

SEMI E132.1-0310

装置クライアントの認証および権限付与(ECA)のための SOAP バインディングの仕様

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 12 月 16 日, global Audits and Reviews Subcommittee にて発行が承認された。2010 年 2 月に www.semi.org で入手可能となる。初版は 2005 年 3 月発行, 前版は 2008 年 11 月に発行された。

目次

- 1 目的
- 2 範囲
- 3 制限
- 4 参照スタンダードおよび文書
 - 4.1 SEMI スタンダード
 - 4.2 ISO スタンダード
 - 4.3 NIST スタンダード
 - 4.4 OMG スタンダード
 - 4.5 W3C スタンダード
 - 4.6 その他のスタンダード
- 5 用語
 - 5.1 略語および頭字語
 - 5.2 定義
- 6 表記法
 - 6.1 UML の XML スキーマへの変換
 - 6.2 XML スキーマと WSDL ファイルの文書化
 - 6.3 図による XML スキーマの文書化
 - 6.4 XML スキーマの例
 - 6.5 UML から WSDL への変換
- 7 SEMI E132 の認証の SSL へのマッピング
 - 7.1 SSL の概要
 - 7.2 資格認定およびデジタル証明書
 - 7.3 SSL 認証の有効化/無効化
- 8 SEMI E132 の UML の XML スキーマおよび WSDL へのマッピング
 - 8.1 WSDL の構成
 - 8.2 SecurityAdmin インターフェイス
 - 8.3 SessionManager インターフェイス
 - 8.4 SessionClient インターフェイス
- 9 メッセージトランスポート

図リスト

- 図 1 XML スキーマ図の例
- 図 2 例の XML
- 図 3 SSL の基本ハンドシェイクフロー
- 図 4 SEMI E132.1 のセキュリティ資格認定
- 図 5 XML スキーマおよび WSDL ファイルの構成
- 図 6 SEMI E132 の WSDL および XML スキーマファイル
- 図 7 SecurityAdmin インターフェイスの WSDL portType へのマッピング
- 図 8 SecurityAdmin WSDL のサービスの定義例
- 図 9 UnauthorizedOperationType に対する XML スキーマの型
- 図 10 GetDefinedPrivilegesRequest に対応する XML スキーマのエレメント
- 図 11 GetDefinedPrivilegesResponse に対応する XML スキーマのエレメント
- 図 12 DefinedPrivilegesType に対する XML スキーマの型
- 図 13 GetDefinedPrivilegesErrorType XML スキーマの型
- 図 14 GetACLRequest に対応する XML スキーマのエレメント
- 図 15 GetACLResponse に対応する XML スキーマのエレメント
- 図 16 ACLEntrySelectorType XML に対するスキーマの型
- 図 17 PrivilegeAssignment に対する XML スキーマの型
- 図 18 RoleAssignment に対する XML スキーマの型
- 図 19 AddACLEntryRequest に対応する XML スキーマのエレメント
- 図 20 AddACLEntryResponse に対応する XML スキーマのエレメント
- 図 21 UnrecognizedPrivilegeType に対する XML スキーマの型
- 図 22 DeleteACLEntryRequest に対応する XML スキーマのエレメント
- 図 23 DeleteACLEntryResponse に対応する XML スキーマのエレメント
- 図 24 GetActiveSessionsRequest に対応する XML スキーマのエレメント
- 図 25 GetActiveSessionsResponse に対応する XML スキーマのエレメント
- 図 26 ActiveSession に対する XML スキーマの型
- 図 27 SetMaxSessionsRequest に対応する XML スキーマのエレメント
- 図 28 SetMaxSessionsResponse に対応する XML スキーマのエレメント
- 図 29 GetMaxSessionsRequest に対応する XML スキーマのエレメント
- 図 30 GetMaxSessionsResponse に対応する XML スキーマのエレメント
- 図 31 SessionManager WSDL サービスの定義例
- 図 32 EstablishSessionRequest に対応する XML スキーマのエレメント
- 図 33 EstablishSessionResponse に対応する XML スキーマのエレメント
- 図 34 SessionPingRequest に対応する XML スキーマのエレメント
- 図 35 SessionPingResponse に対応する XML スキーマのエレメント
- 図 36 CloseSessionRequest に対する XML スキーマの型
- 図 37 CloseSessionResponse に対応する XML スキーマのエレメント
- 図 38 EnhancedEstablishSessionRequest XML スキーマエレメント
- 図 39 EnhancedEstablishSessionResponse XML スキーマエレメント
- 図 40 SessionClientWSDL サービスの定義例
- 図 41 SessionFrozenNotification に対応する XML スキーマのエレメント
- 図 42 SessionClosedNotification に対応する XML スキーマのエレメント

