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

NISTIR 5224

National PDES Testbed
Report Series

An SGML DTD for the STEP Integrated Resource Parts

July, 1993

Scott Bodarky
Scott W. Paisley

NATIONAL PDES TESTBED™

July 08,1993

NIST

NISTIR 5224

National PDES Testbed
Report Series

Sponsored by:

U.S. Department of Defense
CALS Evaluation and
Integration Office
The Pentagon
Washington, DC 20301-8000

U.S. Department of Commerce
Ronald H. Brown,
Secretary of Commerce

National Institute of
Standards and Technology
Arati Prabhakar,Director

NATIONAL PDES TESTBED™

An SGML DTD for the STEP Integrated Resource Parts

July, 1993

Scott Bodarky
Scott W. Paisley

July 08, 1993

DEPARTMENT OF COMMERCE • UNITED STATES OF AMERICA •

Table of Contents

An SGML DTD for the STEP Integrated Resource Parts

Scott Bodarky

Scott W. Paisley

1 Abstract

The Standard for the Exchange of Product Model Data (STEP) is emerging under the International Organization for Standardization (ISO). One major component of this standard is a series of documents called Integrated Resources. The purpose of this document is to present a Document Type Definition (DTD) for the Integrated Resources, in the Standard Generalized Markup Language (SGML). The DTD is a template that encapsulates and formalizes the structure of the Integrated Resources, and permits their semantic content to be converted into SGML. This serves to encode the structure of the Integrated Resources in a formalized, software-based system, thereby eliminating many vagaries of human interaction that are inherent in work done by large committees across international boundaries. In addition, documents written to the structure of a particular DTD can be validated as structurally correct.

2 Introduction

The Standard for the Exchange of Product Model Data (STEP)[1] is emerging under the auspices of the International Organization for Standardization (ISO). It pertains to the interchange of manufacturing information between computer systems. The United States' activity in support of STEP is called Product Data Exchange using STEP (PDES).

Integrated Resources (IRs), such as Part41 - "Fundamentals of Product Description and Support"[4], are the basic building blocks of STEP. They provide information models for generic constructs which are useful in a wide variety of applications. Application Protocols (APs), such as Part 202 - "Application protocol: Associative draughting"[9], are those Parts[2] of STEP which combine the constructs modelled in the IRs, select implementation mechanisms, and specify what product data is to be exchanged and what the meaning of that data is in a particular industrial context.

These STEP Parts are created by projects within working groups under the auspices of an ISO subcommittee. The process is moderated by a Part Editor. The emerging document is structured according to the rules specified in Supplementary Directives[13]. It is then reviewed by others in the STEP community, who verify that it does, in fact, adhere to Supplementary Directives. This process is rife with opportunity for ambiguity and error.

The National PDES Testbed (NPT) at the National Institute of Standards and Technology (NIST) is presently engaged in the creation of an Application Protocol Development Environment (APDE)[11]. The NPT project is principally funded from the Department of Defense Computer-aided Acquisition and Logistic Support (CALS). The APDE will provide Part Editors with a formal, software-based system for creating properly structured documents. Application Protocols will be produced more efficiently, resulting an a standard that is clearer and more internally consistent. In addition, the APDE will provide an environment which allows APs to be developed more quickly.

The core of the APDE is the Application Protocol Information Base (APIB), which will store the fundaments of STEP in a database, from which they will be available to other software mechanisms, and, by extension, the AP developers. The initial set of STEP Parts to be stored will be the Integrated Resources, because they are the foundations from which AP's are created.

[1] The Standard for the Exchange of Product Model Data (STEP) is a project of the International Organization for Standardization (ISO) Technical Committee on Industrial Automation Systems (TC184) Subcommittee on Industrial Data and Global Manufacturing Programming Languages (SC4). For an overview of the standard refer to *Part 1: Overview and Fundamental Principles*[2].

[2] STEP will be released as a collection of specifications; each individual specification is known as a "Part" of STEP.

Before the parts can be stored in a database they will be converted into Standard Generalized Markup Language (SGML)[1] documents. SGML is an international standard which allows the formal specification of a documents structure. When a document is carefully marked up, structural tags can be associated with content. One of the benefits of SGML is that it will allow software systems to recognize content in a document. Once the STEP parts are converted into SGML, these documents will contain tags which will be used to map the contents of the document into a database.

This document is intended for those involved with the development of STEP. In particular, this document addresses problems associated with writing STEP documentation, and enumerates the first steps the APDE is taking to solve these problems.

Stylistic conventions used in this document are as follows:

Times-Roman font for text.
Bold Times-Roman for SGML items inside of paragraph text
`Courier font` for SGML code.

3 SGML Applied to STEP

One of the goals of the APDE is to provide a method for a Part Editor to gain easy access to vital information contained in STEP documents. For example, a Part Editor will need information contained in the STEP Integrated Resources to develop an AP. The Part Editor may only need a small segment of an IR to develop an AP. For the APDE to be able to store the information contained in the Integrated Resources, we must first be able to identify individual pieces of the content (e.g. EXPRESS[3] code) in the documents.

Storing the content of a document in a database requires first differentiating said content from the document's structure. The Standard Generalized Markup Language (SGML) can be used to describe document structure in terms of content, whereas the current STEP documents have their structure described in terms of appearance. Using SGML will allow easy recognition of the content of STEP documents by automated systems. The first step in converting the STEP Integrated Resources into SGML documents is to create a Document Type Definition (DTD) for them. This DTD is a template that encapsulates and formalizes the document structure and permits the semantic content of a STEP Integrated Resource to be represented in an SGML file. The content remains discrete in the SGML file and software systems can easily recognize and process this SGML file. The DTD, presented in appendix A, has been created by analyzing the Supplementary Directives and five of the existing IRs and developing from them a model for a properly structured STEP IR. (Figure 1 depicts the top level structure of an IR, and was derived from the model.)

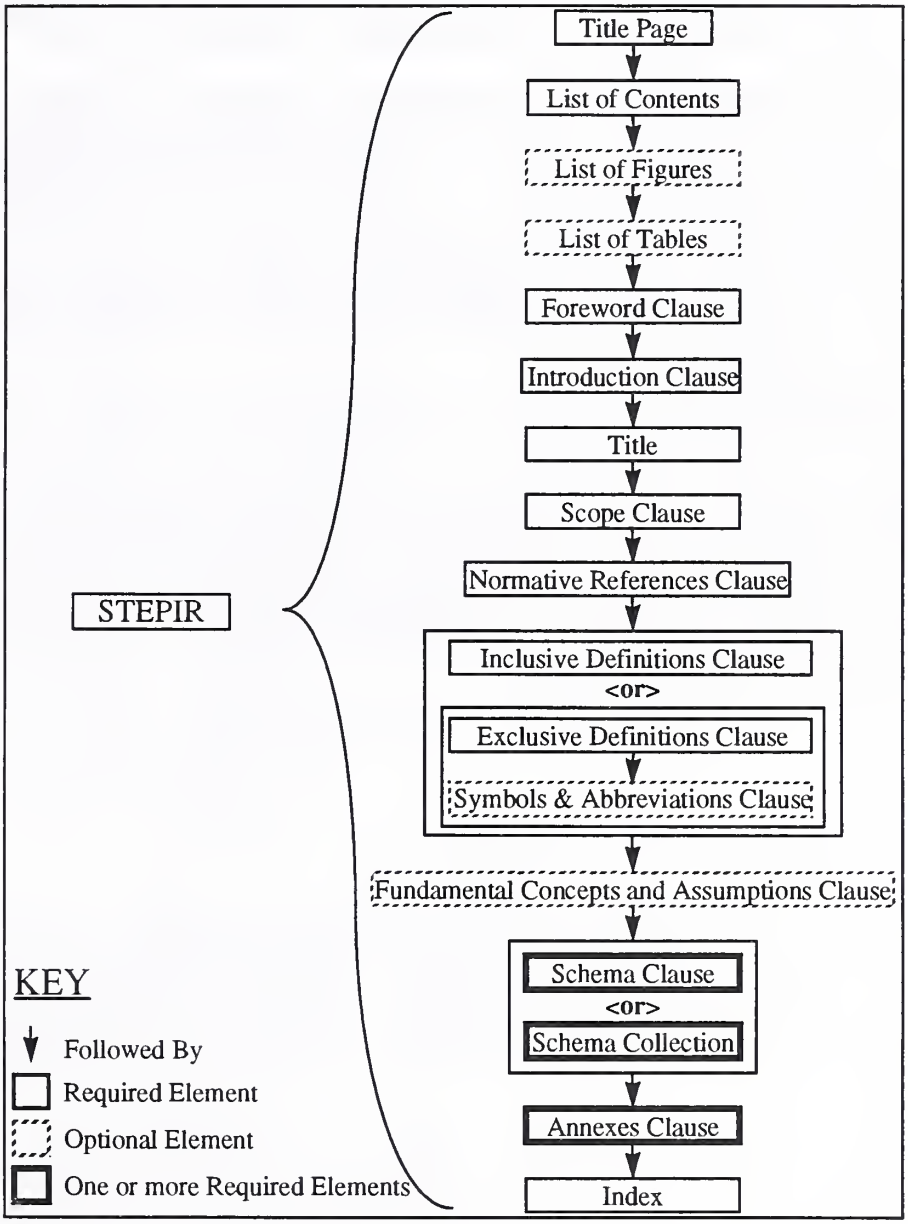
The model was then further developed into a Document Type Definition in the Standard Generalized Markup Language (SGML). The DTD will be used to convert existing STEP IR documents into SGML files. The DTD can also be used by a software system to create *new* STEP Integrated Resources.

