

SEMI E139.3-0709

レシピとパラメータ管理に関する XML/SOAP バインディング

本スタンダードは、global Information & Control Committee で技術的に承認されている。現版は 2009 年 5 月 13 日、global Audits and Reviews Subcommittee にて発行が承認された。2009 年 6 月に www.semi.org で、そして 2009 年 7 月に CD-ROM で入手可能となる。初版は、2008 年 11 月に発行された。

1 目的

1.1 本仕様の目的は、レシピとパラメータ管理に関する SEMI E139 仕様の XML フォーマットおよび SOAP 1.1 プロトコルへの実装マッピングを提供することである。

2 範囲

2.1 本仕様の範囲は、XML スキーマおよび対応する WSDL ポートタイプとバインディングにおける RaP サービスと関連するパラメータの表現である。以下を規定する。

- SEMI E139 に定義されるサービスをサポートするために使用される XML スキーマのデータタイプ
- SEMI E139 に規定されるインターフェイスによって定義されたオペレーションをサポートするために仕様される WSDL ポートタイプとバインディング定義。

2.2 本仕様は、SEMI E139 の仕様に対して新たなドメイン情報や概念を追加するものではない。追加するものは、有益な WSDL や XML スキーマを提供するために必要なものだけである。

2.3 本仕様は、SEMI E132 装置クライアントの認証および権限付与のための仕様に基づいて構築される。本仕様は、SEMI E139 の仕様に対して SEMI E132 の概念を適用するための要求事項を定義する。

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.

注意: 本スタンダードはその使用に関連したすべての安全問題を取り扱うことを意図していない。このスタンダードの利用者は、その責任において、適切な安全および健康上の作業方法を確認し、また使用前に規制上の制限への適用性を判断するものである。

3 制限

3.1 本仕様は、XML/SOAP に基づく他の通信がない場合における RaP サービスの実装を取り扱う。本仕様は、装置上で E132 のサービスを共有する可能性のある他の機能と互換性を持つことを目的としている。しかしながら、万一、類似した他の通信インターフェイス（例えば E134 サービス）が存在した場合の潜在的な相互作用および対立を特定したり、取り扱うものではない。

4 参照スタンダードおよび文書

4.1 SEMI スタンダード

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

SEMI E128-0706 — Provisional Specification for XML Message Structures

SEMI E132 — Specification for Equipment Client Authentication and Authorization

SEMI E134 — Specification for Data Collection Management

SEMI E138 — XML Semiconductor Common Components

4.2 W3C Standards¹

Extensible Markup Language (XML) 1.0 (Second Edition) — W3C, 6 October 2000 (<http://www.w3.org/TR/2000/REC-xml-20001006/>)

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/>)

Web Services Description Language (WSDL) 1.1 — W3C Note (<http://www.w3.org/TR/wsdl/>)

Simple Object Access Protocol (SOAP) 1.1 — W3C Note (<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>)

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

注意： 別途指示のない場合、引用されたすべての文書は最新の出版バージョンである。

5 用語

5.1 略語と頭字語

5.1.1 SOAP(*Simple Object Access Protocol*) — シンプルオブジェクトアクセスプロトコル

5.1.2 UML(*Unified Modeling Language*) — 統一モデリング言語

5.1.3 W3C(*World Wide Web Consortium*) — ワールドワイドウェブコンソーシアム

5.1.4 WSDL(*Web Service Description Language*) — Web Service Description Language

5.1.5 XML(*eXtensible Markup Language*) — 拡張マークアップ言語

5.1.6 SSL(*Secure Sockets Layer*) — Secure Sockets Layer

5.1.7 RaP(*Recipe and Parameter Management*) — レシピとパラメータの管理— 機能定義は SEMI E139 を参照。

5.2 定義

5.2.1 ポートタイプ(*Port Type*) — ポートタイプは抽象的な操作と関連する抽象的なメッセージをまとめて名前を付けたものである。

5.2.2 WSDL バインディング(*WSDL Binding*) — ポートタイプ定義がどのようにして具体的なネットワークプロトコルおよびメッセージフォーマット(例, SOAP)にバインドされるかの定義。

6 規定

6.1 UML から XML スキーマへの変換

6.1.1 読者は、XML, WSDL, SOAP, スキーマ仕様の実務的な知識を有していることを前提とする(¶4.2 を参照)。本文書では、これらの題材について解説は提供しない。

