeated can cause problems because of the way record ends are treated in mixed content. Some SGML products enforce the letter of the standard and come to a full halt when record ends occur where prohibited by a mixed content model. Thus, improperly used mixed content can be a major handicap to interchange.

However, there are certain types of problems which seem most easily solved using a mixed content model. Consider the problem created by having phrase-level elements within paragraphs:

A mixed content model would define such structures as:

```
<!ELEMENT  para  - -  (#PCDATA | phrase)* >
```

A second solution to this problem would be to define phrases as inclusions on the paragraph level. Exceptions would then be placed at as low a level as possible; generic structures such as footnotes and formatting elements such as line breaks would be placed at the document level:

```
<!ELEMENT  para  - -  (#PCDATA) +(phrase) >
```

Generally, this model would not be permitted in PCIS DTDs since they are constructed as monastic DTDs which allow no inclusions.

A third solution that uses neither mixed content nor inclusions to create phrases within paragraphs would create a text “wrapper” to contain all #PCDATA. Thus, #PCDATA is never uncontained:

```
<!ELEMENT  para      - -  (text | phrase)* >

<!ELEMENT  text      - -  (#PCDATA) >

<!ELEMENT  phrase    - -  (#PCDATA) >
```

The tagging necessary to accommodate these models can be accomplished even without the knowledge of any of the users. In practice, however, current SGML products tend to make this sort of tagging obtrusive and unwieldy. The additional text wrappers result in an unreasonable amount of tagging; they interfere with authoring as well as with the search and retrieval processes.

To prevent interchange problems, the PCIS DTDs use mixed content only in models in which all items in the model are connected with “OR” and the entire model is repeatable. In practice, even in Version 1.4, the content model for Contact Information can still present a mixed content problem to the unwary. This item will be fixed in a future release of the PCIS DTDs.

4.7 Architectural Forms and Meta-DTDs

Many large SGML applications are using the concepts of Architectural Forms (introduced in the HyTime standard) and Meta-DTDs. These are mechanisms to allow the users of the SGML application standard to create their own DTDs and create them in predictable ways. The PCIS has taken a different approach; it includes not the building blocks for DTDs but a very permissive DTD. The PCIS expects users to use the DTDs as published to model the information to be interchanged.

Because there was some discussion during the PCIS Architecture meetings of architectural forms and meta-DTDs, and because future extensions of the PCIS may use these techniques to expand the scope of the PCIS, they are discussed briefly below.

HyTime [ISO/IEC 10744:1992(E)] is an international standard for hypermedia documents. HyTime describes mechanisms for locating points within documents and document collections, for synchronizing the rendition of documents, etc. The HyTime meta-DTD describes architectural forms to be used as models for all the elements of a DTD. Architectural forms are a mechanism for communicating what DTDs should/must contain, and are promulgated by a number of applications to ensure that their applications can correctly process information encoded by a wide variety of DTDs. Architectural forms are thus the building blocks of meta-DTDs according to which DTDs are developed, just as DTDs are, in a sense, meta-documents according to which documents are developed. The HyTime standard defines a meta-DTD, to which a real DTD would need to conform, and other meta-DTDs have been written.

The current PCIS is not an effort to create a meta-DTD or to adopt the current HyTime meta-DTD. PCIS has been an effort to create an SGML DTD or set of interlinked DTDs to represent specific information types. The RDD workshops made it clear that there was not enough information available concerning the abstract structure of databooks and datasheets to construct or determine the applicability of a meta-DTD at this time.

The exact relationship between the PCIS, HyTime, and other meta-DTDs is a possible task for a future PCIS efforts. A meta-DTD specific to electronic component information could provide the tools necessary to define more precisely the datasheets and databooks that are the subject of this effort and other related information in use throughout the industry. The current DTDs strive to be as HyTime

enabling as possible, in a style permitting this future growth. HyTime instructions have been placed at the end of the SGML Declaration and all links and references have been implemented in the style of HyTime.

4.8 Processing Instructions

It is essential that SGML applications stay within the letter and spirit of the ISO 8879 standard. Only if applications are simple and in strict compliance with the standard can the benefits of SGML be realized. For this reason, PCIS recommends against the use of processing instructions. In SGML terms, a processing instruction contains system-specific information that describes how a document is to be processed (for example, how to display the document on a screen or print the document on paper). Although processing instructions are legal within SGML, they usually represent non-SGML ways to get ink onto paper correctly. By definition, since they are instructions to a specific system, they are non-interchangeable and contrary to the spirit of SGML.

Processing instructions will be used in the PCIS DTDs only to accommodate HyTime instructions. Processing instructions in PCIS document instances are strongly discouraged.

4.9 SGML Declaration and Feature Use

The SGML Declaration is a mechanism for specifying the capabilities and capacities of the system needed to process an SGML application's documents. The Declaration sets the ground rules for the DTD. The SGML Declaration specifies which SGML features are to be used, selects character sets, and provides a maximum capacity for a number of things which may affect the receiving or processing systems. The SGML Declaration provided with the PCIS is to be used for all the PCIS DTDs and documents.

4.9.1 SGML Features

The PCIS utilizes only a few of the features of SGML. It is anticipated that information creators and information consumers will use far more SGML features in their internal applications, but the interchange form has been kept as simple as possible. This will allow systems of varying degrees of SGML functionality to create and receive PCIS documents.

PCIS uses the following SGML feature:

FORMAL The PCIS architecture has developed formal public identifiers for the PCIS DTDs, DTD modules, and document instances, in anticipation of the creation of a registry for SGML Identifiers. The PCIS formal public identifier structure is described in Section 11.

PCIS does not use the following SGML features:

DATATAG The DATATAG feature allows some of the characters in the document instance to be simultaneously interpreted as markup and as content. The feature is rarely implemented because of the possibility of errors and inherent risk.

OMITTAG The OMITTAG features allows the omission of the tags that start or end an element in the document instance. While this feature is frequently valuable in authoring applications, its use in interchange requires that the receiving application be able to correctly infer the location of all omitted tags. The cost of not using OMITTAG is that interchanged files will be larger than is theoretically required; this is a small price to pay for the added simplicity of usage.

RANK	The RANK feature is a mechanism to create numerically ranked hierarchical structures from the data context. While information creators may choose to include RANK in the systems that create their information, the ranks, if any, must be explicit for interchange.
SHORTTAG	The SHORTTAG feature of SGML allows the omission or abbreviation of a number of SGML tags. This is not permitted in the interchange for the same reasons OMITTAG is not permitted.
LINK	The LINK feature is not permitted in any of its forms. There is no reason to link from an interchange document to processing instructions, which is the most common use for LINK.
SUBDOC	The SUBDOC feature allows the incorporation of SGML documents that are encoded according to other DTDs. While this is a powerful feature, it means that the receiving system cannot anticipate what structures may be present in the SUB-DOCUMENT.
CONCUR	The CONCUR feature allows the use of multiple DTDs to simultaneously describe the same document instance. It is possible that data receivers may wish to use CONCUR to super-impose their preferred structure on received data as well as the structure intended by the creating organization. There is, however, no need for this feature as part of the PCIS.
Delimiters	The PCIS SGML Architecture uses the Reference Delimiter Set for general delimiters. Although there are conflicts with some of the characters in it, and thus entities must be used in the data instead of the keyboard characters for "<" (MDO) and other SGML delimiter characters, the same problem would exist for any character set. Most SGML applications assume the reference delimiter set and there is no good reason not to use it.

The PCIS SGML Architecture also uses the Reserved Names given in the Reference Concrete Syntax, again both because there is no good reason to change it and because most SGML applications assume that it will be used.

The SHORTREF delimiter is set to NONE, so that SHORTREFs are not available in the interchange form.

4.9.2 Quantities and Capacities

The PCIS has declared all quantities and capacities to be arbitrarily high, large enough to accommodate very complex DTDs. Systems capable of creating or interpreting PCIS documents need to be capable of handling large, complex documents. Therefore, it is not necessary to constrain the data creator's ability to create complex nested documents by limiting the SGML quantities or capacities. If at any time a quantity or capacity constrains a person who maintains a PCIS-compliant application, that limit should be expanded.

4.9.3 Character Sets

SGML assumes that one relatively small character set will be used within an application. Any characters required in the document that are not in the basic character set are represented using combinations of those basic characters in constructs called entities. Most SGML applications use the ASCII character set (ISO 646), and so does PCIS.

The entities for the characters not available in this character set must be named and agreed upon for an application. The International Standards Organization (ISO) has identified a large number of special characters, many of which are listed in ISO 8879. The standard ISO entity sets used in the PCIS are referenced in the Pinnacles Character Set module, PCHARSET.ENT.

PCIS applications need additional special characters that are not in the standard ISO character sets. For PCIS, SDATA entities for these characters are incorporated into the PCHARSET.ENT module. Additional characters will be added at the request of organizations implementing the standard. At the time that the PCIS is turned over to a formal standards body, it is recommended that the issue of character sets be revisited and the added characters be submitted to ISO.

Creators of authoring-set or display-set DTDs should understand that the SDATA references included in the standard PCHARSET.ENT module will not permit the display of any character glyphs being referenced. Applications which wish to use actual character glyphs for display will need to amend the PCHARSET.ENT.

4.10 Marked Sections

The SGML Marked Section capability allows portions of a data stream to be included or excluded from the processing stream of an SGML application. Marked sections are typically used to create multiple documents that have related, similar, or highly redundant contents. Since Marked Sections are allowed to cross structural boundaries, using them can create documents which cannot be parsed. The interpretation of Marked Sections is dependent upon the processor and correct interpretation of the inclusion/exclusion mechanism. While there is no way in SGML to prohibit the use of Marked Sections, their use is PROHIBITED in PCIS. This means that any Marked Sections used in document creation MUST be resolved before interchange.

4.11 Markup Consistency

Markup in the PCIS DTDs has been used consistently across contexts. That is, the same element name has been used for a structure that can occur in many places, regardless of the location. For example, there is only one element for Title, not many separate elements such as Division.Title, Table.Title, Figure.Title, etc.

4.12 Use of Existing SGML Application Standards

An SGML application standard specifies how an organization or group of related enterprises will implement SGML. By selecting an SGML application standard that comes close to meeting the current need and customizing it, instead of creating an SGML application from scratch, several benefits may be realized:

- Application development may be faster;
- Documents may be more easily interchanged with other organizations;
- Commercial services such as typographers may find that training and start-up costs are reduced; and
- Commercial utilities (such as automatic tag insertion software) may be available to assist in conversion to and use of the new application.