Several existing DTD's were consulted at the onset of the project to determine whether or not they were applicable to the endeavor. The two most relevant documents were ISO_9573-11: Application at ISO Central Secretariat for International Standards and Technical Reports, First Edition[1] and MIL-M-28001-A: Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text[10]. Both of these documents are written to cover a wide domain of possible documents, and as such are very general and complex. They also make assumptions about document structure that did not apply in the case of the Integrated Resources. The STEP community would be better served by a DTD that was written within the context and vernacular of the STEP standard itself.

After a DTD is developed, it is processed by an SGML parser, which validates the DTD and compiles it into a set of rules. These rules govern the process of converting a document into SGML. The DTD models the structural elements of a document, and the conversion process entails identifying each such element in a document and encapsulating it within SGML delimiters, called tags. This process, called tagging, is often done within an SGML editor, which uses the rules to enforce the structural constraints for the document by only permitting the application of valid tags. The tags can be used by other

SGML-knowledgable software systems to identify the content of the document. The DTD presented in this document will be used to tag the STEP Integrated Resources, at which point it will be possible to automate their incorporation into a database system for the purposes of the APDE. The DTD will also serve to encode the Supplementary Directives in a formalized software system, thereby eliminating many vagaries of human interpretation that are inherent in work done by large committees across international boundaries.

Figure 1: The Structure of an Integrated Resource



Although it was known at the start of the DTD development that the components of STEP were living documents, there was no published documentation to indicate which versions of any document existed, what their differences were, or which was current.

Acquiring this information proved to be an elusive goal, accomplished by making inquiries through the STEP community until an individual was encountered who was recognizably knowledgeable and could provide the information sought.

Once a snapshot was taken depicting the state of STEP development in February 1993, the analysis of the documents began. There were a number of difficulties encountered during this beginning phase. First, the Supplementary Directives, ostensibly the canon for STEP Part development, were found often to be ambiguous, incomplete, or inconsistent. Second, even where the Directives were in fact reasonably clear, the various existing Integrated Resources were often not in compliance with them. Inquiries were made as to the reasons for these anomalies, but there seemed to be none.

It was decided that those elements of a particular Integrated Resource that were in egregious violation of the accepted conventions would be ignored for the purposes of the DTD. The implications of this decision are twofold. First, the DTD accurately reflects the structure of an Integrated Resource as they are supposed to be, rather than as some of them are. It was felt that this was the higher standard of work and the more correct path. Second, those Integrated Resources that are not in conformance with accepted standards must be brought into conformance before they can be integrated with the DTD or any system based thereon.

There was the occasional situation where confusion reigned. Either one of the Supplementary Directives was ambiguous and no example Part could be found to resolve the discrepancy, or there was divergence between groups of Parts. These issues were resolved through the attainment of consensus from available STEP experts. It is worth mentioning that whereas the Supplementary Directives are written in English and are subject to human interpretation, the DTD is a formal grammar and is not. As questions are answered and the answers are incorporated into the DTD, they will be formally and unambiguously codified.

4 The Annotated DTD for STEP IR Parts

Following is an annotation of the SGML DTD for STEP Parts. Constructs from the world of STEP are modelled in the DTD with SGML elements. The principal aspects of an element relevant to this discussion are its Generic Identifier (GI) and its content model. The GI is the name of the element. The content model is what the element contains, and is delineated by parentheses.

An entire STEP Part is represented in the DTD by the SGML element **STEP.Part**:

```
<!ELEMENT STEP.Part - - (STEP.Integrated.Resource | Other)
                      +(%General.Inclusions) >
```

A **STEP.Part** is either a **STEP.Integrated.Resource** or something **Other**: The **Other** element is defined as follows:

```
<!ELEMENT Other - - EMPTY >
```

All STEP Parts that are not Integrated Resources are represented in the DTD by the **Other** category, which is defined as **EMPTY.** The meaning of an empty element varies according to situation, but in this case it means that **Other** is a placeholder and is not really intended for use. When other STEP Parts are modelled and incorporated into the DTD, explicit entries will be added into the content model of **STEP.Part** and will depart from the realm of **Other**.

A **STEP.Part** also has inclusions, called **General.Inclusions**. An inclusion can occur any number of times (i.e., zero or more) anywhere within the element with which it is associated (unless it is excluded at some lower level). A **General.Inclusion**, therefore, is permissible anywhere in a **STEP.Part**. However, since **Other** is defined as being **EMPTY** it implicitly disallows **General.Inclusions**. Since the definition for **STEP.Integrated.Resource** has not yet been presented, all that can be known at this point is that **General.Inclusions** are prohibited in an **Other** but may be allowed in a **STEP.Integrated.Resource**.

```
<!ENTITY % General.Inclusions "Example | Figure    | Footnote |
                               Note    | Reference | Table"     >
```

A **General.Inclusion** is defined as a different SGML construct, called an Entity. For the purposes of this discussion, entities may be viewed as equivalent to elements, albeit with slightly different syntax. Any particular **General.Inclusion** is one of the following: **Example**, **Figure**, **Footnote**, **Note**, **Reference**, or **Table**. Four of these are defined as **Text**.

```
<!ELEMENT Example - - ( Text ) >
<!ELEMENT Footnote - - ( Text ) >
<!ELEMENT Note - - ( Text ) >
<!ELEMENT Reference - - ( Text ) >
```

Text is itself defined as Parsable Character Data (**PCDATA**), which means that it is ASCII text that may contain SGML tags. When an SGML data file is processed by a software system that understands SGML, **PCDATA** is understood to be data, but data that may contain SGML tags.

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

The other two **General.Inclusions**, **Figure** and **Table**, may contain graphics. Since SGML does not address graphics, these must be represented via some external mechanism, and the elements that represent them act as placeholders and are defined as **EMPTY**.

```
<!ELEMENT Figure - - EMPTY >
<!ELEMENT Table - - EMPTY >
```

The **STEP.Integrated.Resource** element represents the STEP Parts known as Integrated Resources. The content model for this element defines the structure of these documents, and consists of a sequence of constructs.[1]

```
<!ELEMENT STEP.Integrated.Resource - -  ( Title.Page,
                                          List.of.Contents,
                                          List.of.Figures?,
                                          List.of.Tables?,
                                          Foreword.C,
                                          Introduction.C,
                                          Title,
                                          Scope.C,
                                          Normative.References.C,
                                          ( Inclusive.Defs.C |
                                            ( Exclusive.Defs.C,
                                              Symbols.and.Abbrev.C?
                                            )
                                          ),
                                          FCandA.C?,
                                          ( Schema.C+ |
                                            Schema.Collection+
                                          ),
                                          Annexes.C,
                                          Index
                                        ) >
```

A construct in the content model for **STEP.Integrated.Resource** is either an element or a group. A group contains elements, groups, or a mixture of both.

The first construct in a **STEP.Integrated.Resource** is the **Title.Page**.

```
<!ELEMENT Title.Page - - ( Secretariat.Title.Page |
                           Placeholder.Title.Page ) >
```

[1] In SGML a "," can be read as "followed by" and preserves order; "&" is used to define a set of elements, "?"makes an item optional; a "+" is one or more items; a "*" is zero or more items.