¹ 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>

6.1.2 UML の XML へのマッピング規約作成に関して XML スタイルと利用のガイド (SEMI E121) が参照された。

6.1.3 SEMI E139 の UML の XML へのマッピングについて重要な規則をここで要約する。

- SEMI E139 の UML クラス属性は XML 要素として表される。
- コンポジション関連は含まれる要素にマッピングされる。

6.1.4 UML クラスから XML スキーマの型への変換は、表 1 に示すように表形式を使用して文書化される。UML クラスで定義した操作は、WSDL での操作に変換される。これらのマッピングは概念的な要求を実装形態に関連付ける。したがって、要求に対する拡張と見なされる。

6.1.5 変換表の見出し行の説明

- 属性名／ロール名 — 属性の場合は、その属性名をここに記載する。関連 (集約やコンポジションも含む) の場合は、UML 図のロール名をここに記載する。コンポジションは多くの場合、ロール名を明記しない。その場合は「なし」と記載する。
- UML 名／タイプ — 属性の場合は、UML 属性のデータタイプをここに記載する。関連の場合は、関連のタイプをここに記載する。関連のタイプの候補は「コンポジション」、「集約」、一般的な「関連」である。UML では、これらの 3 つのタイプの関連を定義する。
- XML 要素／属性 — UML 属性や関連を表現するために使われる XML による構成方法のタイプを記載する。
- XML 名／タイプ — 結果的に得られる XML 構成の名前とデータタイプを記載する。タイプ名は、組み込みタイプ (例えば、xsd:string), あるいは XML スキーマで名前をつけて定義するタイプのどちらかである。

表 1 変換表の例

属性名／ロール名	フォーマット／タイプ	XML 要素／タイプ	XML 名／タイプ
Friend	関連	要素	Friend: HumanReferenceArray
employees	集約	要素	Employees: HumanArray
Family	コンポジション	要素	Family: HumanArray
Name	文字列	要素	Name: xs:string

6.1.6 XML スキーマへの操作の変換

6.1.6.1 UML インターフェイスクラスで定義される操作の XML スキーマタイプへの変換は、表 2 に示す表形式を使用して記録される。対象となる操作の取り得るそれぞれの入力／出力引数毎に 1 行を充てて示すものとする。

表 2 操作の入力／出力引数の変換表の例

引数名	フォーマット	XML 要素／属性	XML 名／タイプ
<抽象規定での引数名>	<抽象規定での引数のフォーマット>	<引数が要素か属性かの何れでモデル化されているかを示す>	<属性／要素の名前>

6.1.7 XML スキーマと WSDL ファイルの文書化

6.1.7.1 本仕様の中核部分である XML スキーマと WSDL ファイルが本書に添付される。この各文書は、表 3 に示す表形式で説明する。

表 3 スキーマ/WSDL 文書説明表の例

ファイル名	<.xsd または.wSDL ファイル名>
ターゲットネームスペース	<スキーマや WSDL ファイル内で定義されるターゲットネームスペース>
インポート元/ 参照元のネームスペース	<スキーマや WSDL ファイルによってインポートあるいは使用されるネームスペース>
説明	<スキーマや WSDL ファイルの目的/機能の簡単な説明>

6.2 図によるXMLスキーマの説明²

6.2.1 本文書は、記載する XML スキーマのデータタイプ定義と要素定義およびそれらの関係を図表示で説明する。これらは読者にスキーマの理解を助けるよう掲載されている。これらの図は XML ファイル内に含まれている要求の図解であり、追加要求と解釈すべきものではない。

6.2.2 XML スキーマの標準的な図示方法はないが、XML ツールが独自の表記を行っている。本文書では、Altova Corporation の XMLSpy が用いる表記法を使用する。図 1 で示すものは XML スキーマ図の例であるが、本文書でこれ以降使用するスキーマの図示による基本事項を説明するために使用される。

