SEMI E139.1-0709
XML SCHEMA FOR THE RaP PDE

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 May 13, 2009. It was available at www.semi.org in June 2009 and on CD-ROM in July 2009. Originally published July 2005; previously published November 2008.

SUPERSEDED

Table of Contents

1 Purpose 2
2 Scope 2
3 Limitations 2
4 Referenced Standards and Documents 2
5 Terminology 2
6 Conventions 3
7 PDE Schema Definition 6
8 Manifest Schema Definition 14
9 Related Documents 15

List of Figures

Figure 1 XML Example Diagram 5
Figure 2 XML for Sample 6
Figure 3 PDE 7
Figure 4 PDEheader 8
Figure 5 ExecutionTarget 9
Figure 6 ReferencedPDE 9
Figure 7 AntecedentData 10
Figure 8 PDEparameter 11
Figure 9 PDEbody 11
Figure 10 PDEbodyReference 12
Figure 11 Manifest 15

List of Tables

Table 1 Example Translation Table 3
Table 2 Altova XMLSPY® Schema Diagram Symbols 4
Table 3 Translation Table for PDE Class 7
Table 4 Translation Table for PDEheader Class 7
Table 5 Translation Table for ExecutionTarget 9
Table 6 Translation Table for ReferencedPDE 9
Table 7 Translation Table for AntecedentData 10
Table 8 Translation Table for PDEparameter 10
Table 9 Translation Table for PDEbody 11
Table 10 Translation Table for PDEbodyReference 12
Table 11 simpleTypes in the PDE Schema 12
Table 12 complexTypes in the PDE Schema 12
Table 13 Translation Table for Manifest Class 14
Table 14 simpleTypes in the Manifest Schema 15

Documents Document Sharing Hub
Full document details: SEMI E139.1-0709

1 Purpose

1.1 The purpose of this document is to define an XML-based format for transferring recipe components. This includes an XML schema that corresponds to the UML model for the RaP Process Definition Element (*PDE*) as defined by SEMI E139. In addition, an XML schema is defined for the Manifest that must be included in the *TransferContainer* with the *PDEs* during transfer.

2 Scope

2.1 The scope of this document is the faithful representation of the *PDE* and Manifest definition from SEMI E139 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 Limitations

3.1 None.

4 Referenced Standards and Documents

4.1 *SEMI Standard*

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

4.2 *OMG Standard*[1]

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

4.3 *W3C Standards*[2]

Canonical XML Version 1.0 — W3C, 15 March 2001; http://www.w3.org/TR/xml-c14n

Extensible Markup Language (XML) 1.0 (Second Edition) — W3C, 6 October 2000; http://www.w3.org/TR

Namespaces in XML — W3C, 14 January 1999; http://www.w3.org/TR/1999/REC-xml-names-19990114/

XML Schema Part 0: Primer — W3C, 2 May 2001; http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures — W3C, 2 May 2001; http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes — W3C, 2 May 2001; http://www.w3.org/TR/xmlschema-2/

XML Path Language (Xpath) — W3C, 16 November 1999; http://www.w3.org/TR

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *UML* — Unified Modeling Language

5.1.2 *W3C* — World Wide Web Consortium

5.1.3 *XML* — eXtensible Markup Language

5.2 *Definitions*

5.2.1 *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.

1 Object Management Group, Inc., 250 First Ave. Suite 100, Needham Mass. 02494, Telephone: 781.444.0404; Fax: 781.444.0320; http://www.omg.org/

2 World Wide Web Consortium, Massachusetts Institute of Technology (MIT), Computer Science and Artificial Intelligence Laboratory (CSAIL), 32 Vassar Street Room 32-G515, Cambridge, MA 02139, USA, Telephone: 617.253.2613, Fax: 617.258.5999; http://www.w3.org

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E139.1-0709

5.2.2 *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.