In a published ISO standard, the title page is supplied by the ISO Central Secretariat. However, before the standard is published a placeholder title page is produced by those working on the standard. A **Title.Page**, therefore, is either a **Secretariat.Title.Page** or a **Placeholder.Title.Page**. Since a **Secretariat.Title.Page** is produced by the ISO Central Secretariat, it is not contained in the document itself, and is defined as **EMPTY**.

```
<!ELEMENT Secretariat.Title.Page - - EMPTY >
```

A **Placeholder.Title.Page** consists of a set of elements.

```
<!ELEMENT Placeholder.Title.Page - -    ( Part.Number &
                                          Part.Title &
                                          Date &
                                          Owner.Editor.Block &
                                          ISO.Designation &
                                          ISO.Predecessor &
                                          Document.Purpose &
                                          Abstract &
                                          Keywords &
                                          Current.Status &
                                          Document.Status &
                                          Comments.to.Reader
                                        ) >
```

Most of the components of the **Placeholder.Title.Page** are **Text**.

```
<!ELEMENT Part.Number           - -     ( Text ) >
<!ELEMENT Part.Title            - -     ( Text ) >
<!ELEMENT Date                  - -     ( Text ) >
<!ELEMENT Owner.Editor.Block    - -     ( Text ) >
<!ELEMENT Document.Purpose      - -     ( Text ) >
<!ELEMENT Abstract              - -     ( Text ) >
<!ELEMENT Keywords              - -     ( Text ) >
<!ELEMENT Current.Status        - -     ( Text ) >
<!ELEMENT Document.Status       - -     ( Text ) >
<!ELEMENT Comments.to.Reader    - -     ( Text ) >
```

Another component of the **Placeholder.Title.Page** is the **ISO.Designation**, which consists of an **ISO.Prefix** followed by an **ISO.Suffix**. An **ISO.Prefix** and an **ISO.Suffix** are **Text**.

```
<!ELEMENT ISO.Designation       - -     ( ISO.Prefix, ISO.Suffix ) >
<!ELEMENT ISO.Prefix            - -     ( Text ) >
<!ELEMENT ISO.Suffix            - -     ( Text ) >
```

The last component of the **Placeholder.Title.Page** consists of an **ISO.Suffix**.

```
<!ELEMENT ISO.Predecessor       - -     ( ISO.Suffix ) >
```

The **Title.Page** is followed by the **List.of.Contents**, which is the table of contents for the document. A table of contents has no content itself, strictly speaking, and is entirely derived from the remainder of the document. **List.of.Contents** should therefore be defined as **EMPTY**, and will be generated by some external software system. However, for the ease of tagging the existing STEP documents and current lack of a software system to generate content information, **List.of.Contents** is currently defined as **Text**. Any component of an Integrated Resource that is in the **List.of.Contents** is either a clause or a subclause. Each clause in the DTD is represented by an element whose name reflects the content of the clause with a **.C** appended. Subclauses are treated the same way, but the appendage is **.SubC**.

```
<!ELEMENT List.of.Contents - - Text >
```

Following the **List.of.Contents** is the **List.of.Figures**, which is required if there are figures in the document and prohibited if there are not. The **List.of.Figures** is defined as **Text** for the same reason the **List.of.Contents** is, and will be defined as **EMPTY** in the future. The **List.of.Tables** follows the **List.of.Figures**. It is required if there are tables in the document and prohibited if there are not. It too is defined as **Text** for the time being.

```
<!ELEMENT List.of.Figures - - Text >
<!ELEMENT List.of.Tables - - Text >
```

Following the **List.of.Tables** is the Foreword Clause (**Foreword.C**). In a published ISO standard, the foreword is supplied by the ISO Central Secretariat. However, before the standard is published a placeholder forward is produced by those working on the standard. A **Foreword.C**, therefore, is either a **Secretariat.Foreword** or a **Placeholder.Foreword**.

```
<!ELEMENT Foreword.C - - ( Secretariat.Foreword |
                           Placeholder.Foreword ) >
```

Since the final foreword is produced by the ISO Central Secretariat late in the process of document production, it is not modelled in the DTD; its location is marked by the **Secretariat.Foreword** element, which is defined as **EMPTY**.

```
 <!ELEMENT Secretariat.Foreword - - EMPTY >
```

The **Placeholder.Foreword**, however, is produced during the development of the document and is represented in the DTD. It consists of a heading (**Foreword.H**) followed by the **Text** of the Foreword.

```
<!ELEMENT Placeholder.Foreword - - ( Foreword.H, Text ) >
```

All headings in the document are represented by an element whose name consists of a **.H** appended to the name of the element for which it is a heading. Most headings begin with an identifying clause number (**C.Number**) that is derived from the logical location and scope of the clause in the document. Since they should be derived, and have no content of their own, the **C.Number** element will be defined as **EMPTY** in the future, as with the various List elements. For now, **C.Number** is defined as Text for the ease of tagging existing STEP documents. Following the **C.Number** is the **Text** of the heading.

```
<!ELEMENT Foreword.H - - ( C.Number, Text ) >
<!ELEMENT C.Number      - - Text >
```

Since most **.H** elements consist of a **C.Number** followed by **Text**, they will not be discussed further, although their declarations will appear in the annotation. Those that differ will be discussed. The reader may wonder why, if most **.H** elements are identical, there is not one element called, say, Heading, which is used in all such cases. This is done to allow future enhancements and differentiation to be done with minimal overhead. The same applies to some other elements which are not different yet are differentiated in this version of the DTD.

Following the Foreword Clause is the Introduction Clause, called **Introduction.C**, which consists of a heading (**Introduction.H**) followed by **Text**.

```
<!ELEMENT Introduction.C - - ( Introduction.H, Text ) >
<!ELEMENT Introduction.H - - ( C.Number, Text ) >
```

The formal **Title** of the document follows the Introduction. It is defined as **Text**.

```
<!ELEMENT Title - - ( Text ) >
```

The **Title** is followed by the Scope Clause, called **Scope.C**, which consists of, in sequence: a Scope Heading (**Scope.H**), **Text**, and zero or more Scope Subclauses (**Scope.SubC**). A **Scope.SubC** consists of a Scope Subclause Heading (**Scope.SubC.H**) followed by **Text**.

```
<!ELEMENT Scope.C - - ( Scope.H, Text, Scope.SubC* ) >
<!ELEMENT Scope.H - - ( C.Number, Text ) >
<!ELEMENT Scope.SubC - - ( Scope.SubC.H, Text ) >
<!ELEMENT Scope.SubC.H - - ( C.Number, Text ) >
```

The Scope Clause is followed by the Normative References Clause (**Normative.References.C**), which consists of, in sequence: a Normative References Heading (**Normative.References.H**), introductory text (**Intro.Text**), and a Normative Reference List (**Normative.Reference.List**). **Intro.Text** is **Text**. A **Normative.Reference.List** consists of one or more **Normative.References**. A **Normative.Reference** is **Text**.

```
<!ELEMENT Normative.References.C - -  ( Normative.References.H,
                                        Intro.Text,
                                        Normative.Reference.List
                                      ) >
<!ELEMENT Normative.References.H - - ( C.Number, Text ) >
<!ELEMENT Intro.Text - - ( Text ) >
<!ELEMENT Normative.Reference.List - - ( Normative.Reference+ ) >
<!ELEMENT Normtive.Reference - - ( Text ) >
```

The definitions, symbols, and abbreviations used in the document follow the Normative References Clause. An Integrated Resource either has a single clause (**Inclusive.Defs.C**) which presents the definitions, symbols, and abbreviations within subclauses, or it has a definitions-only clause (**Exclusive.Defs.C**) followed by a clause for symbols and abbreviations (**Symbols.and.Abbrev.C**).

An **Inclusive.Defs.C** opens with a heading (**Inclusive.Defs.C.H**). The definitions follow, in one or more Definition Block Subclauses (**Def.Block.SubC**). Next comes the Symbols Subclause (**Symbols.SubC**) and/or the Abbreviations Subclause (**Abbrev.-SubC**). The **Symbols.SubC** and the **Abbrev.SubC** are both optional; if both occur, either may come first.

```
<!ELEMENT Inclusive.Defs.C - -  ( Inclusive.Defs.C.H,
                                  Def.Block.SubC+,
                                  ( Symbols.SubC* & Abbrev.SubC* )
                                ) >
<!ELEMENT Inclusive.Defs.C.H - - ( C.Number, Text ) >
```

A definition block subclause presents a list of technical terms used in a document, either with their definitions, or with a reference to another document in which the definitions reside. The **Def.Block.SubC** element consists of, in sequence: a heading (**Def.Block.-SubC.H**), some body of introductory text (**Intro.Text**), and either a list of definition subclauses (**Def.List**) or a list of definition references (**Def.Reference.List**).

```
<!ELEMENT Def.Block.SubC - -  ( Def.Block.SubC.H,
                                Intro.Text,
                                ( Def.List | Def.Reference.List )
                              ) >
<!ELEMENT Def.Block.SubC.H - - ( C.Number, Text ) >
```

A **Def.List** consists of one or more definition subclauses (**Def.SubC**).

```
<!ELEMENT Def.List - - ( Def.SubC+ ) >
```

A **Def.SubC** presents a technical term and its definition. It consists of a heading (**Def.-SubC.H**), which contains the term to be defined, followed by the term's **Definition**. The heading, unlike most other **.H** elements, consists of a **C.Number** followed by a **Term**. A **Term** is **Text**. A **Definition** is also **Text**.

```
<!ELEMENT Def.SubC - - ( Def.SubC.H, Definition ) >
<!ELEMENT Def.SubC.H - - ( C.Number, Term ) >
<!ELEMENT Term - - ( Text ) >
<!ELEMENT Definition - - ( Text ) >
```