表リスト

表 1	変換表の例
表 2	操作の入力／出力引数の変換表の例
表 3	スキーマ／WSDL 文書記述表の例
表 4	Altova XMLSPY スキーマ図の記号
表 5	インターフェイス WSDL portType 表の例
表 6	インターフェイス WSDL バインディング表の例
表 7	操作のバインディング表の例
表 8	PortType の操作表の例
表 9	XML スキーマ
表 10	SecurityAdmin の PortType の定義
表 11	SecurityAdmin のバインディングの定義
表 12	SecurityAdmin の portType
表 13	SecurityAdmin のバインディング
表 14	SEMI E132 のエラーコード
表 15	UnauthorizedOperationError クラスの変換表
表 16	GetDefinedPrivileges の出力引数の変換表
表 17	DefinedPrivilege クラスの変換表
表 18	GetDefinedPrivileges のメッセージ
表 19	GetDefinedPrivileges PortType の操作
表 20	GetDefinedPrivileges の操作のバインディング
表 21	GetACL の出力引数の変換表
表 22	PrivilegeAssignment クラスの変換表
表 23	RoleAssignment クラスの変換表
表 24	Subject クラスの変換表
表 25	Privilege クラスの変換表
表 26	GetACL のメッセージ
表 27	GetACL PortType の操作
表 28	GetACL の操作のバインディング
表 29	AddACLEntry の入力引数の変換表
表 30	AddACLEntry の出力引数の変換表
表 31	UnrecognizedPrivilegeError クラスの変換表
表 32	AddACLEntry のメッセージ
表 33	AddACLEntry の PortType の操作
表 34	AddACLEntry の操作のバインディング
表 35	DeleteACLEntry の入力引数の変換表
表 36	DeleteACLEntry の出力引数の変換表
表 37	DeleteACLEntry のメッセージ
表 38	DeleteACLEntry の PortType の操作
表 39	DeleteACLEntry の操作のバインディング
表 40	GetActiveSessions の出力引数の変換表
表 41	ActiveSession クラスの変換表
表 42	HTTPEndPoint クラスの変換表
表 43	GetActiveSessions のメッセージ
表 44	GetActiveSessions の PortType の操作
表 45	GetActiveSessions の操作のバインディング
表 46	SetMaxSessions の入力引数の変換表
表 47	SetMaxSessions の出力引数の変換表
表 48	SetMaxSessions のメッセージ

表 49	SetMaxSessions の PortType の操作
表 50	SetMaxSessions の操作のバインディング
表 51	GetMaxSessions の出力引数の変換表
表 52	GetMaxSessions のメッセージ
表 53	GetMaxSessions の PortType の操作
表 54	GetMaxSessions の操作のバインディング
表 55	XML スキーマ
表 56	SessionManager の PortType の定義
表 57	SessionManager のバインディングの定義
表 58	SessionManagerportType
表 59	SessionManager のバインディング
表 60	EstablishSession の入力引数の変換表
表 61	EstablishSession の出力引数の変換表
表 62	EstablishSession のメッセージ
表 63	EstablishSession の PortType の操作
表 64	EstablishSession の操作のバインディング
表 65	SessionPing の出力引数の変換表
表 66	SessionPing のメッセージ
表 67	SessionPing の PortType の操作
表 68	SessionPing の操作のバインディング
表 69	CloseSession の入力引数の変換表
表 70	CloseSession の出力引数の変換表
表 71	CloseSession のメッセージ
表 72	CloseSession の PortType の操作
表 73	CloseSession の操作のバインディング
表 74	EstablishSession の入力引数の変換表
表 75	装置機能エントリ値の例
表 76	EstablishSession の出力引数の変換表
表 77	RequestRejectError クラスの変換表
表 78	EnhancedEstablishSession のメッセージ
表 79	EnhancedEstablishSession の PortType の操作
表 80	EnhancedEstablishSession の操作のバインディング
表 81	XML スキーマ
表 82	SessionClient の PortType の定義
表 83	SessionClient のバインディングの定義
表 84	SessionClientportType
表 85	SessionClient のバインディング
表 86	SessionPing の出力引数の変換表
表 87	SessionPing のメッセージ
表 88	SessionPing の PortType の操作
表 89	SessionPing の操作のバインディング
表 90	SessionFrozen のメッセージ
表 91	SessionFrozen の PortType の操作
表 92	SessionFrozen の操作のバインディング
表 93	SessionClosed の入力引数の変換表
表 94	SessionClosed のメッセージ
表 95	SessionClosed の PortType の操作
表 96	SessionClosed の操作のバインディング

1 目的

1.1 本仕様の目的は、SSLを使用した認証の実装およびWSDL/SOAPの実装とSEMI E132 で定義される装置クライアントの認証と権限付与 (Equipment Client Authentication and Authorization) のUMLモデルに対応するXMLスキーマを提示することである。

2 範囲

2.1 本文書の範囲は、XMLスキーマでECAモデルを忠実に表現し、対応するWSDLサービスを表現することである。モデルに新たなドメインの情報や概念を追加するものではない。追加するものは、有用なWSDLやXMLのスキーマを提供するために必要なものだけである。

注意: この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使用者は、その責任において、適切な安全および健康上実施すべき事柄を確立し、また使用前に法規制やその他の制限への適用性を判断するものである。

3 制限

3.1 なし。

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 — Specification for XML Message Structures