4.12.1 Selection of an SGML Application Standard

Several well-known SGML application standards were examined as possible bases for the PCIS. The Department of Defense's MIL-M-28001B is the CALS standard for SGML. The Association of American Publishers (AAP) Electronic Manuscript Standard for Electronic Manuscript Preparation and Markup is an ANSI/NISO standard, and is used by many commercial publishers. The MAJOUR Header application is an SGML application standard for bibliographic information about journal articles. The Air Transport Association's ATA/AIA SGML standard is a variation on the CALS standard for SGML, and is being used for aircraft maintenance information. HyTime, the SGML standard for multimedia documents is also an SGML application standard. Each of these application standards conforms to ISO 8879, the international standard for SGML.

Many users find that by using an SGML application standard, they can save time and money in developing an SGML application and produce DTDs and applications that closely resemble those of organizations with which they interchange documents.

IBM developed an architecture for technical documents that is called the InfoMaster Architecture with a DTD called the IBMIDDOC DTD. This application was developed specifically to accommodate interchange among disparate systems and re-use of information. Since these are important requirements for the PCIS effort, the IBM model was chosen as a base. IBM has provided materials and advice on the use of their application, and has granted the rights to use it provided their copyright is cited and their name is included in all uses of the material.

The PCIS SGML Application has adopted as many of the IBMIDDOC elements as are appropriate. For example, the definitions of lists, list items, etc. have been adopted from IBMIDDOC. More importantly, some design principles, such as what IBM calls the principle of Monastic DTDs, have been adopted. Monastic DTDs are designed so that their structures are context-independent. This means that a piece of text, such as a section, can be copied from anywhere in the document and pasted anywhere else in the document, and if the top level element is valid all of the contents will be. (See Section 4.5.)

4.12.2 Structural Tables

In SGML terms, the words “structural table” or “structure-based table” refer to the way the individual cells of information in a table are tagged. Structure-based tags indicate that the table consists of rows, columns, and row or column spans. Structural table tags are completely unconcerned with the content of the table. For the encoding of tables, the PCIS uses an existing structural table model, the SGML Open Exchange version of the CALS table model, with minor modifications.

There are two uses for structure-based tables in PCIS documents. First, there is a wide variety of tabular information that is used one time only, is unique to one document, or that occurs in a limited number of documents. This information will be treated as structural tables and tagged as rows and columns. Secondly, there are a few categories of information that will be available as content-based markup (in the source) and also mapped to a structure-based tabular representation. All tables in PCIS use the same structural table model to identify the way the table should look. The structural table model is used to describe the columns, rows, headers, and the appearance of all tables.

Two approaches to structure-based tables were considered: the creation of a new table DTD fragment or the adoption of a public one (note: the classic “Make or Buy” decision). There are a number of public table models available; none is perfect, and none is endorsed by industry consensus. The Pinnacles Group nonetheless felt that it would be a great disservice to create a new table model for the PCIS application. A new table model would:

- Be unnecessarily expensive to implement because no off-the-shelf tools could be used;
- Have unknown implementation problems instead of the known problems with the published table models; and
- Unnecessarily complicate the development and maintenance of the standard.

A wide variety of table models is available. In developing the PCIS there was no intent to examine all possible, or even all known, SGML table models. The public (and/or published) table models that were evaluated at the time PCIS 1.0 was drafted include:

- CALS (MIL-M-28001A and MIL-M-28001B);
- SoftQuad tables (based on the AAP Complex tables);
- Exoterica table model (based on the AAP Complex tables);
- AAP simple and complex tables;
- ISO 12083 table models (the updated, edited version of the AAP model);
- ArborText table model (based on the AAP Complex tables);

- TEI Draft Table model; and
- The GRIF Table model.

The CALS table model was the most widely implemented of these table models and seemed to be adequate for the vast majority of the tables in datasheets and databooks. The advantages of selecting the CALS table model were:

- It has been widely implemented, so software capable of handling it is available;
- It is a reasonably robust structural model; and
- It is not perceived as favoring the tools of one vendor over another.

Disadvantages of using the CALS table model included:

- It is extremely tag intensive;
- It is more complex than many table models, almost requiring the use of special table editing tools to create;
- It is completely format-oriented and allows inclusion of content-type information only in a roundabout fashion;
- The maintenance is controlled by an organization with which the Pinnacles Group has little, if any, influence;
- Its specification is interpreted differently by various SGML tools; and
- The model has no mechanism to handle:
    - Stubs, and stub hierarchies;
    - Total rows;
    - Titled subgroups within the table;
    - Table descriptions; and
    - The distinction between table footnotes and regular footnotes referenced from within a table.

Because the CALS table model was so widely adopted, and because it came very close to meeting the identified needs of the PCIS, it was adopted (with minor modifications) for use in versions 1.0 through 1.2 of the PCIS.

In 1995, SGML Open, a consortium of SGML tool vendors, researched the assumptions made about the structure of the CALS Table model supported by SGML Open members' products. SGML Open then proposed a modification to the CALS table model which used only those features of the CALS table model generally supported by member companies' products. The SGML Open version is accompanied by detailed documentation which clarifies many of the significant areas of interpretation of the CALS table model on which various tools disagreed.

In 1996, SGML Open issued an exchange version of the modified table model (in SGML Open Technical Resolution TR 9503:1995 modified 1996 May 8). The exchange subset was designed such that "if an application's tables are tagged according to this subset, there is a high probability that the table will be interoperable among the great majority of SGML Open vendor products."

As SGML Open vendors produce the vast majority of SGML tools likely to be implemented by PCIS users, and because the SGML Open table model is generally agreed to be superior to the CALS table model on which it is based, PCIS 1.3 and PCIS 1.4 use the SGML Open Exchange Table Model.

The SGML Open Exchange Table Model allows only text in table cells. However, for PCIS 1.3 and later versions, the table cell model has been expanded to include phrase-level elements, including Reflection, to meet the minimum needs of PCIS.

5 Document and Element Identification

Note: This section is intended for PCIS application developers and may be skipped by readers with little knowledge of SGML.

The PCIS includes required IDs for virtually every element. This ensures that applications can point to any element from another place either in the document or in another document, and that these cross-references can be stable across versions of the document. This also means that the SGML files will be very large and very difficult to create without specialized software.

Identification of electronic documents and parts of electronic documents is critical to their use in any application more sophisticated than simple page-turners. In a full-function SGML application, there must be the ability to uniquely identify a document itself as well as the parts of the document that may be pointed to (either from within the document or by another document). The PCIS source structures defined briefly in Section 6.1.1 and in more detail in Sections Section 6.2 through Section 6.8, as well as document revision control, require the extensive use of unique identifiers.

5.1 Document Identification

At the highest level, a document management or exchange system must be able to identify a document clearly and unambiguously. A link cannot be correctly followed within a document unless the document can be identified. For this reason, PCIS recommends that every document, and each published version of each document, be uniquely identified with a formal public identifier. The formal public identifier will include a reference to the Pinnacles Component Information Standard, identification of the company that owns the document, and an identifier for the document itself. Section 11 describes the construction of PCIS formal public identifiers.

The PCIS DTDs and DTD modules also are identified with formal public identifiers which are listed in Section 11.

5.2 Element Identification

In electronic documents, virtually any portion of the document may be referenced. Therefore, PCIS DTDs require an ID on all structural elements (such as Paragraphs and Generic Divisions) and all elements within the data sources (such as Characteristics Source, Terminus, and Connection Source). In fact, a unique identifier is required on all elements unless there seems to be no possible reason for the identifier (for example, there seems to be little value to uniquely identifying a forced Line Break). This enables extensive cross-referencing and hypertext linking for future purposes.

5.2.1 ID Stability

PCIS **strongly** recommends that IDs remain stable across varying versions of the document, and bases its revision history mechanism on the assumption that IDs will remain stable.

In an environment where documents are constantly updated and documents point to parts of other documents, it is highly desirable for IDs to remain stable across update cycles. For example, when IDs remain stable, a link pointing from one document to a DC Characteristics Table in another document will remain even if a value in the table is modified.

Unfortunately, there is no way for the PCIS standard to enforce the requirement that IDs remain stable. It is the task of the PCIS application authoring systems to ensure that IDs are stable and that they are not re-used within the same ID namespace, even if a section is deleted or moved.

5.3 Cross-References and Hypertext Links

Within a datasheet or databook there may be several types of cross-references:

- References to a diagram or figure or other non-SGML object (for example, “See Figure 3”);
- Explicit cross-references to other parts of the document (for example, “See Section 3.2”);
- Footnotes and note references;
- References to other documents outside of the current one (for example “according to Appendix D of MIL-M-28001B” or “as explained in Application Note 365A”); and
- References to structures within the datasheet, such as a pin, circuit block, command, a standard (national or international), glossary term, etc.

The PCIS DTDs handle all these references with a small set of cross-reference links: one type of link is used for all internal cross-references that are not footnotes, one type is used for external cross-references, and a third type is used for footnote references.

Hypertext links are a mechanism to associate information in one location or document with information in another location or document. These links, which can be used as active electronic cross-references are used by many software systems to create active books. The reader identifies an interesting link and then “traverses” the link to the associated information.

The PCIS uses two of HyTime's link types, ilink and clink. Ilinks (Independent Links) are HyTime multi-way links that may point in more than one direction. Clinks (Contextual Links) are HyTime single-direction links that work similarly to cross-references, that is, they point from one particular place in the document to somewhere else. The PCIS <ilink> and <clink> correspond directly to the HyTime architectural forms. In addition, the three types of PCIS cross-references use the clink architectural form.

Currently, there are very few advantages to using HyTime links for such references, but there are also few disadvantages. There are no advantages because there are almost no HyTime engines currently in general use. There are few disadvantages because the current hypertext tools are capable of ignoring the HyTime-nature of the information and continuing to use the ID/IDREF structure as it appears in the documents. To enable hyperlinking for purposes that need not even be realized now, all content and structural items have required identifiers.

Real HyTime processing requires the addition of the APPINFO parameter to the SGML Declaration and the insertion of some “HyTime processing instructions” in the DTD. The processing instructions for the most basic simple HyTime implementation have been added to the end of each PCIS DTD. The APPINFO assertion has been included in the PCIS SGML Declaration as a comment.

6 Structured Data

6.1 Source and Reflection

6.1.1 Source

PCIS documents contain a great deal of information which can be described in a highly structured way. This includes information on product characteristics, the conditions under which parameters were measured to produce the characteristics, information on the product's terminuses and connections (such as pins), and product identification information.

There are two distinct and sometimes contradictory goals in the electronic description of such information for databooks. The first goal is good retrieval access and support for parametric searching. The second goal is to provide an arbitrary display of the information on display media such as paper or a CRT screen. While the ability to search on the structured information and to load this information into databases and other tools is critical to the success of ECIX, the ability to create the current print products and electronic views similar to the current products must also be supported.