A **Def.Reference.List** consists of one or more **Terms**.

```
<!ELEMENT Def.Reference.List - - ( Term+ ) >
```

An **Exclusive.Defs.C** consists of a heading (**Exclusive.Defs.C.H**) followed by one or more Definition Block Subclauses (**Def.Block.SubC**).

```
<!ELEMENT Exclusive.Defs.C - - ( Exclusive.Defs.C.H, Def.Block.SubC+)>
<!ELEMENT Exclusive.Defs.C.H - - ( C.Number, Text ) >
```

A **Symbols.and.Abbrev.C** consists of a heading (**Symbols.and.Abbrev.C.H**) followed by, in no particular order, an optional Symbols Subclause (**Symbols.SubC**) and an optional Abbreviations Subclause (**Abbrev.SubC**). A **Symbols.SubC** consists of a heading (**Symbols.SubC.H**) followed by Symbol Definitions (**Symbol.Defs**). **Symbol.Defs** is defined as **Text**.

```
<!ELEMENT Symbols.and.Abbrev.C - - ( Symbols.and.Abbrev.C.H,
                                     ( Symbols.SubC* & Abbrev.SubC* )
                                     ) >
<!ELEMENT Symbols.and.Abbrev.C.H - - (C.Number, Text ) >
<!ELEMENT Symbols.SubC - - ( Symbols.SubC.H, Symbol.Defs ) >
<!ELEMENT Symbols.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Symbol.Defs - - ( Text ) >
```

An **Abbrev.SubC** consists of a heading (**Abbrev.SubC.H**) followed by a list of abbreviations (**Abbrev.List**). An **Abbrev.List** consists of one or more **Abbreviations**. An **Abbreviation** is a **Term**.

```
<!ELEMENT Abbrev.SubC - - ( Abbrev.SubC.H, Intro.Text, Abbrev.List ) >
<!ELEMENT Abbrev.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Abbrev.List - - ( Abbreviation+ ) >
<!ELEMENT Abbreviation - - ( Term ) >
```

An **FCandA.C** is a Fundamental Concepts and Assumptions Clause. Generally an **FCandA.C** applies to a particular schema, but if there is more than one schema in the Integrated Resource it is permissible to have a Fundamental Concepts and Assumptions Clause that applies to the entire Part. If there is such a Part-wide **FCandA.C**, it occurs after the clauses for definitions, symbols, and abbreviations.

There are two types of Fundamental Clauses and Assumptions: general and structured. Any **FCandA.C** consists of a heading (**FCandA.C.H**) followed by either a **General.FCandA** or a **Structured.FCandA**. A **General.FCandA** is **Text** and a **Structured.FCandA** is a **List**.

```
<!ELEMENT FCandA.C - -  ( FCandA.C.H,
                          ( General.FCandA |
                            Structured.FCandA
                          )
                        ) >
<!ELEMENT FCandA.C.H - - ( C.Number, Text ) >
<!ELEMENT General.FCandA - - ( Text ) >
<!ELEMENT Structured.FCandA - - ( List ) >
```

A **List** consists of optional introductory text (**List.Intro.Text**) followed by one or more items (**List.Item**). A **List.Item** is **Text**.

```
<!ELEMENT List - - ( List.Intro.Text?, List.Item+ ) >
<!ELEMENT List.Intro.Text - - ( Text ) >
<!ELEMENT List.Item - - ( Text ) >
```

The principal content of an Integrated Resource is its schemes. Each schema is documented in a Schema Clause (**Schema.C**). The Schema Clauses are presented in one of two ways, following the Part-wide **FCandA.C** (or the definitions, symbols, and abbreviations if there isn't one). The Schema Clauses are either presented in a series (of at least one), or they may be grouped into Schema Collections (**Schema.Collection**), which are themselves presented in series.

A **Schema.Collection** consists of at least two Schema Clauses, which may be preceded by one or more optional introductory clauses (**Schema.Collection.Intro.C**).

```
<!ELEMENT Schema.Collection - - ( Schema.Collection.Intro.C*,
                                  Schema.C,
                                  Schema.C+
                                ) >
```

A **Schema.Collection.Intro.C** consists of a heading (**Schema.Collection.Intro.C.H**) followed by **Text**.

```
<!ELEMENT Schema.Collection.Intro.C - - ( Schema.Collection.Intro.C.H,
                                          Text ) >
<!ELEMENT Schema.Collection.Intro.C.H - - ( C.Number, Text ) >
```

Each Schema Clause consists of a heading (**Schema.H**), an EXPRESS Declaration for the Schema (**Schema.EXPRESS.Decl**), and a series of documentation subclauses. The series consists of two required subclauses followed by four optional subclauses. The required subclauses are an introductory subclause (**Schema.Introduction.SubC**) and a Schema-specific **FCandA.C**. Following them come definitions for Types (**Type.Defs.Section**), Entities (**Entity.Defs.Section**), Rules (**Rule.Defs.Section**), and Functions (**Function.Def.Section**). Each Section is really a subclause, but may be a different sort of subclause depending on how its content.

```
<!ELEMENT Schema.C - -  ( Schema.H,
                          Schema.EXPRESS.Decl,
                          Schema.Introduction.SubC,
                          FCandA.C,
                          Type.Defs.Section?,
                          Entity.Defs.Section?,
                          Rule.Defs.Section?,
                          Function.Def.Section? ) >
<!ELEMENT Schema.H - - (C.Number, Text ) >
```

A Schema Clause is terminated by an End of Schema Declaration (**End.of.Schema.-Decl**), which occurs as the final item within the final Definition Subclause. Since the Definition Subclauses are optional, any one of them may be the final one, and so the **End.of.Schema.Decl** is an optional (final) component for each of them. However, this could theoretically give rise to a situation wherein there is more than one **End.of.Schema.Decl** in a Schema Clause. This is an error which will be corrected in the next version of the DTD.

The Schema EXPRESS Declaration consists of introductory text (**Schema.EXPRESS.Decl.Intro.Text**) followed by an EXPRESS Declaration (**EXPRESS.Decl**). It may also have an optional note (**Schema.Note**) at its end. Both **Schema.EXPRESS.-Decl.Intro.Text** and **Schema.Note** are **Text**.

```
<!ELEMENT Schema.EXPRESS.Decl - - ( Schema.EXPRESS.Decl.Intro.Text,
                                    EXPRESS.Decl,
                                    Schema.Note?
                                  ) >
<!ELEMENT Schema.EXPRESS.Decl.Intro.Text - - ( Text ) >
<!ELEMENT Schema.Note - - ( Text ) >
```

An **EXPRESS.Decl** consists of an introductory label (**EXPRESS.Decl.Label**) followed by EXPRESS code (**EXPRESS.Code**). The introductory label is **Text**, as is **EXPRESS.Code**.

```
<!ELEMENT EXPRESS.Decl - - (EXPRESS.Decl.Label, EXPRESS.Code ) >
<!ELEMENT EXPRESS.Decl.Label - - ( Text ) >
<!ELEMENT EXPRESS.Code - - ( Text ) >
```

The **Schema.Introduction.SubC** consists of a heading (**Schema.Introduction.H**) followed by **Text**.

```
<!ELEMENT Schema.Introduction.SubC - - ( Schema.Introduction.H,
                                          Text
                                        ) >
<!ELEMENT Schema.Introduction.H - - ( C.Number, Text ) >
```

The form of the **Type.Defs.Section** depends on whether there is one or more than one Type Definition in it. If there is only one the **Type.Defs.Section** is a Single Type Definition Subclause (**Single.Type.Def.SubC**). If there is more than one the **Type.Defs.-Section** is a Multiple Type Definitions Subclause (**Multiple.Type.Defs.SubC**). Additionally, if the Integrated Resource has no Entity, Function, or Rule Definitions, the **Type.Defs.Section** will end with an End of Schema Declaration (**End.of.Schema.-Decl**), as discussed previously.

```
<!ELEMENT Type.Defs.Section - - ( ( Single.Type.Def.SubC |
                                    Multiple.Type.Defs.SubC
                                  ),
                                  End.of.Schema.Decl?
                                ) >
```

A **Single.Type.Def.SubC** consists of a heading (**Single.Type.Def.SubC.H**) followed by a single Type Definition (**Type.Def**).

```
<!ELEMENT Single.Type.Def.SubC - - ( Single.Type.Def.SubC.H, Type.Def
                                   ) >
<!ELEMENT Single.Type.Def.SubC.H - - ( C.Number, Text ) >
```