6.2.3 本図では、矩形状の箱が XML 要素の定義を示す。オーナーとなる要素を左側におき、そのオーナー要素に格納される要素を右側におく。図の例では、ParentType が Child1, Child2, Child 3, Child 4 を格納している。また、Child2 は、Child2a, Child2b, Child2c を格納している。意味を添えられた記号(8 角形の箱)は、sequence や choice を示す。これらの記号の説明については、表 4 を参照すること。

表 4 Altova XMLSPY®スキーマ図の記号

	すぐ右側に並んだ要素、原則個数各 1 個の順序付きの要素列(sequence)で、要素の存在が必須と想定されるもの。
	すぐ右側に並んだ要素、原則個数各 1 個の、順序付きの要素列(sequence)で、要素の有無は任意。
	すぐ右側に並んだ要素、個数各 1 個、何れも必須の順序は任意の要素列(all)。
	すぐ右側に並んだ要素の選択肢を示し、存在は必須である。(左図の場合)1 つ以上 2 つ以下の右側の要素が存在しなければならない(choice)。
	要素を示す。輪郭が破線の場合、選択肢を示す。
	左上隅にある表記は、構文解析済み文字データを示す。
	左下隅にある表記は、他の箇所で定義されているグループや要素への参照を示す。

6.2.4 実線を使用した図形は必須要素を示し、破線は任意要素を示す。右下隅の数字は、個数を示す。指定のない場合の個数は 1 である。

6.2.5 図の単純化、あるいは特定の観点を強調できるように、詳細を明らかにしないことがある。八角形の記号の右端には小さな正方形が付く。その箱内にマイナスの記号「-」がある場合は、詳細はすべて表示されている。箱内にプラスの記号「+」がある場合は、その記号の右側の詳細部分は全く表示されない。下図の例では、詳細の表示がある。

² 画像/図は、すべて Altova' s XMLSPY®を使用して作成した。著作権所有 2003 年 Altova GmbH, Altova 社の許可を得て再印刷。

6.2.6 黄色(または白黒の場合は灰色)の箱は、他の定義された型を使用することを示す。従って、Child4 は、型「Child4Type」であり、Child4Type は Child4a と Child4b を定義するが、この詳細は図に示さなくても良い。オブジェクト指向方法論の継承は、ここでは普通、XML では型の extension 要素として表現する。図 1 では、ParentType は、Child3 と Child4 を含む sequence 要素を追加することによって Child1 と 2Type を拡張する。

6.2.7 図 1 から、さらに次の情報が得られる。

1. Child1and2Type は、2 つの項目 Child2 と Child1 の順序付き要素列である。
2. Child1 は、Child1a、1 つまたは複数の Child1b、(有無は任意の)Child1c の順序付きの要素列で、有無は任意の列を格納する。
3. Child2 は、次の Child2a、Child2b、Child2c の各々が 1 つまたは 2 つある中から何れかを選んで格納する。
4. Child3 は、Child3a と Child3b からなる順序が任意の要素列を格納する。