To meet these potentially conflicting requirements without redundant storage of the information, the PCIS architecture provides a two-part structure for such information; a highly structured “Source”, which holds the structured data, and optional “Reflections”, which can display information actually stored in one of the Sources in the body of the document.

The Sources contain the versions of data that will be used for search and retrieval and for loading databases. One or more Reflection(s) will be used to print or display the data. A Reflection DOES NOT CONTAIN the data, it contains pointers to the data in the Sources. This way, if the information in the Sources is updated, the information displayed or printed as part of the document text will be automatically changed too. Similarly, if a Characteristic Value that is referenced four times in a datasheet is updated, there is no need to find and change all four occurrences, only the Characteristic Value in the Characteristics Source need be changed.

For each Characteristic, the Characteristics Source will contain the hierarchical structure (Parameter, Conditions, Characteristic Value, etc.) as described in Section 6.2. The Characteristics Source will include full descriptions of all characteristics, including information that historically has been occasionally but not consistently displayed (such as Connection Identifiers). In addition, the Characteristics Source can identify for each Characteristic the characteristic's aspect(s), the “Product ID”(s) to which the characteristic applies, and an indication of whether the characteristic is“ Reliability” data (or not). The Characteristics Source will also contain an indication of whether or not the Characteristic is “Static”. In summary, the Characteristics Source will contain all the characteristics information that has been described in this section. Each structural piece of the Characteristics Source, such as a Parameter Symbol, will be assigned a unique identifier.

6.1.2 Reflections

Reflections are used to display information that is stored in one of the Sources in the body of the document. Reflected information will look to the end user of the document as if it were replicated at the reflected location, but the data must not actually be duplicated.

The body of the document will contain text, lists, and tables, just as documents do now. Any of these structures may contain reflections of information stored in the Sources. The most complex, and the most common, place that Reflections will occur is in characteristics tables. PCIS uses the SGML Open Table Model for all tables. In order to create a characteristic table (for example the AC Characteristics Table) the user would:

- Create an ordinary structural table (that is, a table with the appropriate number of rows, columns, column widths, etc.);

- Insert all non-reflected text, such as column headings, into the table structure; and
- Reflect as much of the Sources as is appropriate into various cells of the table, including for example: Parameter Symbols, Characteristic Values, Conditions, and Characteristic Group Titles. Then the user may insert additional textual elements (equals sign, spaces, etc.) between Reflections.

All text structures, including, but not limited to, Titles, Paragraphs, List Items, and Definitions, may include Reflections.

Throughout the body of the document, wherever information that is stored in the Sources should appear, a <reflection> of that element as it is encoded in the Sources will appear. Reflections will be maintained as Reflection elements, and not treated as simple copies of the information in the Sources. When displayed to the user, the contents of the part of the Sources that has been reflected will appear to be at the location of the Reflection in the document, so that casual browsers will not know that the data is not actually present in the body of the document.

For example, if the Characteristics Source contains:

```
<cc id="54334" pids="F234″ static="0" reliability="0" acoustic="0
chemical="0" electric =″1″ info="0" magnetic="0" mechan="1" optical="0"
thermal="0"><parm id="54335"><parm.symbol
id="54336">X<sub>YZ</sub></parm.symbol> </parm> <cc.value id="54337"
value.type="MIN"> <number id="54338">12</number> <order.of.mag
id="54339">M</order.of.mag> <unit id="54340">Goods</unit> </cc.value>
<cc.value id="54341" value.type="MAX"><number
id="54342">43</number><order.of.mag id="54343">M</order.of.mag><unit
id="54344">Goods</unit></cc.value> </cc>
```

And a paragraph of text in the SGML-encoded document might be:

```
The whatsit runs at a minimum of <reflection refid="54338"> <reflection
refid="54339"><reflection refid="54340"> and is reliable up to <reflection
refid="54342"> <reflection refid="54343"><reflection refid="54344">.
```

That information could be displayed as:

```
The whatsit runs at a minimum of 12 MGoods and is reliable up to 43 MGoods.
```

A Reflection of a Characteristic Group is likely to include only a reflection of the Characteristic Group's Title. This is likely to occur either as a section title, a table title, or in a row within a table. That is, the Title of the Characteristic Group is likely to be reflected into the content of a Title element of a Table or of a portion of a table, such as a Table Row.

All elements in the PSOURCES.ENT module that contain (or may contain) character data and have IDs can be reflected using the Reflection element. A reflection of a source element that contains other elements (as opposed to an element that contains character data) is actually a series of reflections of all of the elements containing character data that occur inside the higher level element, in the sequence in which the elements occur in that higher-level element.

In the case of an element that contains mixed content (such as the Source Paragraph element, the Reflection should be of the element Source Paragraph because it can contain character data and has an ID. Any elements contained inside the element with mixed content (such as a Not inside a Source Paragraph should be displayed with their formatting consequences as part of the reflected text of the Source Paragraph but should neither be reflected individually nor treated as anything other than a integral part of the reflected text. The data reflected from a single element should be treated as a single stream of data for processing, regardless of how many distinct elements may have originally made up that data stream in the Sources.

Any of the Product IDs that can be created in the Sources, Generic Product ID, Specific Product ID, and Product ID Fragment, can be reflected into PCIS documents using the Product ID Reflection element.

6.1.3 Where Can the Reflections Occur?

The <reflection> tag, which reflects the elements in Sources may be used anywhere text may occur in the body of the document. It is most likely to occur in Titles, Paragraphs, List Items, Footnotes, and the titles and entries in tables.

6.1.4 Where Can Product ID Reflections Occur?

The <pid.reflection> tag, which is used to reflect any of the Product ID elements, may be used in all of the same places <reflection> can be used, but can also be used in any of the Sources other than the Product ID Source as well.

6.1.5 What Should Reflections Contain?

Applications may store and manage Reflections in any way that is convenient.

For authoring purposes, it is probably simplest in the SGML context if Reflection elements and Product ID Reflection elements are created and treated as EMPTY elements.

For interchange a Reflection must have an IDREF that points to the ID in the Sources of the reflected element. The Reflection element may either be EMPTY (in the SGML context) and have a pointer to the data in the Sources that is to be reflected; or it may be permitted to contain data, in which case it may have a replica of the content of the element in Sources that is being reflected.

It is safer to transfer empty <reflection>s and put the burden of populating them on the receiving system because then data is not duplicated and there is no chance of a conflict between the REAL Sources and the reflected data inside the Reflections. This advice, however, is contrary to the insistence in much of the rest of the PCIS that generated text, labels, and other such textual objects must be resolved prior to interchange. Reflections may be resolved prior to information exchange, whether to support such resolution or because it makes interactive browsing easier, but applications that make such resolutions must be prepared to settle conflict between the data contained in a filled Reflection and in the original Sources element. As a general rule, the data in the element contained in the Sources should be considered correct.

6.2 Characteristics

Characteristics (parametric data measurements) are among the most important information in datasheets. The PCIS defines a Characteristic as an inherent or measurable property of a product expressed as a value or values under stated or implied conditions. In current printed datasheets, a Characteristic is usually (though not always) expressed as a single row in a table of similar characteristics (such as DC Characteristics or AC Characteristics). In such a table, a characteristic is a particular Parameter, measured or specified under a known set of Conditions, yielding the results (Characteristic Values) stated. For example, the following table displays one characteristic, for which the Parameter Symbol is V_{IK}. (Note: The column headings merely define the parts of the characteristic and are not part of the characteristic data; the table row below the headings holds the data for one characteristic.)

Table 2. Sample Characteristic

SYMBOL	PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNITS
V_{IK}	Input clamp voltage	V_{ICC}=MIN I_I=-12 MA			-1.5	V

Complicating the issue of representing characteristics is the summary nature of current printed tables. In order to minimize the volume of paper publications and to show similarities and differences between products, only a subset of the characteristics information for any one component is usually published in a print product. Some information is never stated; it is implied. Other data is described once in a table or related paragraph, although it applies to many different characteristics. This means that a characteristic is only meaningful in context. Thus, the printed characteristics row shown above is insufficient to describe the characteristic completely. For example, both the product number and the conditions of operation must be known and are not shown.

6.2.1 Structure of a Characteristic

Characteristics are not always displayed in tabular form. Characteristics can be expressed as plots or graphs or described in textual paragraphs. However they may be displayed, all Characteristics follow the same conceptual structure:

(Note: bullets have been used below to indicate the hierarchical level)

- Parameter, containing at least one of:
  - Parameter Symbol;
  - Parameter Description; and
  - Parameter Definition;
- 0 or more Parameter Dictionary IDs, followed by
- 0 or more Condition Identifiers, followed by
- 1 or more Characteristic Values, each with its “value.type” attribute set, containing:
  - Number, Expression, or Range
  - Unit of Measure
  - Order of Magnitude
- Followed by zero or more Connection Identifiers, followed by
- 0 or more sections on Testing Information such as Test Circuit or Procedures, followed by
- Text objects including paragraphs, graphics, equations, etc.

The parts of a Characteristic are defined as follows:

Parameter — The name or identifying symbol of a product characteristic or of a condition on a characteristic. A full parameter consists of a Parameter Symbol (such as V_{IH}), a Parameter Description (a word or short phrase such as “Input HIGH Voltage” that names the Parameter), and a definition (usually a longer narrative explanation of the meaning of the parameter or a description of how the parameter is measured). When a given characteristic is displayed or printed, not all of the three parts need be shown.

Dictionary Identifier — The optional Parameter Dictionary ID indicates that a dictionary (or dictionaries) defined the Parameter. The ID is the identifier of the parameter in the named dictionary. An attribute on the Parameter Dictionary ID element allows an author to specify the name of the dictionary. Thus, if a Parameter is defined in several dictionaries, each of the dictionaries may be identified through the use of multiple Parameter Dictionary ID elements, and the ID of the Parameter in each dictionary supplied.

Condition	The state under which a product's behavior is measured or reported. A Condition has a structure that is similar to that of a Characteristic and may also have Conditions on it. That is, Conditions, as they have traditionally been expressed in print documents, may nest or may express pre-existing relationships.
Value	A quantity that describes the results of applying the Conditions to the device identified with the "Product ID " attribute for a specific Parameter. The Characteristic Value is usually expressed as a Number although it may be a complex symbolic Expression. Each Characteristic Value is of a specific type, expressed through the "value.type" attribute, such as minimum, maximum, or typical. For example, in Table 2, "-1.5V" is a maximum value.
Connection Identifiers	Defines the scope of the parameter being measured or reported by identifying the connection(s) for which the characteristic or condition is valid.
Testing Information: (Test Circuit/Procedure)	Procedural steps or a test circuit that can be used to measure the Characteristic.
Other information	Additional text or graphics may be needed to fully describe a characteristic. Such material could take the form of text, equations, graphics, etc.