A Type Definition consists of the following items, in sequence: introductory text (**Type.Def.Intro.Text**), an EXPRESS Type Declaration (**EXPRESS.Type.Decl**), an optional set of Enumerated Item Definitions (**Type.Def.Enumerated.Item.Defs**), an optional set of Formal Propositions (**Type.Def.Formal.Propositions**), and an optional set of Informal Propositions (**Type.Def.Informal.Propositions**).

```
<!ELEMENT Type.Def - -  ( Type.Def.Intro.Text,
                          EXPRESS.Type.Decl,
                          Type.Def.Enumerated.Item.Defs?,
                          Type.Def.Formal.Propositions?,
                          Type.Def.Informal.Propositions?
                        ) >
<!ELEMENT Type.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Type.Decl - - ( EXPRESS.Decl ) >
```

The Enumerated Item Definitions consist of introductory text followed by an Enumerated Item Definition List (**Enumerated.Item.Def.List**). An **Enumerated.Item.Def.List** is a **Def.List**.

```
<!ELEMENT Type.Def.Enumerated.Item.Defs - - (Intro.Text,
                                             Enumerated.Item.Def.List
                                            ) >
<!ELEMENT Enumerated.Item.Def.List - - ( Def.List) >
```

The Formal and Informal Propositions are both **Text**.

```
<!ELEMENT Type.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Type.Def.Informal.Propositions - - ( Text ) >
```

A **Multiple.Type.Defs.SubC** consists of a heading (**Multiple.Type.Defs.SubC.H**) followed by at least two Type Definition Subclauses (**Type.Def.SubC**). A **Type.Def.-SubC** consists of a heading (**Type.Def.SubC.H**) followed by a **Type.Def**.

```
<!ELEMENT Multiple.Type.Defs.SubC - - ( Multiple.Type.Defs.SubC.H,
                                        Type.Def.SubC,
                                        Type.Def.SubC+
                                      ) >
<!ELEMENT Multiple.Type.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Type.Def.SubC - - ( Type.Def.SubC.H, Type.Def ) >
<!ELEMENT Type.Def.SubC.H - - ( C.Number, Text ) >
```

An **End.of.Schema.Decl** is an **EXPRESS.Decl**.

```
<!ELEMENT End.of.Schema.Decl - - ( EXPRESS.Decl ) >
```

The Entity Definitions Section is similar to the Type Definitions Section , in that it may consist of a single Entity Definition (**Single.Entity.Def.SubC**) or multiple ones (**Multiple.Entity.Defs.SubC**). It differs, however, in that it may also consist of one or more logical groups of Entity Definitions (**Entity.Def.Logical.Group.SubC**). If there are no Rule or Function Definitions in the Schema the Entity Definitions Section concludes with an **End.of.Schema.Decl**.

```
<!ELEMENT Entity.Defs.Section - - ( ( Single.Entity.Def.SubC |
                                      Multiple.Entity.Defs.SubC |
                                      Entity.Def.Logical.Group.SubC+
                                   .),
                                    End.of.Schema.Decl?
                                  ) >
```

A **Single.Entity.Def.SubC** consists of a heading (**Single.Entity.Def.SubC.H**) followed by one Entity Definition (**Entity.Def**).

```
<!ELEMENT Single.Entity.Def.SubC - - ( Single.Entity.Def.SubC.H,
                           Entity.Def ) >
<!ELEMENT Single.Entity.Def.SubC.H - - ( C.Number, Text ) >
```

An **Entity.Def** is basically similar to a **Type.Def**, differing in that it contains Attribute Definitions (**Entity.Def.Attribute.Defs**) instead of Enumerated Item Definitions.

```
<!ELEMENT Entity.Def - -        ( Entity.Def.Intro.Text,
                                  EXPRESS.Entity.Decl,
                                  Entity.Def.Attribute.Defs?,
                                  Entity.Def.Formal.Propositions?,
                                  Entity.Def.Informal.Propositions?
                                ) >
<!ELEMENT Entity.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Entity.Decl - - ( EXPRESS.Decl ) >

<!ELEMENT Entity.Def.Attribute.Defs - - ( Intro.Text,
                                          Attribute.Def.List
                                        ) >
<!ELEMENT Attribute.Def.List - - ( Def.List ) >
<!ELEMENT Entity.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Entity.Def.Informal.Propositions - - ( Text ) >
```

A **Multiple.Entity.Defs.SubC** is structured the same as a **Multiple.Type.Defs.SubC**.

```
<!ELEMENT Multiple.Entity.Defs.SubC - - ( Multiple.Entity.Defs.SubC.H,
                                          Entity.Def.SubC,
                                          Entity.Def.SubC+
                                        ) >
<!ELEMENT Multiple.Entity.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Entity.Def.SubC - - ( Entity.Def.SubC.H, Entity.Def ) >
<!ELEMENT Entity.Def.SubC.H - - ( C.Number, Text ) >
```

An **Entity.Def.Logical.Group.SubC** is structurally identical to a **Multiple.Entity.-Defs.SubC**, differing only in the text of its headers (which is not characterized in a DTD).

```
<!ELEMENT Entity.Def.Logical.Group.SubC - - (
                                    Entity.Def.Logical.Group.SubC.H,
                                    Entity.Def.SubC,
                                    Entity.Def.SubC+
                                                ) >
<!ELEMENT Entity.Def.Logical.Group.SubC.H - - ( C.Number, Text ) >
```

The Entity Drfinitions are followed by the Rule Definitions. The **Rule.Defs.Section** is structurally similar to the Type Definitions, differing only in that a Rule Definition contains Dependencies (**Rule.Def.Dependency**) instead of Informal Propositions.

```
<!ELEMENT Rule.Defs.Section - - ( ( Single.Rule.Def.SubC |
                                    Multiple.Rule.Defs.SubC
                                  ),
                                  End.of.Schema.Decl?
                                ) >
<!ELEMENT Single.Rule.Def.SubC - - ( Single.Rule.Def.SubC.H, Rule.Def
                                   ) >
<!ELEMENT Single.Rule.Def.SubC.H - - ( C.Number, Text ) >
```

```
<!ELEMENT Rule.Def - -  ( Rule.Def.Intro.Text,
                          EXPRESS.Rule.Decl,
                          Rule.Def.Argument.Defs?,
                          Rule.Def.Formal.Propositions?,
                          Rule.Def.Dependency*
                        ) >
<!ELEMENT Rule.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Rule.Decl - - ( EXPRESS.Decl ) >
<!ELEMENT Rule.Def.Argument.Defs - - ( Argument.Def.List ) >
<!ELEMENT Argument.Def.List - - ( Def.List ) >
<!ELEMENT Rule.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Rule.Def.Dependency - - ( Text ) >



<!ELEMENT Multiple.Rule.Defs.SubC - - ( Multiple.Rule.Defs.SubC.H,
                                        Rule.Def.SubC,
                                        Rule.Def.SubC+
                                      ) >
<!ELEMENT Multiple.Rule.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Rule.Def.SubC - - ( Rule.Def.SubC.H, Rule.Def ) >
<!ELEMENT Rule.Def.SubC.H - - ( C.Number, Text ) >
```

The Rule Definitions are followed by the Function Definitions. The **Function.Defs.Section** is similar to but simpler than the Rule Definitions Section.

```
<!ELEMENT Function.Def.Section - - ( ( Single.Function.Def.SubC |
                                       Multiple.Function.Defs.SubC
                                     ),
                                     End.of.Schema.Decl?
                                   ) >

<!ELEMENT Single.Function.Def.SubC - - ( Single.Function.Def.SubC.H,
                                         Function.Def ) >
<!ELEMENT Single.Function.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Function.Def - -      ( Function.Description,
                                  EXPRESS.Function.Decl,
                                  Function.Argument.Defs?
                                ) >
<!ELEMENT Function.Description - - ( Text ) >
<!ELEMENT EXPRESS.Function.Decl - - ( EXPRESS.Code ) >
<!ELEMENT Function.Argument.Defs - - ( Argument.Def.List ) >

<!ELEMENT Multiple.Function.Defs.SubC - - (
                                      Multiple.Function.Defs.SubC.H,
                                      Function.Def.SubC,
                                      Function.Def.SubC+
                                          ) >
<!ELEMENT Multiple.Function.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Function.Def.SubC - - ( Function.Def.SubC.H, Function.Def+ )
>
<!ELEMENT Function.Def.SubC.H - - ( C.Number, Text ) >
```

Following the Schema Clauses or Schema Collections come the annexes (**Annexes.C**). There are Normative Annexes and Informative Annexes, and the former precede the latter. **Annexes.C** consists of a heading (**Annexes.H**) followed by the annexes.

```
<!ELEMENT Annexes.C - -          ( Annexes.H,
                                   Entity.Short.Names.Annex,
                                   EXPRESS.Listing.Annex,
                                   Bibliography.Annex?,
                                   Model.Scope.Annex?,
                                   Informative.Annex.C
                                 ) >
<!ELEMENT Annexes.H - - ( C.Number, Text ) >
```