SEMI E132 — Specification for Equipment Client Authentication and Authorization

4.2 ISO スタンダード¹

ISO/IEC 11578:1996 — Information Technology – Open Systems Interconnection – Remote Procedure Call (RPC)

4.3 NIST スタンダード²

NIST FIPS PUB 180-1 — Secure Hash Standard

4.4 OMG スタンダード³

OMG 01-09-67 — Unified Modeling Language, v1.4

4.5 W3C スタンダード⁴

Extensible Markup Language (XML) 1.0 (Fourth Edition) — W3C 16 August 2006; <http://www.w3.org/TR/2006/REC-xml-20060816/>

Namespaces in XML (Second Edition) — W3C 16 August 2006; <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

XML Schema Part 0: Primer (Second Edition) — W3C, 28 October 2004; <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

XML Schema Part 1: Structures (Second Edition) — W3C, 28 October 2004; <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

1 International Organization for Standardization, ISO Central Secretariat, 1 rue de Varembe, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30; <http://www.iso.ch>

2 National Institute of Standards and Technology, 100 Bureau Drive, Stop 3460, Gaithersburg, MD 20899-3460, USA. Telephone: 301.975.6478; <http://www.nist.gov>

3 Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, USA; Telephone: 781.444.0404; Fax: 781.444.0320; <http://www.omg.org>

4 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>

XML Schema Part 2: Datatypes (Second Edition) — W3C, 28 October 2004; <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) — W3C, 27 April 2007; <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>

SOAP Version 1.2 Part 2: Adjuncts (Second Edition) — W3C, 27 April 2007; <http://www.w3.org/TR/2007/REC-soap12-part2-20070427/>

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

4.6 その他のスタンダード

Secure Sockets Layer (SSL) Protocol 3.0; available from <http://wp.netscape.com/eng/ssl3/draft302.txt>

Web Services Interoperability (WS-I): Basic Profile Version 1.0a; <http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html>

Internet Engineering Task Force (IETF) RFC 2459 — Internet X.509 Public Key Infrastructure Certificate and CRL Profile; <http://www.ietf.org/rfc/rfc2459.txt>

Internet Engineering Task Force (IETF) RFC 2616 — Hypertext Transfer Protocol – HTTP/1.1; <http://www.ietf.org/rfc/rfc2616.txt>

PKCS #12: Personal Information Exchange Syntax Standard, version 1.0; available from <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-12/>

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 略語および頭字語

5.1.1 HTTP (Hyper Text Transport Protocol) — ハイパーテキストトランスポートプロトコル

5.1.2 SHA (Secure Hash Algorithm) — 一群の関連したハッシュ関数

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

5.1.4 SSL (Secure Sockets Layer) — セキュアソケットレイヤ

5.1.5 UML (Unified Modeling Language) — 統一モデル化言語

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

5.1.7 WSDL (Web Service Description Language) — ウェブサービス定義言語

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

5.2 定義

5.2.1 統一モデル化言語 (Unified Modeling Language (UML)) — オブジェクト指向の設計や観念を表現する表記法で, Booch, Rumbaugh, Jacobsonにより考案された3種の著名な方法論の表記法に加え, その他の使われていた方法論の観点を加味して, 誰もが利用できる様, 一連のオブジェクト指向の記法としたもの。

5.2.2 拡張マークアップ言語 (eXtensible Markup Language (XML)) — 文書や通信の文脈や内容を豊かに表現できるマークアップ言語。XMLはSGMLの拡張で文書化を指向したマークアップ言語である。W3Cがインターネットで使う目的で考案したものである。

6 表記法

6.1 UMLからXMLスキーマへの変換

6.1.1 読者は、UML, XML, WSDL, 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 および ¶4.3を参照)。本文書では、これらの題材については説明しない。

6.1.2 本文書は、SEMI E121 で概説されているXMLのガイドラインに従う。

6.1.3 本文書で従う重要ないくつかのガイドラインについてここで要約する。

- 装置からのメタデータ記述の転送効率を改善するために、UML クラスの属性は XML の属性として一般に表現される。この表記の例外は、XML 属性として表現できないデータ構造やその他の構造に対応する UML 属性であり、構文解析されない文字データ(CDATA)を含む必要があるテキスト値を含むことがある。
- 構成の関連は、単一の XML エlement (または、指定された多重度が 1 より大きい場合は、複数のそのようなElement) にマッピングされる。
- 集約は、集約内容の型のElementとして格納するようにマッピングするか、あるいは集約内容の型の 1 個以上の一意な属性の配列としてマッピングする。
- 継承は、complexType に extension 要素を使用してモデル化される。UML モデル上で抽象系クラスである継承がなされている場合には、型の置き換えによって無秩序な拡張がインスタンスドキュメントで起こらないように制限するために、派生する可能性がある型を各々並べる choice 要素による構成体を付けることとする。

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で記載する。UMLクラスで定義した操作は、WSDLでの操作に変換される。