6.2.2 Values of Characteristics

The information concerning the value(s) of a Characteristic, usually printed as a single row within a table of characteristics data, can be expressed as one of three things:

- A numeric value, which can be expressed as:
  - An actual Number (such as 3, 5.0, or 1.6);
  - An Expression that could be evaluated to a number if additional information were available (such as V_{CC} + 0.5, V_{CC}, or 10%); or
  - A Range (such as ±15);
- A textual object (Source Paragraph), such as a paragraph, equation, or graphic (such as "See AC Characteristics", "TBD", or "No upper limit"); or
- Null.

Each numerical value is associated with an Order of Magnitude and a Unit. Units have values such as seconds and farads. Order of Magnitude is usually a multiplier, such as micro, milli, pico, mega, etc.

Each Characteristic Value, whether it is numeric or textual, may carry with it (through the use of attributes) information concerning its" test level " and "value.type". The "test.level" of a Characteristic Value provides information on how the value was derived. For example, a Characteristic Value could be described as "100% tested", "guaranteed by design", "statistically tested", or "simulated". Each company using the PCIS is free to develop its own list of possible content for the "test.level" attribute. The "value.type" attribute for a Characteristic Value must take as its contents one of the abbreviations in the following list:

max	Maximum	The upper limit at which the provider of the product said the product will operate. (Note: Absolute Maximum is different, that is the risk to the device.)
min	Minimum	The smallest permitted or observed value for a property.

max	Maximum	The upper limit at which the provider of the product said the product will operate. (Note: Absolute Maximum is different, that is the risk to the device.)
typ	Typical	A value suitable for operation, usually determined by some form of statistical averaging, measured at mid-range. Typical values are usually derived from measured values. Typical is NOT synonymous with Nominal.
nom	Nominal	An optimized value suitable for operations of all products in a population. Nominal is the point around which the product is designed. Nominal is the same as Optimal and is not usually measured. Nominal is NOT synonymous with Typical.
ntol	Negative Tolerance	With respect to Nominal.
ptol	Positive Tolerance	With respect to Nominal.
raw	Raw Value	The value as measured.

Condition Values have a structure similar to that of Characteristic Values, with some key differences. They are discussed in Section 6.3.

6.2.3 Aspects of Characteristics and Conditions (for Search and Retrieval)

It is useful to be able to retrieve and display Characteristics and Conditions by the physical properties they are measuring. A searcher might request a single type of Characteristic (such as all the "electrical", all the "switching", or all the "acoustic" characteristics) and should also be able to request combinations of characteristics (all the dynamic (not "static") "electrical" characteristics or all the "static" "electrical" characteristics).

The title of a table in which characteristics are printed or displayed (or the title of a graphic displaying Testing Conditions) cannot provide this information reliably and unambiguously, because titles are inherently variable and may at different times be established by the datasheet author, editor, or a marketing group.

Therefore, to ensure accurate electronic searching, the aspects of Characteristics and Conditions can be explicitly defined through the use of attributes on each such element. These aspects may be viewed as a series of yes/no questions. For example: "Is this a "thermal" characteristic?" "Is this an "optical" condition?" If the logical answer for a particular Characteristic or Condition is neither "yes" nor "no" but "not applicable", the attribute in question takes no value. The aspects are not necessarily mutually exclusive; there may be more than one aspect with a "yes" value on an individual Characteristic or Condition. However, for any given characteristic or condition aspect, "yes" and "no" are mutually exclusive (something cannot both be an optical property and not be an optical property simultaneously).

The list of named characteristic and condition aspects is expected to grow over time. New aspects will be added as experience with electronic components data in computer-sensible form clarifies the selections and groupings of characteristics and conditions that are most useful during retrieval and as the state of the art of electronic component changes. For PCIS, at least through Version 1.4, the following aspects have been defined:

acoustic Does this characteristic describe energy transfer that involves mechanical vibration? (Examples of "acoustic" characteristics include: sound pressure, transducer sensitivity, attenuation and resonance frequency.)

chemical Is this characteristic descriptive of the molecular composition of a material? (Examples of “chemical” characteristics include: alloy compositing and dopant compositing.)

electric Is this characteristic a measurement of the movement or storage of electrical energy? (Examples of “electric” characteristics include: VCC, Input High Voltage, Input LOW Voltage, Input Leakage Test, and Input LOW Current.)

info Is this characteristic or condition information-related? “Information” is defined as intelligence or knowledge capable of being represented in forms suitable for communication, storage or processing.

magnetic Is the presence or absence of a magnetic field relevant to this characteristic or condition? For example, does this characteristic generate or measure a magnetic field? (Examples of “magnetic” characteristics include: permeability, magnetic field intensity, field sensitivity, and shielding.)

mechanical Does this characteristic or condition measure dimension, movement, or the speed of motion of a physical object? (Examples of “mechanical” characteristics include: size, mass, angular or linear displacement, spatial relationships, and information capacity.)

optical Is the presence or absence of energy in the infrared to ultraviolet range of the electromagnetic spectrum relevant to this characteristic or condition? For example, does this characteristic describe a response to or generation of energy in this range? (Examples of “optical” characteristics include: Saturation signal VSAT, Image-area well capacity, Smear, Pixel uniformity, and CCD Spectral Responsivity.)

periodic Is a periodic occurrence relevant to this characteristic or condition? “Periodic” is defined as happening at particular instances and not at others.

radiation Is the presence or absence of ionizing radiation relevant to this characteristic or condition? (An example of an ionizing “radiation” characteristic is X-radiation.)

thermal Is the presence or absence of heat transfer, heat dissipation, or heat generation relevant to this characteristic or condition? (Examples of “thermal” characteristics include: power dissipation, temperature, air flow and sensitivity.)

timing Is the presence or absence of a time interval between two events, or a time interval between two values of the same event, or a rate of change of an event relevant to this characteristic or condition? Is this characteristic or condition measured in time-related units? (Examples of “timing” characteristics include: EPROM programming erasure time, setup time, hold time and clock frequency.)

6.2.4 Additional Information about Characteristics

In addition to the aspects described in Section 6.2.3, the following information may be provided about each Characteristic. Information on the “Product IDs” being described and whether a Characteristic is “Static” are considered so fundamental to the understanding of the Characteristic that they are required.

Product IDs (“pids”) Each Characteristic describes a product or portion of a product. This attribute allows an author to identify the products or portions of products to which the Characteristic applies.

Static versus Dynamic ("static")

Static Characteristics are time invariant; that is, the Characteristic Value does not change for the duration of the measurement, as in DC Characteristics. Non-static (i.e., Dynamic) Characteristics describe connections that vary with time; for example, AC Characteristics. A given Characteristic can be described as Static or Not Static.

Reliability Data ("reliability")

Reliability is defined as the ability of a device to perform a required function under stated conditions for a stated period of time. Thus, "reliability" is a measure of the quality remaining after some time has elapsed during which the device was exposed to particular operating stresses. A given Characteristic may or may not be a measure of reliability.

6.3 Conditions

A Condition usually defines an environmental limit, the state under which a product's behavior is measured or reported. The Conditions under which a Characteristic is measured have a similar internal structure to that of a Characteristics as described in Section 6.2.1, with certain exceptions: while Characteristic Values may take a variety of "value.type"s, all Condition Values take the "value.type" "spec", meaning specified; in addition, each Condition only has a single Condition Value. In other words, characteristics are measured; conditions are set.

6.3.1 Structure of a Condition

Conditions are not always displayed in tabular form. Conditions can be expressed as plots or graphs or described in textual paragraphs. However they may be displayed, all Conditions follow the same conceptual structure:

(Note: bullets have been used below to indicate the hierarchical level)

- Parameter, containing at least one of:
  - Parameter Symbol;
  - Parameter Description; and
  - Parameter Definition;
- Parameter Dictionary ID; followed by
- Zero or More Condition IDs, followed by
- One and only one Condition Value ("value.type" = "spec"), consisting of:
  - Number, Expression, or Range
  - Order of Magnitude
  - Unit of Measure
- Followed by zero or more Connection Identifiers, followed by
- Testing Information such as Test Circuit or Procedures, followed by
- Text objects including paragraphs, graphics, equations, etc.

6.3.2 Aspects of a Condition

Each Condition can be additionally described by the physical properties it is specifying. The list of condition aspects is described in Section 6.2.3.

6.3.3 Additional Information About Conditions

In addition to the aspects named and described in Section 6.2.3, an information creator must provide one additional piece of information about each Condition. Unlike the Characteristic model, the only information crucial enough about a Condition to be required is whether it is “Static”. “Static” is defined more fully in Section 6.2.4.

6.4 Groupings of Characteristics and Conditions

Current documents often gather together collections of characteristics or conditions that have a meaningful relationship to one another. Such relationships vary; some groups are established by function, others by conditions which apply to them. Such groups may then be printed as a table under a single title, such as “A/D Characteristics” or “AC Characteristics”. The ability to find and display groupings of characteristics and conditions, regardless of their print or display association, is necessary for successful electronic retrieval. The attributes of Characteristics and Conditions that describe particular physical aspects are not intended to supplant or replace such traditional groups. The aspect decision is made at the individual characteristic or condition level not the group level. There is no necessity for all the characteristics or conditions in a grouping to be of the same type.

Groups of Characteristics are identified in the Characteristic Group Source. Each Characteristic Group that is created must have a Title describing the group. In addition, the IDs of the Characteristics in the group, the IDs of any Conditions that apply to the group as a whole, and any testing information that applies to the group, can be identified. Condition Groups are defined in the Condition Group Source, and must also have a Title. They may identify the Conditions in the group through their Condition IDs, and any testing information that applies to the whole group.

Certain graphic objects (Audio/Visual Objects), particularly visual representations of Typical Performance, may refer to Characteristic Group(s) or Condition Group(s) associated with them through the “cc.group.ids” or the “cn.group.ids” attributes.

6.5 Connection and Terminus Information

In order to completely describe many products, it is useful to describe the information that is transferred to and from the product and the physical locations at which that information is transferred. PCIS allows information creators to separate information about the logical and physical information transfer points because there are often multiple physical ways in which the same logical information may be transferred. (For example, if a product may be sold in several packages, it is possible that the same logical connection would occur on different physical pins in the different packages.)

Connection and terminus information has traditionally been carried in a printed datasheet or databook but is not necessarily collected in one place or described consistently. PCIS has created a Connection Source in which connection and terminus information can be recorded in a more rigorous fashion if information creators so choose.