The first annex contains Entity Short Names (**Entity.Short.Names.Annex**) and is a Normative Annex (**Normative.Annex.C**). A **Normative.Annex.C** consists of a heading (**Normative.Annex.H**), a label identifying the annex as normative (**Normative.Annex.Tag**), the official title of the annex (**Normative.Annex.Title**), and the **Text** of the annex. The tag and the title are **Text**.

```
<!ELEMENT Entity.Short.Names.Annex - - ( Normative.Annex.C ) >

<!ELEMENT Normative.Annex.C - - ( Normative.Annex.H,
                                  Normative.Annex.Tag,
                                  Normative.Annex.Title,
                                  Text
                                ) >
<!ELEMENT Normative.Annex.H - - ( C.Number, Text ) >
<!ELEMENT Normative.Annex.Tag - - ( Text ) >
<!ELEMENT Normative.Annex.Title - - ( Text ) >
```

The second annex contains the EXPRESS Listing for the Part (**EXPRESS.Listing.Annex**), and is an **Informative.Annex.C**. An **Informative.Annex.C** is structurally identical to a **Normative.Annex.C**.

```
<!ELEMENT EXPRESS.Listing.Annex - - ( Informative.Annex.C ) >

<!ELEMENT Informative.Annex.C - -       ( Informative.Annex.H,
                                          Informative.Annex.Tag,
                                          Informative.Annex.Title,
                                          Text
                                        ) >
<!ELEMENT Informative.Annex.H - - ( C.Number, Text ) >
<!ELEMENT Informative.Annex.Tag - - ( Text ) >
<!ELEMENT Informative.Annex.Title - - ( Text ) >
```

The **EXPRESS.Listing.Annex** is followed by the **Bibliography.Annex** and the **Model.Scope.Annex**, each of which is an **Informative.Annex.C**.

```
<!ELEMENT Bibliography.Annex - - ( Informative.Annex.C ) >
<!ELEMENT Model.Scope.Annex - - ( Informative.Annex.C ) >
```

The final element in an Integrated Resource is the **Index**. Since the **Index** should be generated entirely from the text of the document, it should be defined to be **EMPTY**. However, for the ease of tagging the existing STEP documents, it is currently defined as **Text**.

```
<!ELEMENT Index - - Text >
```

5 Conclusion

The DTD presently addresses the structure of an Integrated Resource. The Supplementary Directives also mandate various matters of content and wording, which have not been addressed by this version of the DTD and will be addressed in subsequent ones. The DTD presented herein is an initial draft, and is undergoing further development and refinement. The DTD will have progressed further by the time this document is published, but should nevertheless give strong indication of the direction in which the work is going, and serve as an invitation for comment and feedback by the community for which the work is being done.