6.1.5 変換見出し行の説明

- 属性名／役割名 (Attribute or Role Name) — 属性の場合は、その属性名をここに記載する。関連(集約や構成も含む)の場合は、UML 図の役割名をここに記載する。構成は多くの場合、役割名を明記しない。その場合は「なし(none)」と記載する。
- UML 名／型(UML Name/Type) — 属性の場合は、UML 属性のデータ型をここに記載する。関連の場合は、関連の型をここに記載する。記載できる関連の型は「構成(Composition)」、「集約(Aggregation)」、一般的な「関連(Association)」である。UML では、これらの 3 つの型の関連を定義する。
- XML Element／属性の区分(XML Element or Attribute) — UML 属性や関連を表現するために使われる XML による構成方法の種類を記載する。
- XML 名／型(XML Name/Type) — 結果的に得られる XML 構成単位名とデータ型を記載する。型名は、組み込み型(例えば, xs:string), あるいは XML スキーマで名称をつけて定義する型のどちらかである。

表 1 変換表の例

属性／役割名	UML 名／型	XML Element／属性の区分	XML 名／型
friend	関連 (association)	Element (element)	Friend: HumanReferenceArray
employees	集約 (aggregation)	Element (element)	Employees: HumanArray
family	構成 (composition)	Element (element)	Family: HumanArray
name	文字列 (string)	Element (element)	Name: xs:string

6.1.6 XMLスキーマへの変換

6.1.6.1 UMLインターフェイスクラスで定義される操作からXMLスキーマの型への変換は、表 2 に示す表形式で説明する。当該操作で使われる可能性のある各入力／出力引数に 1 行ずつ充てるものとする。

表 2 操作の入力／出力引数の変換表の例

引数名	形態	XML エlement／属性の区分	XML 名／型
<抽象規定での引数名>	<抽象規定での引数形式>	<引数がElementか属性かの何れでモデル化されるかを示す>	<属性／Elementのネームスペースを指定する名前>

6.2 XMLスキーマとWSDLファイルの説明方法

6.2.1 本仕様の中核の構成要素であるXMLスキーマとWSDLファイルが本書に付随するものである。この各説明資料は、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>

6.3 図によるXMLスキーマの説明⁵

6.3.1 本文書は、記載するXMLスキーマのデータ型を図形で説明する。XMLスキーマの標準的な図示方法はないが、様々な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。

6.3.2 本図では、矩形の箱がXMLElementの定義を示す。左側にオーナーElementを記載し、その右側にそのオーナーElementに格納されるElementを記載する。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（八角形の箱）は、順序(sequence)や選択(choice)を示す。これらの記号の説明については、表 4 を参照のこと。

表 4 Altova XMLSPY スキーマ図の記号

	すぐ右側に並んだElement, 原則個数各 1 個の順序つきの要素列(sequence)で、Elementの存在が必須と想定されるもの。
	すぐ右側に並んだElement, 原則個数各 1 個の、順序つきの要素列(sequence)で、Elementの有無は任意。
	すぐ右側に並んだElement, 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Elementの選択肢を示し、存在は必須である。 (左図の場合)1 以上 2 以下の右側のElementが存在しなければならない(choice)。
	構文解析された文字データを格納しているElementを示す。
	他の箇所 で定義されているグループやElementへの参照を示す。

6.3.3 実線を使用した図形は必須Elementを示し、破線は任意Element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。

