

SEMI E120.1-0705

XML SCHEMA FOR THE COMMON EQUIPMENT MODEL (CEM)

This standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on April 7, 2005. It was available at www.semi.org June 2005 and on CD-ROM in July 2005. Originally published in 2004.

Table of Contents

1 Purpose.....	2
2 Scope.....	2
3 Referenced Standards and Documents.....	2
3.1 SEMI Standards.....	2
3.2 OMG Standards.....	2
3.3 W3C Standards.....	2
4 Terminology.....	3
4.1 Abbreviations and Acronyms.....	3
4.2 Definitions and Acronyms.....	3
5 Conventions.....	3
5.3 Translating UML to XML.....	3
5.7 Translation Table Column Header Description.....	3
5.8 Documenting The XML With Diagrams.....	4
6 CEM Class Mapping to XML.....	6
6.2 Abstract Classes.....	7
6.2.2 Nameable => NameableType.....	7
6.2.3 EquipmentElement => EquipmentElementType.....	7
6.2.4 ExecutionElement => ExecutionElementType.....	8
6.2.5 Extension => ExtensionType.....	9
6.3 Concrete Classes.....	9
6.3.5 Equipment => EquipmentType.....	10
6.3.6 Module => ModuleType.....	12
6.3.7 Subsystem => SubsystemType.....	13
6.3.8 IODevice => IODeviceType.....	14
6.3.9 MaterialLocation => MaterialLocationType.....	15
6.3.10 SoftwareModule => SoftwareModuleType.....	15
6.4 Added XML Constructs.....	16
6.4.2 XML simpleTypes.....	16
6.4.3 Added XML complexTypes.....	16
6.4.4 Equipment Element.....	17
7 XML Schema.....	17
8 Related Documents.....	17

List of Figures

Figure 1 XML Example Diagram.....	5
Figure 2 XML for Sample.....	6
Figure 3 - NameableType.....	7
Figure 4 - EquipmentElementType.....	8
Figure 5 - ExecutionElementType.....	8
Figure 6 - ExtensionType.....	9
Figure 7 - Example Of The XML Structure For Shared Components.....	10
Figure 8 - EquipmentType.....	11
Figure 9 - ModuleType.....	13
Figure 10 - SubsystemType.....	14
Figure 11 - IODeviceType.....	15
Figure 12 - MaterialLocationType.....	15
Figure 13 - SoftwareModuleType.....	16
Figure 14 - Equipment Element.....	17

List of Tables

Table 1 - Example Translation Table	4
Table 2 - XML Diagram Symbols.....	4
Table 3 - Translation Table For Nameable.....	7
Table 4 - Translation Table For EquipmentElement.....	7
Table 5 - Translation Table For ExecutionElement.....	8
Table 6 - Translation Table For Extension	9
Table 7 - Translation Table For Equipment.....	10
Table 8 - Translation Table For Module Class.....	12
Table 9 - Translation Table For Subsystem	14
Table 10 - Translation Table For IODevice	14
Table 11 - Translation Table For MaterialLocation.....	15
Table 12 - Translation Table For SoftwareModule	15
Table 13 - simpleTypes In The CEM Schema	16
Table 14 - complexTypes In The CEM Schema	16

1 Purpose

1.1 The purpose of this specification is to provide an XML schema that corresponds to the UML model for equipment defined by SEMI E120. This schema is available to equipment communication standards that want to use CEM-defined equipment structural information in XML formatted equipment communication.

2 Scope

2.1 The scope of this document is the faithful representation of the CEM model in an XML schema. It will not add new domain information or concepts to the model. The only additions made are those needed to render a useful XML schema.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Referenced Standards and Documents

3.1 SEMI Standards

SEMI E120 — Provisional Specification for the Common Equipment Model (CEM)

SEMI E121 — Guide for Style & Usage of XML for Semiconductor Manufacturing Applications

3.2 OMG Standards

Unified Modeling Language (UML) Specification¹, Version 1.4, OMG Specification 01-09-67

3.3 W3C Standards

Extensible Markup Language (XML) 1.0 (Second Edition)² — W3C, 6 October 2000

Namespaces in XML³ — W3C, 14 January 1999

XML Schema Part 0: Primer⁴ — W3C, 2 May 2001

XML Schema Part 1: Structures⁵ — W3C, 2 May 2001

XML Schema Part 2: Datatypes⁶ — W3C, 2 May 2001

¹ Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, U.S.A., Ph: 1-781-444 0404, Fax: 1-781-444 0320, http://www.omg.org/technology/documents/modeling_spec_catalog.htm

² World Wide Web Consortium (W3C), <http://www.w3.org/TR/2000/REC-xml-20001006/>

³ World Wide Web Consortium (W3C), <http://www.w3.org/TR/1999/REC-xml-names-19990114/>

⁴ World Wide Web Consortium (W3C), <http://www.w3.org/TR/xmlschema-0/>

⁵ World Wide Web Consortium (W3C), <http://www.w3.org/TR/xmlschema-1/>

⁶ World Wide Web Consortium (W3C), <http://www.w3.org/TR/xmlschema-2/>

XML Path Language (XPath)⁷ — W3C, 16 November 1999

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

4 Terminology

4.1 Abbreviations and Acronyms

4.1.1 *UML* — Unified Modeling Language

4.1.2 *W3C* — World Wide Web Consortium

4.1.3 *XML* — eXtensible Markup Language

4.2 Definitions and Acronyms

4.2.1 *UML (Unified Modeling Language)* — a notation for representing object-oriented designs and views created by Booch, Rumbaugh, and Jacobson in order to merge their three popular notations plus aspects of other existing notations into a single object-oriented notation intended to be usable by all.