A Connection is a description of a single valued phenomenon that conveys information through which a product communicates or interacts with the external world or with another part of itself. The Connection is always associated with a particular “product id”, and is located at a specific point on a specific package, known as a Terminus. The package can also be referred to as a Terminus Group.

Connections are often the first thing conceptualized about a new product; the relationships between connections and terminuses are often finalized later in the process of developing the product. In products that are available in multiple versions, such as multiple packages, the same connection may occur on a different terminus in different physical renditions.

A Connection Group is a named collection of Connections, such as “Inputs”. Not only may Connections appear in more than one logical Connection Group, a Connection Group may contain other Connection Groups. It is desirable that connection information, such as Connection Name, Connection Description, and Connection Mnemonic be stored only once for each Connection, and that the data model the relationships as accurately as possible without requiring redundant storage of any information.

The Connection Source stores all information concerning the Connections, Connection Groups, and Terminus Groups for a given document or set of documents. It is expected that, at the top level, a Terminus Group will represent a package. Within packages, there may be Terminus Groups that are identified by placement or function.

There are likely to be many connections described for a product which may be grouped into Connection Groups for discussion. There may be a different group of terminus information (Terminus Group) for each package or configuration in which a particular product is described. The model for Connections Source, including terminus information, is as follows:

(Note: Bullets have been used in the following example to indicate the hierarchical level.)

Connection Source:

- Connection (a logical phenomenon, such as input, output, power, or ground);
  - Connection Name
  - Connection Mnemonic
  - Connection Description
- Connection Group, (a useful group of connections, as all those of a specific type or function, e.g. Inputs);
  - Connection Group Name
  - Connection Group Description
  - Connection IDs
  - Connection Group IDs
- Terminus Group (a group of physical terminuses, typically a package)
  - Terminus Group Name
  - Terminus Group Description
  - Device Marking
  - Terminus
    - Terminus Name (typically a pin number)
    - Connection IDs

6.6 Support Tools

There is a level of information concerning support tools that is not published in current printed datasheets, at least not in a simple collected fashion. This material includes information on the appropriate usage of the tool, the tool vendor, etc. PCIS provides a mechanism to include this information in a datasheet in the same way that the Characteristic and Condition Sources are included. The logical model for support tool information is as follows:

(Note: Bullets have been used in the following example to indicate the hierarchical level.)

Tool Source

- Tool
  - Tool Information
    - Tool Name
    - Tool Type
    - File Format
    - Publication Date
    - Revision History
  - Usage Information
  - Vendor Information
  - Information About the Intended Application of the Tool
  - General (or Generic) Divisions (including titles, paragraphs, lists, etc.).

6.7 Models and Modeling Information

Modeling information (such as SPICE or VHDL models) can be interchanged as part of PCIS documents in the model's appropriate electronic form. A user receiving a PCIS document containing a model may choose to examine the raw model data, pass the model to another application, or launch an application that supports the model. Therefore, a PCIS-conformant SGML document needs to contain sufficient information about the model for the receiving application to recognize and use the supplied model.

There may be multiple models included with a single document. The PCIS Model element contains some metadata concerning the model, such as the date and author and an indication of whether the model is a package model. The actual model is one or more associated files to which an SGML attribute on the File Format element points. The element hierarchy for the PCIS Model element is defined below.

(Note: Bullets have been used in the following example to indicate the hierarchical level.)

The Model Source contains:

- Model
  - Model Information
    - Model Format
    - File Format (with a pointer to an external file);
    - Publication Date (defined as creation date);
    - Associated Models (see below);
    - Author Information
    - Revision History
  - Usage Information
  - Vendor Information
  - Information About the Intended Application of the Tool
  - General (or Generic) Divisions (including titles, paragraphs, lists, etc.).

Several known modeling tools require an icon for launching. The Pinnacles workshops decided that this "model icon" was actually a small model (used to launch an application) that was associated with another, usually larger, model, that is the application itself. This small launching model was named Associated Model. By extension then, an Associated Model is any object, in addition to the Model itself, that is needed to make use of a computer-sensible model; for example, a schematic symbol as used in schematic capture software packages.

6.8 Product Identification

Datasheets, as defined by the Pinnacles Group, contain information pertaining to one or more products that are manufactured by (or could be manufactured by) a corporation. The words "component", "part", and "device" are frequently used in the industry as synonymous with "product". Since many products are complex combinations of large numbers of other components, the word "product" was selected as the least ambiguous choice. Further, and by extension, a unique process, material, or piece of software could be a product although it is not a component or a device. Most datasheets are straightforward component datasheets that describe the properties and characteristics of a manufactured product directly. There are also family datasheets that describe the similarities and differences in one related group of products. Process datasheets describe different products that may be manufactured by or otherwise related to a particular process or series of processing steps. In a software datasheet, the software itself may be a product and the software must run on or utilize hardware components.

It must always be possible in electronic component documents to identify a product unambiguously because many pieces of information in a datasheet concern a particular product or group of products. The common phrase for this identification is "part number". Unfortunately "part number" in and of itself is not specific enough to describe the different levels of product identification which are used in current documents. A datasheet is considered to be about a product, but the product ID discussed at the datasheet level (often the Generic Product ID), is rarely, if ever, the number used to purchase the product; a more detailed string, the Specific Product ID, that specifies speed, package, etc., must be used instead.

6.8.1 Product Identification Elements

Most PCIS-conformant documents will be about one or more products, and portions of many PCIS-conformant documents will be about one or more products. Part of the Sources contains Product Identification Information. Once Product IDs have been created and stored in the Product ID Source, an information creator may easily associate portions of the document with any one or more of those Product IDs. There is no expectation that all possible products will be identified in the Product ID Source, or that all product IDs will be included in the Sources. The Product ID Source allows information creators to provide full details about particular product IDs as necessary, once, and to use these "Product IDs" throughout the document as needed.

In the PCIS, there are three levels of product identification:

Specific Product ID — Identification for a physical product that can be purchased. (Some companies call this the order number, since this describes all the option codes a customer would need to order a product.)

Product ID Fragment — Identification for a group of products that are related in some manner, such as packaging or temperature characteristics. A Product ID Fragment identifies a useful set of products for some purpose, such as discussion in a characteristics table. These may or may not be equivalent to physical objects that can be purchased.

Generic Product ID — The identification for a specific product or collection of specific products at the level of specificity described in a single document (such as a single datasheet, application note, or databook). This is the ID of the product that is the subject of the document. Such a collection is usually related by functionality, as in a component series or family. (A document may describe a

number of Generic Product IDs and list them separately; each is a generic product if a more specific code can be created for a product that can be purchased.)

The full Specific Product ID is rarely used inside a datasheet; it is more common to use a Product ID Fragment including a wildcard, such as using “the 94xx processor” to mean “the 9419, 9420, and 9430 processors”.

There is no way to tell by looking at a particular Product ID which type of Product ID it is.

In addition, some corporation may also use an Alternate Product ID, which is an alias for the Specific Product ID. An alias may be used for an internal warehouse or inventory number, a customer's name for a product, or a street name for a product.

6.8.2 Using Product IDs in Electronic Component Documents

Many levels of information in a PCIS-conformant document can be associated with a Product ID. The whole document PCIS Datasheet, a Division, a Paragraph, even a Warning, can be associated with a Specific Product ID or with a group of products, a Product ID Fragment. By inference, if a structure is not specifically associated with any individual product ID defined for that document, it is assumed to be about the product ID of the element in which it is nested, working up the element hierarchy. For example, if a Paragraph is about a particular product or group of products, the information creator may indicate that by giving the IDs of the particular product(s) in the “product ids” attribute of the Paragraph. However, if a Paragraph does not have any explicit value for its “product ids” attribute, it is assumed that the information in that Paragraph refers to the same product(s) as the Division in which the Paragraph appears. If that Division does not have any explicit value specified for the “product ids” attribute, but appears inside another Division which DOES, everything inside each of the nested Divisions is assumed to be associated with those products. In very simple documents, the only product IDs may be associated with the entire document, but in many documents, some sections discuss one product ID or set of product options, while other sections of the document discuss others.

A “product ID” associated with an element may contain a value that represents any of the levels of product id; it may be a Generic Product ID, a Product ID Fragment, or a Specific Product ID.

6.8.3 Product ID Segments

A Product ID is frequently a concatenated series of short strings, each of which may contain layers of meaning. For example, a Specific Product ID may be composed of a base code which indicates the product family, attached to a series of codes that describe the type of packaging, the quality level, speed, voltage, etc. These additional (non-base) codes may precede the base, follow the base, or occur between pieces of the base code. The Pinnacles workshops called these strings Product ID Segments. A PCIS Product ID is composed of Product ID Segments; each Product ID, whether a Generic Product ID, a Specific Product ID, or a Product ID Fragment, always contains at least one segment and may contain many. Within current printed documents some companies use a paragraph-level element called an Ordering Information Template, usually displayed as a Table or Audio/Visual Object, to explain the meaning of each Product ID Segment. From information in the Ordering Information Template, a Specific Product ID can be constructed for a particular component based on the desired specifications. (There is no requirement to provide this template.)

Within the Product ID Source, the description of each Product ID Segment may be refined with additional information contained in four different attributes: “segment.base”, “segment.connect”, “segment.type”, and “segment.meaning”:

“segment.base” Stores information as to whether the given segment is the base segment in the Product ID.

“segment.connect” Stores information that the given segment is a connector (such as a hyphen or period), and unlikely to have any encoded significance.

“segment.type” Describes the general type of information this segment describes; each company is free to devise its own “seg.type” attribute values, which are expected to include such things as device package, device type, intended customer, manufacturer ID, process technology, quality level, speed, specification set, version, temperature range, and test level.

“segment.meaning” Describes the function or meaning of the segment in more detail than the “segment.type” attribute can convey.

There is no consistency between companies (or even inside a single company) on what the segments mean or how they are attached to the base. The base may or may not be same as the Generic Product ID.

6.8.4 Product Identification in the Title of a Document

There are two models of document titles. Some documents simply contain Titles, while others (often including datasheets) have composite titles which incorporate the number and description of the product. A composite title is called a Document Title <doc.title>, and is composed of a Product Name Title <prod.name.title> (which often contains the part number or numbers as a Product ID Reflection) and a Descriptive Title <title.descriptive> (which contains a description of the product or products). An Alternate Descriptive Title <alt.descriptive.title> is often also associated with these documents. The Alternate Descriptive Title may be shorter than the Descriptive Title, and is used in running heads and listings such as tables of contents.