⁵ 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。

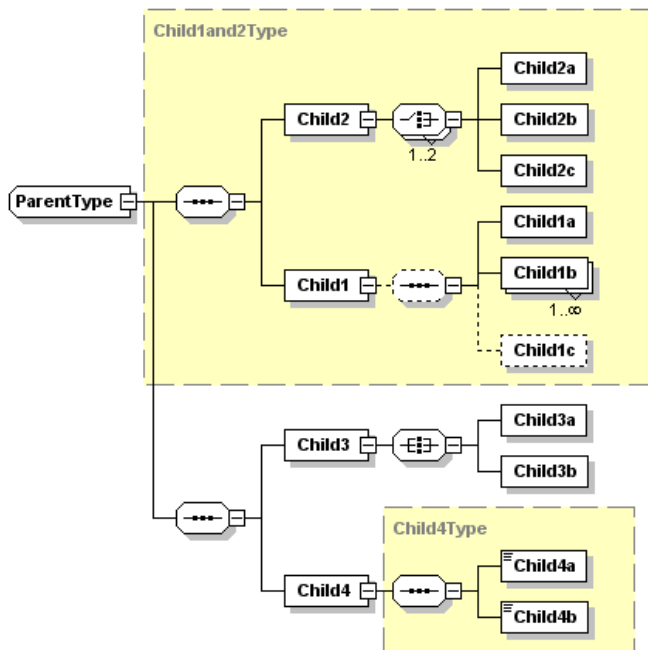


図 1
XML スキーマ図の例

6.3.4 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。この例では,詳細の表示がある。

6.3.5 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Child1and2Typeを拡張する。

6.3.6 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。

6.4 XMLスキーマの例

6.4.1 図1に示す例のXMLスキーマを次に示す。XMLスキーマの構文および意味の完全な説明については,¶4.3で引用したXML文書を参照すること。

```
<xs:element name="Parent">
  <xs:complexType>
    <xs:sequence>
      <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: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="Child3">
      <xs:complexType>
        <xs:all>
          <xs:element name="Child3a"/>
          <xs:element name="Child3b"/>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

図 2
例の XML

※ 6.3.5 項の説明は、図 2 とは一致していません。次版で正しい図と差換えられる予定です。

6.5 UMLからWSDL への変換

6.5.1 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 ポート型 (Port Type) 表の例

クラス名	<抽象規定での UML インターフェイス名>
WSDL のポートタイプ名	<WSDL 内で使用されるポートタイプ名>
SEMI E125 での操作名 → WSDL による操作	<UML インターフェイスからの操作名> → <WSDL ポート型の操作名>

表 6 インターフェイス WSDL バインディング表の例

SEMI E125 のクラス名	<抽象規定での UML インターフェイス名>
WSDL のバインディング名	<WSDL 内で使用されるバインディング名>
SOAP のバインディング形態	<RPC または文書>
SOAP の伝送先	<伝送を扱う URI>

6.5.2 任意のUMLインターフェイスクラスについて定義される各操作には、対応する操作の定義とバインディングがある。操作は、表 7 と表 8 に示す表形式で記述される。

表 7 操作バインディング表の例

SOAPAction	< Content-Type ヘッダ「action」の値で、この操作に使用されるもの>
入力ヘッダ (WSDL メッセージ, 必須)	<入力ヘッダを規定する WSDL メッセージ名前, ならびに、そのヘッダが必須か否か>
出力ヘッダ (WSDL メッセージ, 必須)	<出力ヘッダを規定する WSDL メッセージ名前, ならびに、そのヘッダが必須か否か>

表 8 PortType 操作表の例

入力メッセージ名	<WSDL ポート型 (port type) の操作の入力として使用される WSDL メッセージ名>
出力メッセージ名	<WSDL ポート型 (port type) の操作の出力として使用される WSDL メッセージ名>

7 SEMI E132 の認証のSSLへのマッピング

7.1 SSLの概要

7.1.1 本項では、SSLセキュリティプロトコルの使用方法およびSEMI E132 に記述されている認証スキームを実装するために使われる技術について説明する。SSLは独立したアプリケーションプロトコルであり、HTTPによるSOAPなど、より高い層の任意のプロトコルをユーザに意識させることなくサポートする。

7.1.2 図 3 に、安全な接続を確立するための基本メッセージフロー (ハンドシェイク) およびSEMI E132 の認証メッセージへのマッピング方法を示すが、SSLの完全な仕様については § 4.6 で参照されているSSLの文書を参照すること。本仕様では、SEMI E132 の認証を実装するために最小限必要なSSLの機能について制限事項をいくつか定義する。

7.1.3 SSLに関するSEMI E132 の制限事項