Appendix A: The DTD for STEP Integrated Resources

```
<!--************************************************************************-->
<!--                                                                        -->
<!--      An SGML Document Type Def for component documents of              -->
<!--      the Standard for the Exchange of Product Model Data (STEP).       -->
<!--                                                                        -->
<!--      Author: Scott Bodarky [bodarky@cme.nist.gov]                      -->
<!--      $Revision: 1.4 $                                                  -->
<!--      $Date: 1993/05/17 14:33:17 $                                      -->
<!--                                                                        -->
<!--      Naming Conventions (that might be vaguely inscrutable):           -->
<!--                                                                        -->
<!--             Abbrev    Abbreviation(s)                                  -->
<!--             C         Clause                                           -->
<!--             Decl      Declaration                                      -->
<!--             Def(s)    Definition(s)                                    -->
<!--             FCandA    Fundamental Concepts and Assumptions             -->
<!--             H         Heading                                          -->
<!--             Intro     (Introduction,Introductory)                      -->
<!--             SubC      Subclause                                        -->
<!--                                                                        -->
<!--************************************************************************-->

<!--************************************************************************-->
<!-- [Level x] General Entity Definitions                                   -->
<!--                                                                        -->
<!ENTITY % General.Inclusions "Example | Figure     | Footnote |
                               Note    | Reference | Table"     >

<!--************************************************************************-->
<!-- [Level 0] STEP.Document                                                -->
<!--                                                                        -->
<!ELEMENT STEP.Part - - (STEP.Integrated.Resource | Other)
                       +(%General.Inclusions) >


<!--************************************************************************-->
<!-- [Level x] Global "Variables"                                           -->
<!--                                                                        -->
<!ELEMENT C.Number      - - Text >
<!ELEMENT Example - - ( Text ) >
<!ELEMENT Figure - - EMPTY >
<!ELEMENT Footnote - - ( Text ) >
<!ELEMENT Intro.Text - - ( Text ) >
<!ELEMENT List - - ( List.Intro.Text?, List.Item+ ) >
<!ELEMENT List.Intro.Text - - ( Text ) >
<!ELEMENT List.Item - - ( Text ) >
<!ELEMENT Note - - ( Text ) >
<!ELEMENT Reference - - ( Text ) >
<!ELEMENT Table - - EMPTY >
<!ELEMENT Text - - ( #PCDATA ) >
```

```
<!--***********************************************************************-->
<!-- [Level 1] STEP.Document: STEP Integrated Resource                     -->
<!--                                                                       -->
<!ELEMENT STEP.Integrated.Resource - -  ( Title.Page,
                                          List.of.Contents,
                                          List.of.Figures?,
                                          List.of.Tables?,
                                          Foreword.C,
                                          Introduction.C,
                                          Title,
                                          Scope.C,
                                          Normative.References.C,
                                          ( Inclusive.Defs.C |
                                            ( Exclusive.Defs.C,
                                              Symbols.and.Abbrev.C?
                                            )
                                          ),
                                          FCandA.C?,
                                          ( Schema.C+ |
                                            Schema.Collection+
                                          ),
                                          Annexes.C,
                                          Index
                                        ) >
<!ELEMENT Other - - EMPTY >


<!--***********************************************************************-->
<!-- [Level 1.x] Elements and Data Types Common to all STEP IR's           -->
<!--                                                                       -->
<!ELEMENT Def.List - - ( Def.SubC+ ) >
<!ELEMENT Def.SubC - - ( Def.SubC.H, Definition ) >
<!ELEMENT Def.SubC.H - - ( C.Number, Term ) >
<!ELEMENT Term - - ( Text ) >
<!ELEMENT Definition - - ( Text ) >
<!ELEMENT EXPRESS.Decl - - (EXPRESS.Decl.Label, EXPRESS.Code ) >
<!ELEMENT EXPRESS.Decl.Label - - ( Text ) >
<!ELEMENT EXPRESS.Code - - ( Text ) >
<!ELEMENT FCandA.C - -  ( FCandA.C.H,
                          ( General.FCandA |
                            Structured.FCandA
                          )
                        ) >
<!ELEMENT FCandA.C.H - - ( C.Number, Text ) >
<!ELEMENT General.FCandA - - ( Text ) >
<!ELEMENT Structured.FCandA - - ( List ) >
```

```
<!--************************************************************************-->
<!-- [Level 2-3] STEP IR: Title Page                                        -->
<!--                                                                        -->
<!ELEMENT Title.Page - - ( Secretariat.Title.Page | Placeholder.Title.Page ) >
<!ELEMENT Secretariat.Title.Page - - EMPTY >
<!ELEMENT Placeholder.Title.Page - -    ( ( Part.Number &
                                            Part.Title &
                                            Date &
                                            Owner.Editor.Block &
                                            ISO.Designation &
                                            ISO.Predecessor
                                          ) &
                                          ( Document.Purpose &
                                            Abstract &
                                            Keywords &
                                            Current.Status &
                                            Document.Status &
                                            Comments.to.Reader
                                          )
                                        ) >

<!--************************************************************************-->
<!-- [Level 2.x] Elements and Data Types Common to all IR Def Cs -->
<!--                                                                        -->
<!ELEMENT Argument.Def.List - - ( Def.List ) >
<!ELEMENT End.of.Schema.Decl - - ( EXPRESS.Decl ) >

<!--           *************************************************************-->
<!-- [Level 4] STEP IR Placeholder Title Page: Components                   -->
<!--                                                                        -->
<!ELEMENT Abstract              - -     ( Text ) >
<!ELEMENT Comments.to.Reader    - -     ( Text ) >
<!ELEMENT Current.Status        - -     ( Text ) >
<!ELEMENT Date                  - -     ( Text ) >
<!ELEMENT Document.Purpose      - -     ( Text ) >
<!ELEMENT Document.Status       - -     ( Text ) >
<!ELEMENT ISO.Designation       - -     ( ISO.Prefix, ISO.Suffix ) >
<!ELEMENT ISO.Prefix            - -     ( Text ) >
<!ELEMENT ISO.Suffix            - -     ( Text ) >
<!ELEMENT Keywords              - -     ( Text ) >
<!ELEMENT Owner.Editor.Block    - -     ( Text ) >
<!ELEMENT Part.Number           - -     ( Text ) >
<!ELEMENT Part.Title            - -     ( Text ) >
<!ELEMENT ISO.Predecessor       - -     ( ISO.Suffix ) >

<!--************************************************************************-->
<!-- [Level 2] STEP IR: Generated Lists, of Contents, Figures, and Tables   -->
<!--                                                                        -->
<!ELEMENT List.of.Contents - - Text >
<!ELEMENT List.of.Figures - - Text >
<!ELEMENT List.of.Tables - - Text >
```

```
<!--***********************************************************************-->
<!-- [Level 2+] STEP IR: Foreword Clause                                   -->
<!--                                                                       -->
<!ELEMENT Foreword.C - - ( Secretariat.Foreword | Placeholder.Foreword ) >
<!ELEMENT Placeholder.Foreword - - ( Foreword.H, Text ) >
<!ELEMENT Secretariat.Foreword - - EMPTY >
<!ELEMENT Foreword.H - - ( C.Number, Text ) >


<!--***********************************************************************-->
<!-- [Level 2+] STEP IR: Introduction Clause                               -->
<!--                                                                       -->
<!ELEMENT Introduction.C - - ( Introduction.H, Text ) >
<!ELEMENT Introduction.H - - ( C.Number, Text ) >

<!--***********************************************************************-->
<!-- [Level 2] STEP IR: Official Title of the STEP IR                      -->
<!--                                                                       -->
<!ELEMENT Title - - ( Text ) >

<!--***********************************************************************-->
<!-- [Level 2+] STEP IR: Scope Clause                                      -->
<!--                                                                       -->
<!ELEMENT Scope.C - - ( Scope.H, Text, Scope.SubC* ) >
<!ELEMENT Scope.H - - ( C.Number, Text ) >
<!ELEMENT Scope.SubC - - ( Scope.SubC.H, Text ) >
<!ELEMENT Scope.SubC.H - - ( C.Number, Text ) >

<!--***********************************************************************-->
<!-- [Level 2+] STEP IR: Normative References Clause                       -->
<!--                                                                       -->
<!ELEMENT Normative.References.C - -  ( Normative.References.H,
                                        Intro.Text,
                                        Normative.Reference.List
                                      ) >
<!ELEMENT Normative.References.H - - ( C.Number, Text ) >
<!ELEMENT Normative.Reference.List - - ( Normative.Reference+ ) >
<!ELEMENT Normative.Reference - - ( Text ) >

<!--*********************************************************************-->
<!-- [Level 2] STEP IR: Inclusive Definitions Clause                       -->
<!--                   Exclusive Definitions Clause                        -->
<!--                   Symbols and Abbreviations Clause                    -->
<!--                                                                       -->
<!ELEMENT Inclusive.Defs.C - -  ( Inclusive.Defs.C.H,
                                  Def.Block.SubC+,
                                  ( Symbols.SubC* & Abbrev.SubC* )
                                ) >
<!ELEMENT Exclusive.Defs.C - - ( Exclusive.Defs.C.H, Def.Block.SubC+ ) >
<!ELEMENT Symbols.and.Abbrev.C - - ( Symbols.and.Abbrev.C.H,
                                     ( Symbols.SubC* & Abbrev.SubC* )
                                   ) >
```

```
<!--           ************************************************************-->
<!-- [Level 3] STEP IR Inclusive Definitions C: Components                 -->
<!--                  Exclusive Definitions C: Components                  -->
<!--                  Symbols and Abbreviations C: Components              -->
<!--                                                                       -->
<!ELEMENT Abbrev.SubC - - ( Abbrev.SubC.H, Intro.Text, Abbrev.List ) >
<!ELEMENT Def.Block.SubC - -  ( Def.Block.SubC.H,
                                Intro.Text,
                                ( Def.List | Def.Reference.List )
                              ) >
<!ELEMENT Exclusive.Defs.C.H - - ( C.Number, Text ) >
<!ELEMENT Inclusive.Defs.C.H - - ( C.Number, Text ) >
<!ELEMENT Symbols.and.Abbrev.C.H - - (C.Number, Text ) >
<!ELEMENT Symbols.SubC - - ( Symbols.SubC.H, Symbol.Defs ) >

<!--         **************************************************************-->
<!-- [Level 4+] STEP IR Abbreviations SubC: Components                     -->
<!--                                                                       -->
<!ELEMENT Abbrev.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Abbrev.List - - ( Abbreviation+ ) >
<!ELEMENT Abbreviation - - ( Term ) >

<!--         **************************************************************-->
<!-- [Level 4+] STEP IR Definition Block SubC: Components                  -->
<!--                                                                       -->
<!ELEMENT Def.Block.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Def.Reference.List - - ( Term+ ) >

<!--         **************************************************************-->
<!-- [Level 4] STEP IR Symbols SubC: Components                            -->
<!--                                                                       -->
<!ELEMENT Symbols.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Symbol.Defs - - ( Text ) >

<!--***********************************************************************-->
<!-- [Level 2] STEP IR: Schema C,                                          -->
<!--                   Schema Collection                                   -->
<!--                                                                       -->
<!ELEMENT Schema.C - -  ( Schema.H,
                          Schema.EXPRESS.Decl,
                          Schema.Introduction.SubC
                          FCandA.C,
                          Type.Defs.Section?,
                          Entity.Defs.Section?,
                          Rule.Defs.Section?,
                          Function.Def.Section?
                        ) >

<!ELEMENT Schema.Collection - - ( Schema.Collection.Intro.C*,
                                  Schema.C,
                                  Schema.C+
                                ) >
```

```
<!--         ***************************************************************-->
<!-- [Level 3] STEP IR Schema C: Components                                 -->
<!--                                                                        -->
<!ELEMENT Schema.H - - (C.Number, Text ) >
<!ELEMENT Schema.EXPRESS.Decl - - ( Schema.EXPRESS.Decl.Intro.Text,
                                    EXPRESS.Decl,
                                    Schema.Note?
                                  ) >
<!ELEMENT Schema.Introduction.SubC - -  ( Schema.Introduction.H,
                                          Text
                                        ) >
<!ELEMENT Type.Defs.Section - - ( ( Single.Type.Def.SubC |
                                    Multiple.Type.Defs.SubC
                                  ),
                                  End.of.Schema.Decl?
                                ) >
<!ELEMENT Entity.Defs.Section - -       ( ( Single.Entity.Def.SubC |
                                            Multiple.Entity.Defs.SubC |
                                            Entity.Def.Logical.Group.SubC+
                                          ),
                                          End.of.Schema.Decl?
                                        ) >
<!ELEMENT Rule.Defs.Section - - ( ( Single.Rule.Def.SubC |
                                    Multiple.Rule.Defs.SubC
                                  ),
                                  End.of.Schema.Decl?
                                ) >
<!ELEMENT Function.Def.Section - -      ( ( Single.Function.Def.SubC |
                                            Multiple.Function.Defs.SubC
                                          ),
                                          End.of.Schema.Decl?
                                        ) >

<!--         ***************************************************************-->
<!-- [Level 4] STEP IR Schema EXPRESS Declaration: Components               -->
<!--                                                                        -->
<!ELEMENT Schema.EXPRESS.Decl.Intro.Text - - ( Text ) >
<!ELEMENT Schema.Note - - ( Text ) >

<!--         ***************************************************************-->
<!-- [Level 4] STEP IR Schema Introduction SubC: Components                 -->
<!--                                                                        -->
<!ELEMENT Schema.Introduction.H - - ( C.Number, Text ) >

<!--         ***************************************************************-->
<!-- [Level 4+] STEP IR Schema Type Definitions Section: Components         -->
<!--                                                                        -->
<!ELEMENT Single.Type.Def.SubC - - ( Single.Type.Def.SubC.H, Type.Def ) >
<!ELEMENT Multiple.Type.Defs.SubC - - ( Multiple.Type.Defs.SubC.H,
                                        Type.Def.SubC,
                                        Type.Def.SubC+
                                      ) >
<!ELEMENT Single.Type.Def.SubC.H - - ( C.Number, Text ) >
```

```
<!ELEMENT Type.Def - -  ( Type.Def.Intro.Text,
                          EXPRESS.Type.Decl,
                          Type.Def.Enumerated.Item.Defs?,
                          Type.Def.Formal.Propositions?,
                          Type.Def.Informal.Propositions?
                        ) >
<!ELEMENT Multiple.Type.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Type.Def.SubC - - ( Type.Def.SubC.H, Type.Def ) >
<!ELEMENT Type.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Type.Decl - - ( EXPRESS.Decl ) >
<!ELEMENT Type.Def.Enumerated.Item.Defs - - ( Intro.Text,
                                              Enumerated.Item.Def.List
                                            ) >
<!ELEMENT Type.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Type.Def.Informal.Propositions - - ( Text ) >
<!ELEMENT Type.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Enumerated.Item.Def.List - - ( Def.List) >

<!--         ***************************************************************-->
<!-- [Level 4+] STEP IR Schema Entity Definitions Section: Components       -->
<!--                                                                        -->
<!ELEMENT Single.Entity.Def.SubC - - ( Single.Entity.Def.SubC.H, Entity.Def ) >
<!ELEMENT Multiple.Entity.Defs.SubC - - ( Multiple.Entity.Defs.SubC.H,
                                          Entity.Def.SubC,
                                          Entity.Def.SubC+
                                        ) >
<!ELEMENT Entity.Def.Logical.Group.SubC - - ( Entity.Def.Logical.Group.SubC.H,
                                              Entity.Def.SubC,
                                              Entity.Def.SubC+
                                            ) >
<!ELEMENT Single.Entity.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Entity.Def - -        ( Entity.Def.Intro.Text,
                                  EXPRESS.Entity.Decl,
                                  Entity.Def.Attribute.Defs?,
                                  Entity.Def.Formal.Propositions?,
                                  Entity.Def.Informal.Propositions?
                                ) >
<!ELEMENT Multiple.Entity.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Entity.Def.SubC - - ( Entity.Def.SubC.H, Entity.Def ) >
<!ELEMENT Entity.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Entity.Decl - - ( EXPRESS.Decl ) >
<!ELEMENT Entity.Def.Attribute.Defs - - ( Intro.Text,
                                          Attribute.Def.List
                                        ) >
<!ELEMENT Entity.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Entity.Def.Informal.Propositions - - ( Text ) >
<!ELEMENT Entity.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Entity.Def.Logical.Group.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Attribute.Def.List - - ( Def.List ) >
```

```
<!--         **************************************************************-->
<!-- [Level 4+] STEP IR Schema Rule Definitions Section: Components        -->
<!--                                                                       -->
<!ELEMENT Single.Rule.Def.SubC - - ( Single.Rule.Def.SubC.H, Rule.Def ) >
<!ELEMENT Multiple.Rule.Defs.SubC - - ( Multiple.Rule.Defs.SubC.H,
                                        Rule.Def.SubC,
                                        Rule.Def.SubC+
                                      ) >
<!ELEMENT Single.Rule.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Rule.Def - -  ( Rule.Def.Intro.Text,
                          EXPRESS.Rule.Decl,
                          Rule.Def.Argument.Defs?,
                          Rule.Def.Formal.Propositions?,
                          Rule.Def.Dependency*
                        ) >
<!ELEMENT Multiple.Rule.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Rule.Def.SubC - - ( Rule.Def.SubC.H, Rule.Def ) >
<!ELEMENT Rule.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Rule.Def.Intro.Text - - ( Text ) >
<!ELEMENT EXPRESS.Rule.Decl - - ( EXPRESS.Decl ) >
<!ELEMENT Rule.Def.Argument.Defs - - ( Argument.Def.List ) >
<!ELEMENT Rule.Def.Formal.Propositions - - ( Text ) >
<!ELEMENT Rule.Def.Dependency - - ( Text ) >

<!--         **************************************************************-->
<!-- [Level 4+] STEP IR Schema Function Definitions Section: Components    -->
<!--                                                                       -->
<!ELEMENT Single.Function.Def.SubC - - ( Single.Function.Def.SubC.H,
                                         Function.Def ) >
<!ELEMENT Multiple.Function.Defs.SubC - -     ( Multiple.Function.Defs.SubC.H,
                                                 Function.Def.SubC,
                                                 Function.Def.SubC+
                                               ) >
<!ELEMENT Single.Function.Def.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Function.Def - -      ( Function.Description,
                                  EXPRESS.Function.Decl,
                                  Function.Argument.Defs?
                                ) >
<!ELEMENT Multiple.Function.Defs.SubC.H - - ( C.Number, Text ) >
<!ELEMENT Function.Def.SubC - - ( Function.Def.SubC.H, Function.Def+ ) >
<!ELEMENT Function.Description - - ( Text ) >
<!ELEMENT EXPRESS.Function.Decl - - ( EXPRESS.Code ) >
<!ELEMENT Function.Argument.Defs - - ( Argument.Def.List ) >
<!ELEMENT Function.Def.SubC.H - - ( C.Number, Text ) >

<!--         **************************************************************-->
<!-- [Level 3+] STEP IR Schema Collection: Components                      -->
<!--                                                                       -->
<!ELEMENT Schema.Collection.Intro.C - - ( Schema.Collection.Intro.C.H, Text ) >
<!ELEMENT Schema.Collection.Intro.C.H - - ( C.Number, Text ) >
```

```
<!--***********************************************************************-->
<!-- [Level 2+] STEP IR: Annexes Clause                                    -->
<!--                                                                       -->
<!ELEMENT Annexes.C - -         ( Annexes.H,
                                  Entity.Short.Names.Annex,
                                  EXPRESS.Listing.Annex,
                                  Bibliography.Annex?,
                                  Model.Scope.Annex?,
                                  Informative.Annex.C
                                ) >
<!ELEMENT Annexes.H - - ( C.Number, Text ) >
<!ELEMENT Entity.Short.Names.Annex - - ( Normative.Annex.C ) >
<!ELEMENT EXPRESS.Listing.Annex - - ( Informative.Annex.C ) >
<!ELEMENT Bibliography.Annex - - ( Informative.Annex.C ) >
<!ELEMENT Model.Scope.Annex - - ( Informative.Annex.C ) >
<!ELEMENT Informative.Annex.C - -       ( Informative.Annex.H,
                                          Informative.Annex.Tag,
                                          Informative.Annex.Title,
                                          Text
                                        ) >
<!ELEMENT Normative.Annex.C - - ( Normative.Annex.H,
                                  Normative.Annex.Tag,
                                  Normative.Annex.Title,
                                  Text
                                ) >
<!ELEMENT Informative.Annex.H - - ( C.Number, Text ) >
<!ELEMENT Informative.Annex.Tag - - ( Text ) >
<!ELEMENT Informative.Annex.Title - - ( Text ) >
<!ELEMENT Normative.Annex.H - - ( C.Number, Text ) >
<!ELEMENT Normative.Annex.Tag - - ( Text ) >
<!ELEMENT Normative.Annex.Title - - ( Text ) >

<!--***********************************************************************-->
<!-- [Level 2] STEP IR: Index                                              -->
<!--                                                                       -->
<!ELEMENT Index - - Text >
```