7 Noteworthy Structures

7.1 Revision History

The ability to include a detailed Revision History is one of the strengths of electronic documents. It is very useful for information users to be able to find out quickly what part of a document changed and why. The PCIS Revision History element includes:

- Date of change;
- Identifier for the change, change level, or version;
- Reason for change;
- Type of change (the product changed, or the document changed without reflecting a change to the product);
- Whether or not the change prompted re-publication (that is, whether the change was intended to be released at a later time when the document was re-issued, or the document was re-released because this change was made); and
- Indication of what part of the document changed (the IDs of the elements involved).

Thus, within the PCIS documents, all version, revision, and change information is gathered into a Revision History element. There are separate Revision Histories for the document and the product(s).

This is one of two approaches the original Pinnacles Group considered. The workshops rejected change indicators (such as change dates or revision levels) scattered throughout the document on each changed element. If the revision dates or change levels are distributed throughout the document, the authoring system will not keep IDs stable across versions and the entire document will need to be scanned to find everything that changed. Such an approach would also add another type of information (probably in the form of attributes) to virtually every element.

If IDs are stable across versions as the PCIS intends, the Revision History can include the IDs of all structures that were affected by a particular change or change set. Revision History information is centralized in the metadata section of the document and points to changes throughout the document.

7.2 Trademarks

Elements relating to trademarks and registered trademarks are of particular interest to the creators and users of electronic component information. Therefore, the PCIS provides several levels of trademark identification. At the lowest level, the trademark symbol (™) and registered trademark symbol (®) may be included within PCIS documents as the entities "`&trade;`" and "`&reg;`" respectively. (Note: the copyright symbol (©) is also available as the entity "`&copy;`".) These entities may be used anywhere within the text of a document.

The PCIS DTDs also provide for:

Trademark Statement — A sentence or short phrase that describes a current trademark or registered trademark. This sentence or group of sentences usually provides both the name of the trademark and the enterprise (company) which owns that trademark, for example "RDD is a trademark of the ATLIS Consulting Group."

Trademarked — The name of a product, process, service, etc. that is the trademark or registered trademark of a particular enterprise. Trademarked is used like a highlight or emphasis element, to surround a single word or short phrase that is the name of the trademarked object within text.

Trademarks List The highest level of trademark information, containing a collection of Trademark Statements. The Trademarks List usually appears in the introductory section of a document and contains the current trademarks and registered trademarks of the enterprise that produced the document. This may be a complete list, not just a list of the trademarks cited in the document. The list may also include other companies' trademarks if they appear within the document.

7.3 Patent-Related Information

Patent-related information (identifying the owners of particular trademarks and the registration sources of those trademarks) is also a significant portion of information contained in electronic component documents.

PCIS accommodates members of the electronic component industry who wish to encode this type of information with a section that can occur throughout a document, Patents List, and which itself has subsidiary information for encoding each Patent Statement, the Patent Owner, and any Patent Authority.

7.4 Index Structures

Databooks are typically a compilation of documents including: Datasheets, Application Notes, miscellaneous Documents such as descriptions of families, introductions to sections of the databook, and a variety of indexes and selection guides intended to assist users in selecting the appropriate part and finding the appropriate portion of the databook. These indexes may include “Cross-references by Manufacturers Part Number”, “Cross-references by Function”, and “Cross-references by Package”. Such indexes vary greatly, and it is impossible to predict the form they will take. Instead of trying to model the content of these structures, the PCIS defines them as miscellaneous documents. Probably, most of them will be tagged in the form of structural tables with one or more columns of each table containing cross-references to other parts of the same document or to other documents. In print documents these cross-references take the form of product names or numbers, or document titles. In electronic databooks they can also be hypertext links; the users identifying one can automatically open the correct section of the mentioned document.

To assist in the creation of these index documents, and to allow the transfer of indexing information within the documents, an Index Entry element has been provided. The Index Entry may appear almost anywhere text may occur, and is to be used to identify portions of a document that should appear in any of various indices. Each Index Entry may identify which index the entry belongs in, allowing the creation of multiple indexes automatically from the same text. The Index Entry then contains between one and four levels of index terms. These can be used to create a one to four level index.

7.5 Other Information Types (Structural Elements)

It would be optimistic to think that the document analysis process that created the PCIS was complete or perfect, and thus it would be unrealistic to assume that the PCIS is perfect or complete. Further, even if the process had captured all possible and useful structures, and had they all been included in the PCIS, many legitimate requirements continue to arise for either information of a type that did not exist when the PCIS analysis was originally done or for which models have not yet accomplished in any of the update and revision cycles.

To accommodate information types that were not identified in the analysis (as well as those types that were not included in the PCIS because workshop participants couldn't define them, couldn't agree on what they meant, or agreed that they were relatively unimportant and those types that didn’t exist at the time of the original analysis) a mechanism has been provided to include “heads and text” at any level in the hierarchical structure of PCIS-conformant documents. This mechanism is a set of structural elements that can be used to describe information on any subject matter. At the highest level, instead of named divisions such as Architectural/Functional Description or Soldering and Mounting, the user can

use the “Division” element (called Product Technical Division). When a Product Technical Division element is used, the information creator indicates the Information Class of the division through the “div.class” attribute, and indicates the specific subject matter in the Title of the Division. The division can then include Paragraphs, Tables, Lists, and other Divisions (both specific named divisions or unnamed Divisions), on any subject matter.

Content elements, such as the named division Absolute Maximum, are identified based on the kind of information they contain. Structural elements, such as Paragraphs or Titles, are identified based on their structure within the document, not on their content. For example, a numbered list can be about any subject matter, and numbered lists are allowed in many contexts in PCIS documents. The Ordered List, and the List Items that it is made of, are structural elements.

7.6 Graphics: Audio/Visual Objects

Not all of the information in a databook or datasheet is provided in the text. Existing print documents contain a high percentage of diagrams, schematics, charts, graphs, plots, etc. In the future, electronic databooks may well also include video, audio clips, animation, etc. Therefore, the Pinnacles DTDs refer to a graphic (or other non-SGML inclusion) as an Audio/Visual Object.

7.6.1 Audio/Visual Object Notation

Ideally, the Pinnacles Component Information Standard would name a single graphics format for each category of Audio/Visual Object (such as MPEG for video) and all objects of that type would be interchanged in that format. In practice, the original Pinnacles Group felt that they lacked the information necessary to make well-informed decisions on a robust set of graphics formats in time for the original release of the standard. The rising popularity of the World Wide Web, the advent of numerous new graphics packages and competing standards, and the increasing complexity of graphic objects that are part of electronic component information, make that task no easier in PCIS Version 1.4 than it was for PCIS Version 1.0.

PCIS defines a number of graphics formats that are suitable for interchange. That list includes:

CGM-Binary	(Computer Graphics Metafile). Often used for the presentation of complex graphics;
EPSI	(Encapsulated PostScript Interchange Format) The Adobe standard EPSI, bit-map data format;
GIF	(Graphic Interchange Format), often used for the encoding of graphics on the World Wide Web;
TeX	TeX, for mathematical notation; and
TIFF Uncompressed	When TIFF color images are needed.

It is expected that exploration of additional formats for Audio/Visual Objects, and the issues involved in their interchange will be an important part of the development of a future release.

7.6.2 Elements Associated with Audio/Visual Objects

Audio/Visual Objects may have Titles, Captions, and Descriptions. In addition, some Audio/Visual Objects are associated with Characteristic Groups, and may contain information on the Characteristic and Condition Group(s) depicted, and the x and y scales of the graphics. (For a detailed description of the mechanism for describing this information, look in theTag Library under Audio/Visual Object Characteristics Information <avo.cc.info>.

7.6.3 Audio/Visual Object Attributes

There are a small number of attributes associated with Audio/Visual Objects in the PCIS DTDs, used to identify the way the Audio/Visual Object is encoded (the notation), the unique ID of the object, and whether it should be placed in-line or as a callout. In addition, when a CGM graphic file is transmitted as part of a PCIS-encoded interchange document, an additional attribute, "profile.ptr" is used to name the profile with which that graphic file was created.

These attributes do not, in and of themselves, contain sufficient information to allow the creators of authoring or display-set DTDs to adopt them without modification or the addition of new attributes.

Depending on the requirements for a particular DTD set, information creators may wish to add attributes to further control the presentation of an Audio/Visual Object on the screen on on paper, to specify size, column placement, leading, or additional other layout details.

DTD creators and maintainers would be well advised to discuss the formatting and layout requirements of any tool or application into which PCIS-conformant SGML should be ported or transformed to determine the additional information elements that will be required to sucessfully handle Audio/Visual Objects in working applications.

7.6.4 Audio/Visual Objects and Entity Declarations

Audio/Visual Objects, as non-SGML files, are referenced in PCIS-conformant documents in a two-step process. An "entity declaration" at the beginning of the document (in the document prefix or header) names each Audio/Visual Object and identifies it either with a SYSTEM identifier (for a paraticular machine or network) or with a PUBLIC identifier (a more abstract identifier). Once the entity declaration for a particular Audio/Visual Object has been created, that entity name is referenced each time that Audio/Visual Object should appear in the document through the "avo.ptr" attribute of the Audio/Visual Object.

8 PCIS Glossary

8.1 SGML Terms, Elements, and Contexts

AAP SGML Application Standards (ANSI/NISO Z39.59-1988; ISO 12083)
An SGML application standard that describes a set of elements and document models for books, journal articles, tables, and math processing. These standards were designed by the Association of American Publishers to facilitate exchange of manuscripts among authors and publishers.

attribute
An attribute is an SGML construct associated with an element that provides additional information about the element. For example, the "Document Status" (Preliminary, Obsolete, etc.) that prints on a datasheet is information known about that datasheet. This "doc.status " attribute is associated with the PCIS Datasheet element. One attribute may be associated with many different elements; one element may have many different attributes associated with it.

autotagger
Software that inserts SGML tags into the appropriate places in a file (such as a word-processing file or flat ASCII file). The process is called autotagging or automatic markup. Some autotaggers also "uncut" files, that is, convert from an SGML-tagged file to a desktop publishing system or word-processing file. Autotagging functions can be built into a scanner, built into an editor or word processor, or performed by a stand-alone autotagging product.

CALS (Continuous Acquisition and Life-Cycle Support) Formerly Computer-Aided Acquisition and Logistics Support
A United States Department of Defense initiative that was among the first to make extensive use of SGML. The relevance to PCIS is that the CALS MIL-M-28001B Table model was the most widely used and supported structural table model through the mid-1990s. The PCIS initially adopted the CALS model as the way to tag rows and columns of information. SGML Open has revised this model for interchange and PCIS, as of Version 1.3, includes the SGML Open Exchange Table Model as a replacement for the original CALS table model.

content model
The SGML definition of the elements that may be contained inside another element and the relationships among those elements. Content models may indicate sequence of elements, rules for repeatability, and whether or not elements may be omitted.

data element
Also called "element". An identifiable part of a document or information collection; for example: a chapter, title, part number, or parameter symbol.

document
A coherent, ordered collection of information (like a book or a magazine article or a database).