7.1.3.1 RFC 2459 で定義されているX.509 v3 公開鍵証明書を使用して装置とクライアントのアイデンティティを確立しなければならない。X.509 の構造と内容の詳細については § 4.6 で参照されているX.509 の文書を参照すること。SSLハンドシェイク中、必要なときに証明書が送信される。

7.1.3.2 相互認証が必須である。装置は、SSLハンドシェイクを行う際にはいつもその証明書を送信し、クライアントの証明書を要求しなければならない。それに対応し、クライアントは装置に自分の証明書と証明検証メッセージを送信することで、クライアントの認証が可能となる。

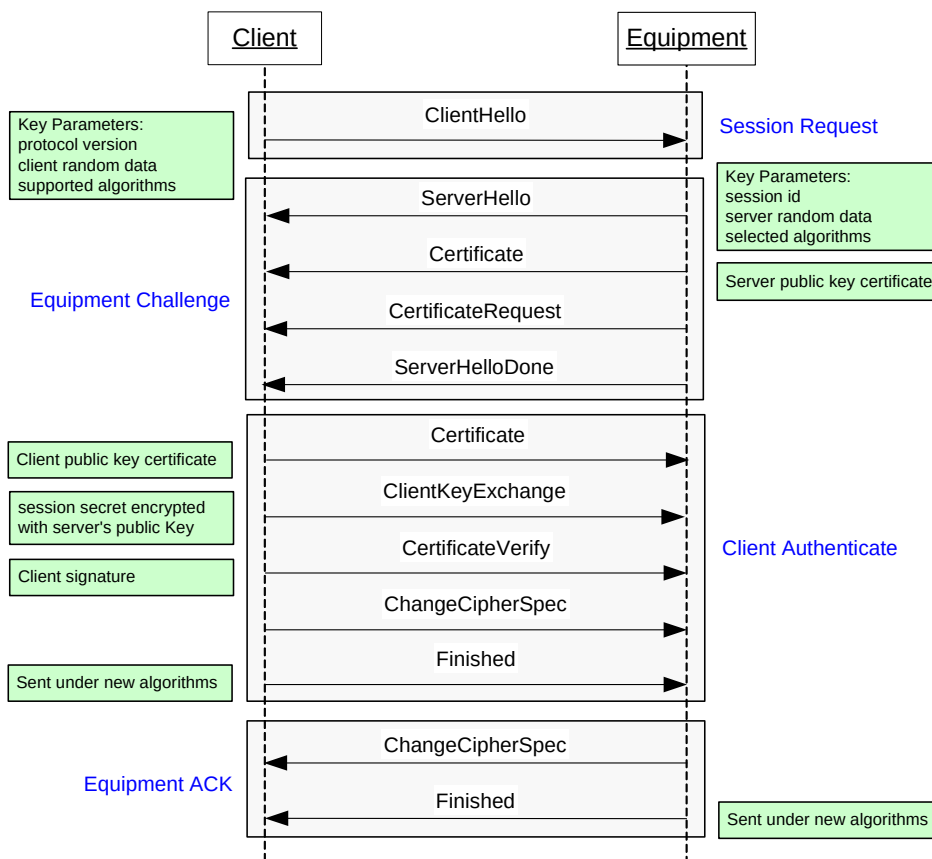


図 3
SSL 基本ハンドシェイクのフロー

7.2 資格認定およびデジタル証明書

7.2.1 概要

7.2.1.1 X.509 の証明書を使用して装置とアプリケーションクライアントのアイデンティティを確立しなければならない。この証明書はSSLによって認証に使用され、装置とクライアントの間の安全な接続を確立する。証明書とACLの役割を示すSEMI E132.1 のセキュリティ資格認定の図を図 4 に示す。この図に示すように、X.509 の証明書は次の項で説明するCAと呼ぶエージェントによって管理される公開キーにIDをバインドする。ACLエントリは、セキュリティ管理 (Security Admin) によって管理される付与権限の集合にIDをバインドする。付与権限の管理、ACLエントリ、セキュリティ管理については、SEMI E132 で定義される通りである。付与権限の項を参照すること。

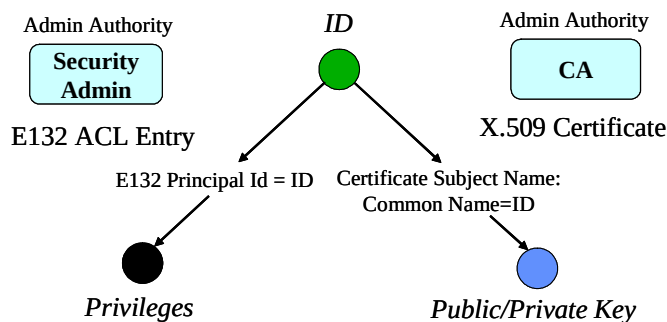


図 4
SEMI E132.1 のセキュリティ資格認定

7.2.1.2 CAは、SSLによって通信したい装置とアプリケーションクライアントに証明書を発行する証明オーソリティである。証明書の真偽は証明書内のCAシグニチャを使って確認されるので、実質的には証明書に格納されている情報の有効性はCAが証明していることになり、さらに重要なことは、CAは証明書内で識別されるエンティティとSSLで認証用途に使用される公開キーの間のバインディングを証明していることである。証明書の有効性およびシグネチャ照合の詳細については、§ 4.6 で参照されているX.509 v3 の文書を参照すること。SEMI E132 で定義されているように、CAは、アプリケーションクライアントのアクセス制限に関する許可を管理するSecurity Adminと同じであっても違って構わない。CAとセキュリティ管理の両方の実装とメンテナンスは、証明書の発行およびセキュリティ資格認定の管理も含め、装置ユーザの責任である。装置サプライヤは、CAやセキュリティ管理の実装を提供する必要はない。

7.2.1.3 本仕様では、証明書の対象を区別する名(DN)の公開名(CN)属性だけが必要であり、これは図 4 に示すようにIDとして表現される。権限付与をACLEntry内の主idにバインドするときに、セキュリティ管理者は同じIDを使用しなければならない。要約すると、SEMI E132 のPrincipalは、Principalに対して発行される証明書の対象区別名フィールドの公開名(CN)属性によって識別される。

7.2.1.4 SEMI E132 のCAによって発行される証明書は、対応する秘密キーに従ってPKCS#12 フォーマットでエンコードしなければならない。装置は、CAによって発行される証明書および関連する任意の秘密キーをデコードするためにPKCS#12 をサポートしなければならない。証明書のローカルのエンコードフォーマットについて、制限はない。完全な詳細については、§ 4.6 で参照されているPKCS#12 の文書を参照すること。CA証明書も同じフォーマットで送られるが、秘密キーはない。CA証明書はSSLによって証明書の検証に使用される。

7.2.1.5 証明書内にKeyUsage拡張フィールドが存在する場合、証明書は少なくともサイン機能をサポートしなければならない。具体的に言えば、digitalSignatureビットが有効にされていなければならない。証明書のプロファイルの詳細については、X.509 v3 の仕様書を参照すること。

7.2.2 装置資格認定の確立

7.2.2.1 装置には、CAの公開証明書および、装置自体の証明書の両方を対応する秘密キーとともにインストールしていなければならないが、クライアントの証明書を照合するCA証明書は、信頼されているCAと同一からもたらされた証明書である。装置は、PKCS#12 でエンコードされたCAからの証明書および秘密キーを、それ自体の実装によってサポートされているフォーマットにデコードする責任がある。