Appendix B: References

[1] Information Processing - SGML Support Facilities - Techniques for using SGML - Part 11: Application at ISO Central Secretariat for International Standards and Technical Reports. First Edition, TR9573 PT11, 82 pages, 1992

[2] ISO DIS 10303-1, Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles, International Organization for Standardization, January 6, 1993

[3] ISO/DIS 10303-11, Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual, International Organization for Standardization, August 31, 1992.

[4] ISO DIS 10303 – 41, Product Data Representation and Exchange – Part 41, Fundamentals of Product Description and Support, ISO TC184/SC4 Document N197, March 22, 1993.

[5] ISO DIS 10303 - 42, Product Data Representation and Exchange – Part 42, Integrated Resources: Geometric and Topological Representation, ISO TC184/SC4 Document N198, March 22, 1993.

[6] ISO DIS 10303 – 43, Product Data Representation and Exchange – Part 43, Integrated Resources: Representation Structures, ISO TC184/SC4 Document N195, March 22, 1993.

[7] ISO DIS 10303 – 44, Product Data Representation and Exchange – Part 44, Integrated Resources: Product Structure Configuration, ISO TC184/SC4 Document N194, March 22, 1993.

[8] ISO DIS 10303 – 46, Product Data Representation and Exchange – Part 46, Integrated Resources: Visual Presentation, ISO TC184/SC4 Document N190, February 15, 1993.

[9] ISO/CDC 10303-202 Industrial automation systems and integration -- Product data representation and exchange -- Part 202: Application protocol: Associative draughting, International Organization for Standardization, February 2, 1993.

[10] Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text, MIL-M-28001, July 20, 1990

[11] Requirements for an Application Protocol Development Environment, Allison Barnard Feeney, Stephen Nowland Clark, James E. Fowler, NISTIR #5197, May 27, 1993.

[12] The SGML Handbook, Charles F. Goldfarb, Oxford University Press, New York, 1990,

[13] Supplementary directives for the drafting and presentation of ISO 10303, ISO TC184/SC4 Editing N-20, March 17, 1993