DSSSL
Document Style Semantics and Specification Language — a companion ISO standard to SGML. DSSSL will provide a means to specify document processing, such as formatting for print and display media and data management functions, similarly to the way SGML provides a means to specify document and information structure.

DTD
Document Type Definition: an SGML model of the structure or content of a document (or group of similar documents). This model identifies the tags that will be used in documents and the permissible relationships among the information identified by the tags.

element
Also called "data element". A part (piece) of a document or information base. There are many kinds of elements; some of the possibilities include:

- Structural elements, such as chapters, lists, or paragraphs;

- Content elements, such as phone numbers, part numbers, or personal names;
- Informational elements, such as an index entries; and
- Linking or connection elements, such as cross-references or hypertext links.

entity An SGML construct that contains information used for character replacement. General entities can be used in SGML-tagged documents to bring in boilerplate text or non-ASCII special characters such as omega and arrow. SGML parameter entities are used within DTDs to create useful groupings of elements, attributes, or other information. External entities are references to external objects such as data files, which use PUBLIC identifiers or full SYSTEM-specific path names.

formatter Software that interprets an SGML tagged file to create formatted output such as composed pages or screen displays. Formatters may handle SGML tags directly or translate the tags into specific codes for another product (such as a desktop publishing system).

granularity The level of specificity with which parts of a document are identified. For example, a part of a document could be identified as a section, with no further breakdown, or it could be divided into a title and paragraphs, or within the paragraphs each part number could be identified. Part numbers could be subdivided into main and sub-parts, each of which is identified. How finely to divide the data is a granularity decision.

HyTime Hypermedia/Time-based Structuring Language — An ISO standard that describes a language for the structured representation of hypertext multimedia information. HyTime is an application of SGML that uses DTDs written in SGML syntax to describe constructs used to organize time and spatial information.

ISO 8879 The International Standards Organization (ISO) standard that defines the SGML rules. (See glossary entry for *SGML*).

ISO 12083 (The AAP standard) An ISO version of the American Association of Publishers standards, developed in 1993, and reflecting changes in the state of the art since the original AAP Standards were developed in 1986. This SGML application standard describes a set of elements and document models for books and journal articles, and includes techniques for specifying Braille, Large Print, and Voice Synthesis versions of documents.

metadata Data concerning data. That is, information that concerns or describes other information. For example, a book may be data and the bibliographic information about the book would be metadata.

MIL-M-28001 (formerly MIL-M-28001A, then MIL-M-28001B, then MIL-M-28001C) The CALS SGML application standard that describes the elements and document models for technical manuals written according to MIL-M-38784 (MIL-M-38784B). MIL-M-28001 includes a standard DTD and variations, a model DTD, a standard Output Specification DTD (OS DTD), and a model Format Output Specification Instance (FOSI).

output specification Description of the style and format associated with each element and attribute of a document type for one presentation of a document. An SGML output specification usually includes the formatting consequences (if any) of every tag in the DTD in the contexts in which the element may occur.

parameter entity An SGML construct used to simplify a DTD or to allow DTD modification through substitution. An entity (SGML object) is named and then defined as being equivalent to a string of characters. Whenever the entity name is used,

an SGML parser substitutes the (usually much longer) character string. One parameter entity can be referred to many times. Therefore, if something is defined as a parameter entity and that entity is then used five times, a change to the entity will change all five places it is used.

parser See glossary entry for *SGML parser*.

RDD Rapid DTD Development. A technique, proprietary to the ATLIS Consulting Group, for performing SGML document analysis that uses highly structured, one-time, one-place, facilitated decision-making workshops to enable users and creators of information to analyze their documents. The information-gathering phase of the PCIS was conducted as a series of RDD workshops.
(Note: may be used in a sentence; as RDD process, RDD method, or RDD workshop.)

"real SGML" A real SGML application must use a DTD and an SGML parser, not just tags in text or pointy brackets "< >".

SDATA entity Specific Character Data Entity. An entity where the replacement is defined to be a system-specific reference. SDATA entities are used to define special characters (such as omega, acute e, or smiley-face) that must be mapped to actual character representations in a particular system.

SGML Standard Generalized Markup Language, an International Standard for the Markup of Text. ISO 8879. SGML is a language for describing the structure of documents (or information) and for describing a tagging scheme to delineate that structure within text.

SGML application One concrete instance (that includes hardware, software, and procedures) by which a specific user community uses the features of SGML to process information (for example, to produce a formatted document or access a collection of data).

SGML application standard
A specific set of tags, element definitions, and element models usually intended for use by a wide variety of organizations with similar documents. SGML application standards specify subsets of SGML and are fully compliant with the SGML standard, ISO 8879.

SGML parser An SGML software product that verifies DTDs against the rules of SGML and also checks a tagged document against the rules of its DTD. May be a stand-alone package or may be built into another SGML product such as an SGML editor or conversion program.

table Information, logically arranged in more than one dimension, that is displayed in rows and columns. Structural tagging for a table describes the table in terms of these rows and columns, regardless of the content of the table cells.

tag A string inserted in a document to identify the information contained in that part of the document. Tags delimit (mark, surround, or enclose) an element in an SGML-encoded file. An SGML-encoded document file will contain both tags and text. Each tag is formed from three pieces: a start delimiter (usually "<" for an element start tag and "</" for an element end tag), the short name of an element, and an end delimiter (usually ">"), like this:

```
<tag>This is a tagged element.</tag> <paragraph>...lots
of sentences...</paragraph>
```

TEI Text Encoding Initiative — An SGML application standard that describes a set of elements and document models for scholarly research and analysis.

transformers Computer programs used to convert documents coded with one encoding scheme into documents encoded with another. In the broadest sense, this includes autotaggers that translate from something else into SGML or out of SGML into something else. As the term is frequently used in the electronic publishing industry, transformers are used to translate documents tagged according to one DTD into SGML documents tagged according to another DTD.

8.2 Electronic Component Terms and Contexts

Characteristic An inherent or measurable property of a product expressed as a value or values under stated or implied conditions. In current printed datasheets, a characteristic is usually (though not always) expressed as a single row in a table of similar characteristics (such as DC Characteristics or AC Characteristics). In such a table, a characteristic is a particular parameter, measured or specified under a known set of conditions, yielding the results (values) stated.

Characteristic Source An element in the psources.ent module which contains elements designed to provide support for parametric searching of characteristic information and to describe those elements which can create an an arbitrary display of the parametric information. The Characteristic Source is the version of the characteristics data that will be used for search and retrieval and for loading databases. For each characteristic, the Characteristic Source will contain the full hierarchical structure for characteristics (parameter, conditions, values, etc.) One or more Reflection(s) may be used to print or display the characteristics data. (See also glossary entry for *reflection*).

chip set A group of electronic components (chips) which work together as a group (set). They are usually described in a single document (datasheet or databook), although they may also be described separately in individual documents (datasheets).

data element Approximately synonymous with the PCIS term, Characteristic. As used by some electronic component companies, a data element is a group of information including a parameter letter symbol, a parameter description, the value sets (such as min, typ, and max), the unit, and all referenced conditions. (For example, in the current printed datasheets, this would be one row in an electrical characteristics table.)

databook A collection of datasheets and the related supporting information.

datasheet A product-specific document that contains the essential information about a product.
(Note: “Product” in this sense may mean a range of related products, for example many different voltages of the “same” part, if these are all described in a single document.)

division Divisions are hierarchical structures within text that usually begin with a title and which may nest. General divisions will contain any content desired. Some specifically named divisions are predefined to contain specific types of content to aid in retrieval. Examples include Architectural/Functional Description and Product Mechanical Information. Each division will be pre-assigned to one of the information classes described in Section 3.3.

parameter An essential electrical or physical characteristic of an electronic component (such as a chip or integrated circuit). Examples of parameters would include: Ambient Operating Temperature, Output Current, Collector Voltage, etc.

reflection

One or more Reflections(s) will be used to print or display data that is stored, searched, and accessed through the hierarchical structure of the various source elements. (See glossary entry for *Characteristic Source* for an example.) Reflection(s) may be in the form of graphics, paragraphs, or tables. Each Reflection contains references (pointers) to information stored in one of the Sources instead of actual data. This way, if the Sources are updated, the text or tabular Reflection will be automatically changed, too.

9 Element Naming Conventions

Element names for the Pinnacles Component Information Standard were chosen by the workshop participants. Names were intended to describe, as clearly as possible, the content of the element; that is, an attempt was made to make names self-defining but not to be trapped by the fallacy that "Everybody knows what that means." The common industry word for an object (such as component "part number") was often NOT chosen, because analysis revealed that the word was used in different senses by different companies or in more than one way within the electronic component industry. It was considered better to invent a new name for which everyone would need to read a definition than to use a standard name for which everyone "thinks" they know the meaning.

PCIS tag names and attribute names were constructed by the Pinnacles Group by taking the critical words from the element name or attribute description, abbreviating them if they were too long, and inserting a period between each word. Thus, for example, the element Architectural/Functional Description was given the tag <arch.func.desc>. The word abbreviations that were used to construct the names are listed below.

(Note: For PCIS element names, attributes names, and attribute values, upper and lower case letters are equivalent. Thus "table" and "TABLE" and "Table" are the same.)

The following abbreviations have been used in PCIS tag and abbreviation names:

abs	absolute
acknowledge	acknowledgments
admin	administrative
alt	alternative
appnote	application note
arch	architectural
avo	audio/visual object
cc	characteristic
char	characterization
clink	contextual link
cn	condition
conn	connection
def	definition
desc	description
desig	designator
dict	dictionary
div	division
doc	document
fig	figure
func	functional
gen	general
id	identifier
ilink	independent link
info	information

lang	language
li	list item
mag	magnitude
max	maximum
p	paragraph
parm	parameter
pcis	Pinnacles Component Information Standard
pid	product ID
prod	product
ref	reference
seg	segment
stds	standard(s)
sub	subscript
super	superscript
tech	technical
toc	table of contents
vers	version
xref	cross-reference

The following abbreviations have been used in the SGML Open Exchange Table Module element and abbreviation names:

align	alignment
char	character
charoff	character offset
col	column
cols	columns
mdl	model
nameend	name end
namest	name start
num	number
pgwide	page wide placement
sep	separator
spec	specification
tbody	table body
tgroup	table group
thead	table head
valign	vertical alignment

10 Initial Pinnacles Guidelines

Before beginning the development of the PCIS, the Pinnacles Group developed the following Guidelines:

- Element granularity decisions should be driven by the logical content of the data, not by the format or structure of the information.
- Data elements will be described to the extent necessary to make them retrievable by their uniquely named content rather than by location within a structure.
- The PCIS should create a data format that can be loaded from current publishing tools, in so far as content and structure in existing data can be inferred from or are encoded in current data format.
- The data structure should be compatible with current on-line delivery tools.
- The Pinnacles Group anticipates a Phase 2 of this initiative. During the workshops many good ideas will surface that are related to the project but outside the scope of the current effort. These will be recorded as candidate goals for phase 2.