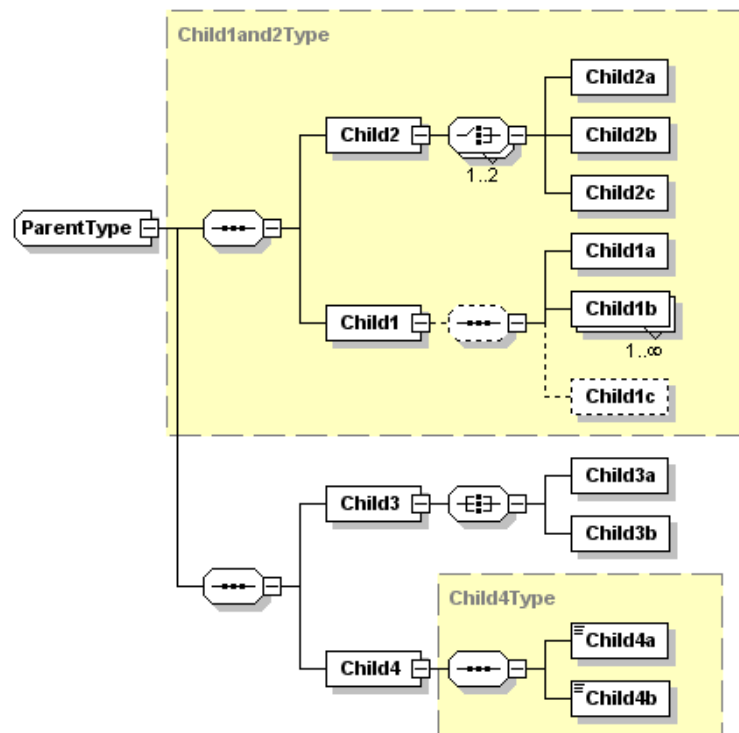


図 1
XML スキーマ図の例

6.2.8 XML スキーマの例

6.2.8.1 図 1 に示される XML スキーマの例は以下の図 2 に表されている。XML スキーマの文法および意味の完全な説明については、¶4.2 で引用した XML 文書を参照すること。

```
<?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="Child1and2Type">
        <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="Child1and2Type" 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>
```

図 2
例のための XML

6.2.9 UML から WSDL への変換

6.2.9.1 一般に、抽象規定内で定義される UML インターフェイスクラスは、portType の定義として WSDL に変換され、portType の各定義には対応する WSDL バインディングの定義を置く。次の表では、任意の UML インターフェイスクラスについて WSDL のポート型(port type)とバインディングの定義の文書化に使用される変換を示す。

表 5 インターフェイス WSDL ポートタイプ表の例

クラス名	<抽象規定での UML インターフェイス名>
WSDL ポートタイプ名	<WSDL 内で使用されるポートタイプ名>
SEMI E139 での操作 → WSDL による操作	<ML インターフェイスでの操作名> → <WSDL ポートタイプの操作名>

表 6 インターフェイス WSDL バインディング表の例

SEMI E139 クラス名	<抽象規定での UML インターフェイス名>
WSDL バインディング名	<WSDL 内で使用されるバインディング名>
SOAP バインディングスタイル	<RPC または文書>
SOAP トランスポート	<トランスポートを特定する URI>

7 概要

7.1 本仕様をより良く理解するために、本セクションでは要求事項の詳細へと移る前に本仕様の概要を取り扱う。この概要のセクションではいかなる要求事項も規定しない。ここに含まれる記述は、以降の要求事項のセクションで規定される特定の要求を前提とする場合がある。

7.2 本仕様は、SEMI E132 を基に構築される。E132 は、クライアントからサービスプロバイダーへの安全な接続を行うためのメカニズムを提供する。本仕様の実装は E125 装置自己記述の仕様のような E132 に準拠するほかのスタンダードと共存することがありえる。したがって、E132 のポートは適切な状況において共有することができる。実装には個別のポートを使用することを選ぶ選択肢がある。

7.2.1 本書は E139 の元のフォームから XML/SOAP メッセージで使用可能なフォームへと変換されたメッセージ定義の仕様に焦点を当てる。

7.2.2 本仕様の目的上、SEMI E132 で「装置」と言及される箇所は「サービスプロバイダー」と解釈されなければならない。

7.2.3 特権は E139 のサービスに対して定義され、かつ SEMI E132 のメカニズムを使用して実施される。個別の特権は各々の RaP サービス(requestToSendを除いて)に対して定義される。

7.3 XML/SOAP の通信アプローチは、SECS のような他の通信プロトコルと共存するように意図されている。これにより変換経路(Transition Path)が提供されることが期待される。例えば、従来の SECS での通信リンクの場合、XML/SOAP に準拠する通信の追加によって、工場が SECS のホストアプリケーションを変更する前に、オフツールエディタが XML/SOAP のメッセージングを介してツールに接続できるようになる。

7.4 SEMI E139 は処理中に起こりうるエラー報告のためのメッセージパラメータを定義する。しかしながら、新たなプロトコルとともに起こりうるエラーがさらに生じる。プロトコルのエラーが個別で取り扱われるように RaP サービスに対する応答メッセージが構造化されている。メッセージの内容は、E139 で定義された応答もしくは個別のプロトコルエラー構造のどちらかで返答可能な XML の選択(choice)構想として設計される。