4.2.2 *XML (eXtensible Markup Language)* — a markup language used for representing data rich with context and content in documents and in communications. XML is an extension of SGML, a document-oriented markup language. It was created by W3C for use on the Internet. XML can represent object-oriented structures.

5 Conventions

5.1 This section discusses the conventions used in this specification for translating UML to XML and for documenting the XML. These conventions are heavily influenced by the SEMI E121.

5.2 The reader is expected to have a working knowledge of the UML, XML, and Schema specifications (See ¶3.2 and ¶3.3). This document does not provide tutorial information on these subjects.

5.3 Translating UML to XML

5.4 This document follows the guidelines for XML as outlined in SEMI E121.

5.5 Some of the key guidelines followed in this specification are summarized here:

- Attributes of a class are generally represented as elements such that they can be easily extended by other applications.
- Inheritance is modeled as element extension.
- Compositions always turn into contained elements (or arrays if multiples are allowed).
- Associations and Aggregations are modeled as contained elements or arrays wherever possible to simplify the instance model.

5.6 The translation of a UML class to XML is documented using a table format illustrated Table 1. Note that SEMI E120 does not define services, so only attribute and association representations are addressed here. Any superclass information is addressed in the text and the diagrams.

5.7 Translation Table Column Header Description

- *Attribute or Role Name* — If an attribute, the name of the attribute is placed here. If an association (including aggregation or composition), the role name from the UML diagram is placed here. Compositions are often not assigned roll names. In that case “none” is placed here.
- *UML Name/Type* — If an attribute, the data type of the UML attribute is place here. If an association, the type of association is placed here. The possible types are “Composition”, “Aggregation”, or the basic “Association”. UML defines these three types of association.
- *XML Element or Attribute* — Lists the type of XML construct used to represent the UML attribute or association.

⁷ World Wide Web Consortium (W3C), <http://www.w3.org/TR/xpath/>

- *XML Name/Type* — Provides the name and data type of the resulting XML construct. The type may be a built-in type (for example, xs:string), or a named type defined within the XML schema.

Table 1 Example Translation Table

<i>Attribute or Role Name</i>	<i>UML Name/Type</i>	<i>XML Element or Attribute</i>	<i>XML Name/Type</i>
friend	association	element	Friend: HumanReferenceArray
employees	aggregation	element	Employees: HumanArray
family	composition	element	Family: HumanArray
name	string	element	Name: xs:string

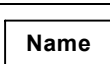
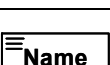
5.8 Documenting The XML With Diagrams

5.8.1 This document provides graphical representations of the included XML. Although no standard graphical notation for XML could be found, various XML tools have their own notation. This document will use the notation provided by XMLSPY® from Altova Corporation⁸. There is no requirement for use of XMLSPY® in order to comply with this specification. Figure 1 shows a sample XML diagram that will be used to provide a basis for explanation of the XML graphical notation used in the rest of the document.

5.8.2 In the diagram, rectangular boxes represent XML elements. Ownership or containment is read from right to left in the diagrams. In the sample diagram, ParentType contains Child1, Child2, Child3, and Child4. In turn, Child2 contains Child2a, Child2b, and Child2c. The additional symbols (8-sided boxes) represent sequences or choices. See Table 2 for an explanation of these symbols.

5.8.3 Please note that “element”, “sequence”, “all”, and “choice” are XML terms (see the W3C standards referenced in ¶3.3). The reader is expected to be familiar with XML.

Table 2 Altova XMLSPY Schema Diagram Symbols

	Denotes a required ordered sequence of the right hand elements with a cardinality of one for each element. (sequence)
	Denotes an optional ordered sequence of the right hand elements with a cardinality of one for each element. (sequence)
	Denotes a required, but unordered, sequence of the right hand elements with a cardinality of one for each element. (all)
	Denotes a required choice of the right hand elements. Exactly one or two of the right hand elements must be present. (choice)
	Denotes an element.
	Denotes an element that contains parsed character data.

5.8.4 A graphic using a solid line is a required element; using a dashed line represents an optional element. Numbers or ranges in the lower right hand corner represent cardinality. The default cardinality is one.

5.8.5 To simplify a diagram or help focus on a particular aspect, detail may be hidden. The 8-sided symbols have a small square on the right end. If a minus sign “-” is in the box, then all detail is shown. If the box contains a plus sign “+”, then all detail to the right of that symbol is hidden. The example has no hidden detail.