  The architecture developed in this phase should be, in as much as possible, extensible, so that it can be used as a basis for phase 2 development.
- It is the intent of the Pinnacles Group to produce DTDs that can be used as a base for authoring-set DTDs that will be created and used by each company. The PCIS will identify the information to be interchanged among companies, and recognizes that companies are likely to have additional information on these components and documents for internal use. The companies should be able to add descriptions of this additional information to the DTDs produced in this effort, rather than needing to develop completely new DTDs for use internally.
- The data structures defined should include a generic way to include notations; for modeling for instance (for example: IGES, VHDL, SPICE), which is end-user extensible. It should not be necessary to modify the standard to include additional notations in documents.
- The documentation of the standard should include not only what the workshops decided, but also the reasons behind the decisions, and why another way was not chosen. This applies both to the document architecture and to the details of the DTDs.
- The workshops should identify applicable standards for consideration during the development of or for incorporation by reference into the standard.
- The Pinnacles Group thinks that is likely that the PCIS will make use of SGML architectural forms in development of the standard, and possible that the standard will use the specific architectural forms identified in the HyTime standard.
- The PCIS will be designed to be maintainable. In an industry that is changing as fast as the electronic component industry, it is a basic design premise that the requirements will constantly change. The mechanism for maintaining the standard is specifically excluded from this project, however. Eventually, the Pinnacles Group should come up with a mechanism for maintaining the standard. It is likely that the Pinnacles Group will turn the draft standard over to a recognized standards body, who will promulgate and maintain it.
- The Pinnacles workshops are working from technology snapshots therefore analyzing “current documents” only. That is, the workshops will analyze those documents for each Host company that have been published as of the date of the workshop at that host. If there are additional requirements for structures that a company identifies after their workshop, these needs will be addressed after the draft standard is completed, as a maintenance activity.
- This effort will not create a graphics standard; it will not select a graphics standard, and it will not choose or limit the graphics standards the various companies may choose to use. It is possible that the PCIS may name one or more preferred graphics formats.

- The PCIS includes a “safety net” for information that is unknown at this time or that may be missed during the workshop analyses. The Pinnacles Group assumes that there is more material in the documents than we know about, and direct that the PCIS should provide ways to include this information in the SGML files. It is likely that this information will be accommodated through the inclusion of generic structural elements.

11 PCIS Formal Public Identifiers

The PCIS uses SGML formal public identifiers, as defined in ISO 8879, to identify all PCIS-related documents. A formal public identifier provides a unique identifier of any text that is to be made public, for example, a PCIS document that two organizations will exchange. This section describes how these identifiers are constructed. It is intended for the benefit of PCIS implementors and contains technical terminology.

11.1 Introduction

A formal public identifier is a unique application-independent and system-independent name that identifies the owner of a document (in this case the enterprise) and the specific document. All the DTDs and DTD fragments within the PCIS include formal public identifiers. As datasheets, databooks, and other documents are created using the PCIS, each document instance will also be given a formal public identifier.

According to Dr. Charles Goldfarb's *The SGML Handbook* and ISO 8879, a formal public identifier is:

- An SGML minimum literal;
- Divided into meaningful pieces through the use of a double slash; and
- Comprised of two parts, an owner identifier and a text identifier, separated by the double slash like this:

  `“owner identifier//text identifier”.`

An SGML minimum literal:

- Is surrounded by either single or double quotes;
- Contains only alphabetic characters (upper and lower case), digits, and the SGML special minimum data characters ("?", "=", ":", "/", ".", "-", ",", "+", "(", ")", "'", record start, record end, and space); and
- Has all sequences of interior blanks reduced to a single blank and no leading or trailing blanks.

11.2 Owner Identifier: PCIS

The owner identifier identifies the enterprise that is responsible for the document (not the manufacturer of the product, if the manufacturer and the enterprise are different). For PCIS, this identifier is an Registered Owner Identifier as defined in ISO 8879 and is composed of:

- The characters “+//” followed by
- Minimum data, in this case a registered owner name.

The registered owner name for documents (DTDs and DTD entities) that are part of the PCIS Version 1.4 consists of:

```
+//ISO/IEC 9070/RA::A00007::US::ECIX
```

11.3 Owner Identifier: PCIS-Conformant Documents

The owner identifier for PCIS-conformant documents consists of:

- The prefix “PCIS”;
- Two colons (“::”); and
- The owner name component, which is the ISBN prefix of the enterprise (company) involved.

11.3.1 ISBN Prefix

The ISBN prefix is constructed according to the rules of “ISO/IEC 9070: Information Technology - SGML support facilities - Registration procedures for public text owner identifiers” and consists of:

- The letters “ISBN”;
- A space;
- The group identifier of an ISBN number; and
- The publisher identifier of an ISBN number (if applicable), preceded by a hyphen.

11.4 Text Identifier

The text identifier, which follows the registered owner name for DTDs and DTD modules (see Section 11.2) and follows the owner identifier and ISBN prefix for documents created in conformance with the PCIS DTDs (see Section 11.3), names the document and provides some distinguishing information concerning it. The text identifier consists of:

- A public text class (taken from the list of public text classes explained in Section 11.5);
- A space;
- A document identifier that should be designed to uniquely identify the particular document or entity among all other documents or entities that have the same owner identifier. The content of this document identifier or object name has no particular restrictions imposed by either ISO 8879 or ISO 9070, except that it should not exceed 100 characters. The specifications used for PCIS standards-related documents and PCIS-compliant SGML documents are described in Section 11.6, but companies working with the ECIX standards should recognzie that they are free to use or ignore any of the guidelines contained in Section 11.6 as long as their documents or modules are easily distinguished from one another;
- Two slashes; and
- The principal language of the document, expressed as a 2-character code from ISO 639 (English is “EN”).

11.5 Public Text Classes

The following subset of the ISO 8879 public text classes may be used in constructing formal public identifiers for PCIS-conformant documents:

CHARSET	A character set
DOCUMENT	A complete SGML document, including an SGML Declaration, a DTD, and a document instance
DTD	Document Type Declaration subset
ELEMENTS	A grouping of element, attribute, notation, or parameter entity declarations
ENTITIES	An entity set
NONSGML	A non-SGML data entity
NOTATION	A definition for a notation
TEXT	An SGML document or a portion of an SGML document

11.6 Conventions Used in PCIS-Related Object Name Creation

A document identifier or object name should be designed to uniquely identify the particular document or entity among all other documents or entities that have the same owner identifier. The content of this document identifier or object name has no particular restrictions imposed by either ISO 8879 or ISO 9970, except that it should not exceed 100 characters. The specification used for PCIS standards-related documents and PCIS-compliant SGML documents through PCIS Version 1.4 consists of:

- For complete documents, the literature order number of the document (such as M21T006); or
- For DTDs and DTD modules, the base filename of the module (such as PINNBASE.ENT or PSHEET.DTD);
- A space;
- A version number (For document instances, each company is free to determine its own format for this number. For PCIS standards documents, the current style for creating a version number is described in Section 11.6.1);
- A space; and
- A title or name for the document or a short version of the title. While this text can be enclosed in parentheses (and often has been in PCIS Versions 1.0 through 1.3), there is no standards-related reason for doing so.

11.6.1 Version Number Conventions Used for Standards-Related Documents Associated with PCIS

For PCIS standards documents, the version number is created by concatenating:

- The date in ISO standard format (YYYYMMDD),
- A space;
- The word “Draft”;
- A space; and
- A two-part version number, where the first digit is the major release, the second digit is the revision, and the two numbers are separated by a period (e.g., 1.0, 1.1., 1.2, etc.).

11.7 Sample Full Formal Public Identifier for a PCIS-Conformant Document

Putting together the owner identifier and the text identifier, the full formal public identifier for a PCIS-compliant document instance might be:

```
"-//PCIS::ISBN gggg-pppp//TEXT T145-12A C2.0(74F821 10-BitD-Type
Flip-Flop)//EN"
```

11.8 Formal Public Identifiers for PCIS DTDs and DTD Modules

The formal public identifiers for the Pinnacles Component Information Standard DTDs and DTD modules, as of Version 1.4, are as follows:

11.8.1 DTDs

For the Datasheet DTD:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PSHEET.DTD 19971231
Draft 1.4 PCIS Datasheet DTD//EN"
```

For the Application Note DTD:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PAPPNOTE.DTD 19971231
Draft 1.4 PCIS Application Note DTD//EN"
```

For the Databook DTD:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PDBOOK.DTD 19971231
Draft 1.4 PCIS Databook DTD//EN"
```

For the Pinnacles Document or Document Fragment DTD:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PCISDOC.DTD 19971231
Draft 1.4 PCIS Document DTD//EN"
```

11.8.2 DTD Modules

For the Pinnacles Base Module:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PINNBASE.ENT 19971231
Draft 1.4 PCIS Basic Elements and Entities//EN"
```

For the Pinnacles Administrative Information module:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PADMIN.ENT 19971231
Draft 1.4 PCIS Administrative Information//EN"
```

For the Pinnacles Sources module (for characteristics, models, etc.):

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//DTD PSOURCES.ENT 19971231
Draft 1.4 PCIS CCs, CNs, Conns, Models, Tools, Product IDs//EN"
```

For the Pinnacles Special Characters and Character Sets module:

```
"+// ISO/IEC 9070/RA::A00007::US::ECIX::PCIS//ENTITIES PCHARSET.ENT
19971231 Draft 1.4 PCIS Special Characters//EN"
```

For the SGML Open Exchange Table module called by the PCIS DTDs:

```
"-//SGML Open//DTD Exchange Table Model 19960430//EN"
```

12 Bibliography

12.1 Other Standards Cited

Guidelines for Electronic Text Encoding and Interchange (TEI), ed. C. M. Sperberg-McQueen and Lou Burnard, copyright 1990, 1992, 1993, 1994, ACH, ACL, ALLC, Chicago and Oxford

InfoMaster Architecture (IBMIDDOC DTD)