7.2.2.2 装置のSecurity Adminを示すACLEntryは、装置のローカルなインターフェイスを使用して与えなければならない。これは、装置のアクセス許可するリモート管理を可能にして、装置の設置の際に行わなければならない。

7.2.3 アプリケーション資格認定の確立

7.2.3.1 何れかの装置とSEMI E132 のセッションを確立したいアプリケーションクライアントは、如何なる通信にも先んじて、まずCAによって発行された有効な証明書を保持しなければならない。装置は、信頼のあるCAによって発行された有効な証明書を提示しないクライアントとのSSLハンドシェイクは何れも確実に中断するようにしなければならない。装置証明書と同様に、クライアント証明書およびCAからの対応する秘密キーは何れも、PKCS#12 フォーマットでエンコードされるものとする。アプリケーションクライアントはCA証明書もインストールして装置証明チェーンを検証しなければならない。

7.2.3.2 各クライアントは、装置とSEMI E132 のセッションを確立する前に、クライアントが機能する様に、装置上で定義され、適切な権限を付与するACLEntryも持っていなければならない。これは、Security Adminが装置コンソールを介して付与可能なものである。

7.2.3.3 信頼できるCAが発行した有効な証明書をクライアントが持っても、定められたACLEntryをクライアントが持っていない場合は何れも、装置はSEMI E132 セッション確立要求を拒否しなければならない。

7.3 SSL認証の有効化／無効化

7.3.1 クライアントが装置と通信するための認証機構を装置やクライアントに実装することは、実際的あるいは実現可能ではないような統合、開発、テスト環境もある。そのような環境だけに限定して、現実的な配慮として、装置は、装置上でSSL認証を有効／無効にできる機能をサポートしなければならない。

7.3.2 認証が有効である場合、7.1 に記述されているように、装置とのすべてのSEMI E132 に即した通信は、相互認証SSLによって行わなければならない。

7.3.3 認証が無効である場合、装置とのすべてのSEMI E132 に即した通信は、通常の認証なしのHTTPトランスポートによって行わなければならない。無効にされている間、依然として要求は、SEMI E132 のヘッダのFROMフィールドで提供されるクライアントIDを使用した通常の付与権限の照合方法に従わなければならないが、要求が認証されないため、要求しているアプリケーションが信頼できるものであることを保証できないことがある。SSL認証を無効にすると、装置へネットワークからアクセスできるアプリケーション（工場で許可されているもの、あるいはないもの）は、SEMI E132 の装置の操作の何れにも無制限にアクセスできるということをユーザは認識していなければならない。

7.3.4 装置は、SSL通信と非SSL通信を切り替えることができる設定オプションを提供しなければならない。切り替え可能な通信ポートは、常に 1 つだけがアクティブでなければならない。アクティブだったポートは変更後の要求にはもはや使用できなくなり、装置は変更についてクライアントには通知しないので、設定変更時に際して、変更前に既存のセッションをクローズすることは装置ユーザの責任である。

7.3.5 SEMI E132 Session State Modelで定義されているように、予め存在していた持続セッションがあれば、設定変更後もアクティブなままでなければならない。変更後にこれらのセッションによって行われるすべての要求は新しいポートで発生することをユーザは認識していなければならない。装置は、アクティブな非持続セッションを設定変更後に保持する必要はない。SEMI E132 Session State Modelで定義されているように、この設定を変更するために装置のリセットが必要な場合、装置シャットダウンについては、継続セッションは再開し、非継続セッションはクローズしなければならない。

8 SEMI E132 に関するUMLからのXMLスキーマおよびWSDLへのマッピング

8.1 WSDLの構成

8.1.1 SEMI E132 の各インターフェイスの定義は、WSDLのportTypeおよびバインディングの定義にマッピングされる。WSDLのportTypeの定義は、SEMI E132 に記載に即してインターフェイスおよびその操作を基に名前が付けられる。各portTypeのWSDLバインディングの定義を使用して、各操作に対するSOAP 1.2 のエンベロープの内容を規定し、対応するXMLのエンコードスタイルおよびHTTPヘッダの使用方法を定義する。SEMI E132 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、グローバルエレメントの定義を使用して、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。