⁸ All images/graphics were created using Altova’s XMLSPY®. Copyright 2003 Altova GmbH and reprinted with permission of Altova.

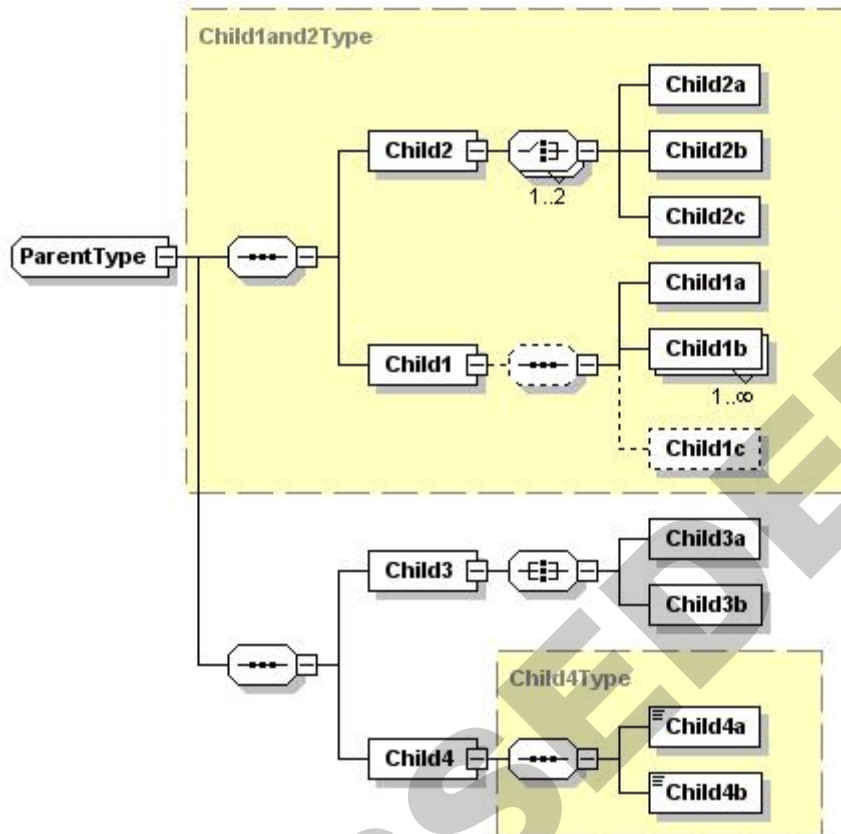


Figure 1
XML Example Diagram

5.8.6 The yellow (or gray if printed in monochrome) boxes indicate the use of other defined types. So, Child4 is of type “Child4Type”. Child4Type defines Child4a and Child4b. This detail may be hidden in the diagram. Object oriented inheritance is typically represented in XML as type extension. In Figure 1, ParentType extends Child1and2Type by adding a sequence that includes Child3 and Child4.

5.8.7 Reading Figure 1 would yield the following additional information:

1. Child1and2Type is an ordered sequence of two items: Child2 and Child1.
2. Child1 contains an optional ordered sequence of Child1a, one or more Child1b, and (optionally) Child1c.
3. Child2 contains a choice of one or two of the following: Child2a, Child2b, and Child2c.
4. Child3 contains an unordered sequence of Child3a and Child3b.

5.8.8 XML Sample

5.8.8.1 The sample XML for the example shown in Figure 1, is presented below. Refer to the XML documentation referenced in ¶3.3 for a complete description of the syntax and semantics of XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:complexType name="ParentType" abstract="true">
    <xs:complexContent>
      <xs:extension base="Childland2Type">
        <xs:sequence>
          <xs:element name="Child3">
            <xs:complexType>
              <xs:all>
                <xs:element name="Child3a"/>
                <xs:element name="Child3b"/>
              </xs:all>
            </xs:complexType>
          </xs:element>
          <xs:element name="Child4" type="Child4Type" nillable="false"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="Child4Type" abstract="true">
    <xs:sequence>
      <xs:element name="Child4a" type="xs:integer" nillable="false"/>
      <xs:element name="Child4b" type="xs:string" nillable="false"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Childland2Type" abstract="true">
    <xs:sequence>
      <xs:element name="Child2">
        <xs:complexType>
          <xs:choice maxOccurs="2">
            <xs:element name="Child2a"/>
            <xs:element name="Child2b"/>
            <xs:element name="Child2c"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
      <xs:element name="Child1">
        <xs:complexType>
          <xs:sequence minOccurs="0">
            <xs:element name="Child1a"/>
            <xs:element name="Child1b" maxOccurs="unbounded"/>
            <xs:element name="Child1c" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Parent" type="ParentType" final="#all"/>
</xs:schema>
```

Figure 2
XML for Sample

6 CEM Class Mapping to XML

6.1 This section describes how the CEM classes from SEMI E120 were mapped to XML. The descriptions in this section are provided to support and explain the XML schema document.