6 Conventions

6.1 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 — Guide for Style & Usage of XML for Semiconductor Manufacturing Applications.

6.2 The reader is expected to have a working knowledge of the UML, XML, and Schema specifications (see ¶ 4.2 and ¶ 4.3). This document does not provide tutorial information on these subjects.

6.3 *Translating UML to XML*

6.3.1 This document follows the guidelines for XML as outlined in SEMI E121 — Guide for Style & Usage of XML for Semiconductor Manufacturing Applications.

6.3.2 Some of the key guidelines 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 contained collections if multiples are allowed).
- Associations and Aggregations are modeled as contained elements or arrays wherever possible to simplify the instance model.

6.3.3 The translation of a UML class to XML is documented using a table format illustrated by Table 1. Note that the PDE model in SEMI E139 does not include services, so only attribute and association representations are addressed here. Any superclass information is addressed in the text and the diagrams.

6.4 *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 role names. In that case "none" is placed here.
- *UML Type* — If an attribute, the data type of the UML attribute is placed 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.
- *XML Name/Type* — Provides the name and data type of the resulting XML construct. The type may be a built-in type (e.g., xs:string), or a named type defined within the XML schema.

Table 1 Example Translation Table

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

6.5 *Documenting the XML with Diagrams*

6.5.1 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

SUPERSEDED

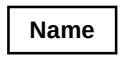
Documents Document Sharing Hub
Full document details: SEMI E139.1-0709

provided by XMLSPY® from Altova Corporation.[3] 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.

6.5.2 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, Child 3, and Child 4. 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.

6.5.3 Please note that "element," "sequence," "all," and "choice" are XML terms (see the W3C standards referenced in ¶ 4.3).

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. Optional element if outline is dashed line.
	Denotes an element that contains parsed character data.

6.5.4 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.

6.5.5 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.

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

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E139.1-0709

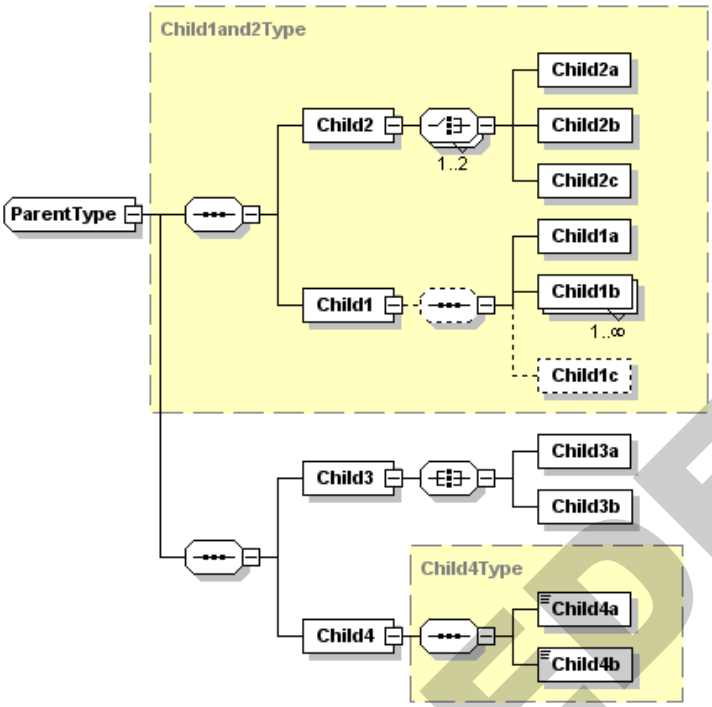


Figure 1
XML Example Diagram

6.5.6 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.

6.5.6.1 Reading Figure 1 would yield the following additional information:

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

6.5.7 *XML Schema Sample*

6.5.7.1 The sample XML for the example shown in Figure 1 is presented below. Refer to the XML documentation referenced in ¶ 4.3 for a complete description of the syntax and semantics of XML schemas.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E139.1-0709