7.5 本仕様における TransferContainer の取り扱い是非常に単純である。TransferContainer は base64 符号付きバイナリメッセージ要素として SOAP メッセージに含まれる。

7.5.1 実装者は、大容量の TransferContainer を含むメッセージのメモリのインスタンス化を避けるために、いくつかの予防措置を取らなければならないことに注意するべきである。RaPnode は、アプリケーションに利用可能なメモリ量に基づいて、受信されるメッセージの実質的なサイズに制限を持つ場合がある。

8 要求事項

8.1 本仕様への準拠には、本仕様内で提示されるすべての要求事項を満たすことが求められる。

8.2 本仕様は SEMI E139 をそのまま前提にして確立されるものである。本仕様の実装は、SEMI E139 の要求事項に準拠しなければならない。

8.3 1 つの RaPnode は 1 つ以上の定義済みプロトコルを介して RaP サービスを提供することがある。例えば、SECS (SEMI E139.2 に基づく) および XML/SOAP (本仕様に基づく) の両方ともが提供される場合がある。

8.3.1 RaPnode が SEMI E139.2 と E139.3 の両方に準拠する場合、RaP サービスは両方のプロトコルにおいて同時に利用可能でなければならない。

8.4 転送目的に対する TransferContainer の形式は、すべての関連要求および制限を含めた本仕様の公開日時点での E139.2 の現行バージョンで定義されたものと同じ ZIP 構造でなければならない。

8.5 SEMI E132 基準

8.5.1 本仕様の実装は、SEMI E132 (装置クライアントの認証および権限付与のための仕様) に準拠しなければならない。

8.5.1.1 SEMI E132 は複数のクライアントに対してサービス供給者へのアクセスを同時に提供することに注意すること。FICS が RaP と接続するとき 1 つ以上のレシピエディタを装置に同時に接続しなければならない場合があるため、RaP にとって必要なことである。

8.5.2 RaP の通信セッションは一方方向でなければならない。つまり、RaP サービス要求はクライアントによってのみ開始できる。

8.5.2.1 通信する両者が RaPnode の場合 (つまり、両者が RaP サービスを提供する場合)、個別の接続は反対方向で確立できることに注意すること。

8.5.3 SEMI E132 は、EnhancedEstablishSession サービスを提供する。このサービスによって、クライアントはこのサービスが規定する機能のみを提供するセッションの確立ができるようになる。RaP サービスの集合は、E132 のセッション機能の一つの構成要素とならなければならない。

8.5.3.1 この RaP サービス機能の名前は E132 によって「SEMI-E139-MMY」の形式となるように規定され、MMY は E139 スタンドアロンの公開された月年をあらわすことに注意すること。例えば、2008 年 7 月公開の RaP は「SEMI-E139-0708」と名づけられた機能となる。

8.6 XML スキーマおよび WSDL ファイル

8.6.1 SEMI E139 の各インターフェイスの定義は、WSDL の portType とバインディングの定義にマッピングされる。各々の WSDL の portType の定義は、SEMI E139 の記載に即してインターフェイスとその操作を基に名前が付けられる。portType ごとに WSDL バインディングの定義を使用して、各操作に対する SOAP 1.1 のエンベロープの内容を規定し、対応する XML のエンコーディングスタイルおよび HTTP ヘッダの使用方法を定義する。SEMI E139.3 のすべての WSDL インターフェイスでは、文書/文字によるエンコーディング形式を使用し、グローバル要素の定義を使用して、SOAP ヘッダとボディの完全形として XML スキーマファイル内で定義する。

8.6.1.1 本仕様の実装は、以下の表 7 で定義されるスキーマファイルに準拠しなければならない (‘ファイル名’を参照)。スキーマは SEMI E139 で定義される RaPnode の 3 タイプすべてに適用しなければならない。この RaP サービスのスキーマによって定義またはインポートされたネームスペースに関する追加情報を表 7 に示す。