8.1.2 図 5 で示しているのは、WSDLバインディング、portTypeの定義、各インターフェイスに使用するXMLスキーマの型の関係である。portTypeの定義は各操作で使用するXMLスキーマ型の定義を、XMLスキーマの「import」文で取り込む。WSDLのバインディングの定義は、portTypeの定義をWSDLの「import」文で取り込む。

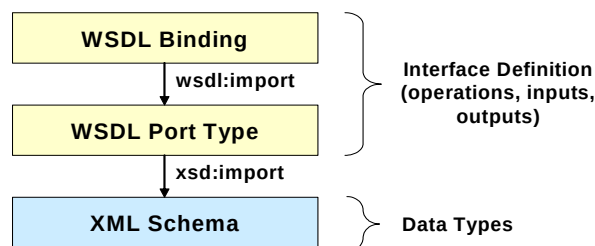


図 5

XML スキーマおよび WSDL ファイルの構成

8.1.3 図 6 は、SEMI E132 専用に定義したXMLスキーマとWSDLファイルを示したものである。

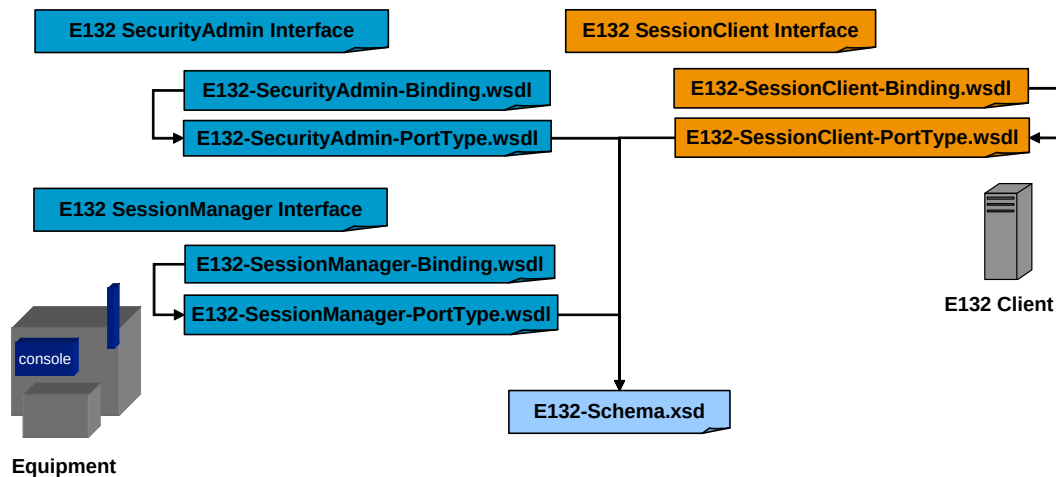


図 6
SEMI E132 の WSDL および XML スキーマファイル

8.2 SecurityAdmin インターフェイス

8.2.1 Security Admin の ACLEntry の確立

8.2.1.1 Security Admin 定める ACLEntry は、装置のローカルなインターフェイスを使用して装置導入時に与えなければならない。ACLEntry の科目 id は、Security Admin に対して発行された証明書の科目を区別する名前である公開名 (CN) 属性と一致しなければならない。『7.2.1.3 を参照すること。SEMI E132 で定義されているように、Security Admin の ACLEntry は、予約されている権限「urn:semi-org:auth:securityAdminPrivileges」を付与しなければならない。

8.2.1.2 Security Admin について、ただ 1 つの ACLEntry が存在しなければならない。

8.2.2 XML スキーマおよび WSDL ファイル

8.2.2.1 SecurityAdmin インターフェイスについて本仕様が定義する XML スキーマおよび WSDL は次の文書に掲載する。

表 9 XML スキーマ

ファイル名	E132-1-V0310-Schema.xsd
ターゲットネームスペース	urn:semi-org:xsd:E132-1.V0310.auth
インポートされる／参照されるネームスペース	http://www.w3.org/2001/XMLSchema urn:semi-org:xsd:E138.0709.ccs urn:semi-org:xsd:E128.V0706.xms
説明	このファイルは SecurityAdmin, SessionManager, SessionClient の各インターフェイスが使用する SEMI E132 のデータ型とエレメントのすべてを定義する。

表 10 SecurityAdmin の PortType の定義

ファイル名	E132-1-V0310-SecurityAdminPortType.wsdl
ターゲットネームスペース	urn:semi-org:ws:E132-1.V0310.SecurityAdminPortType
インポートされる／参照されるネームスペース	urn:semi-org:xsd:E132-1.V0310.auth http://schemas.xmlsoap.org/wsdl/ http://www.w3.org/2001/XMLSchema urn:semi-org:xsd:E128.V0706.xms
説明	このファイルは、SEMI E132 の XML スキーマで定義されるデータ型に基づいて、SecurityAdmin インターフェイスに使用されるすべての入力／出力メッセージおよび操作を定義する。

表 11 SecurityAdmin のバインディングの定義

ファイル名	E132-1-V0310-SecurityAdmin-Binding.wsdl
ターゲットネームスペース	urn:semi-org:ws.E132-1.V0310.SecurityAdminBinding
インポートされる／ 参照されるネームスペース	urn:semi-org:ws.E132-1.V0310.SecurityAdminPortType urn:semi-org:ws.E132-1.V0310.SecurityAdminPortType http://schemas.xmlsoap.org/wsdl/soap12/ http://schemas.xmlsoap.org/wsdl/
説明	このファイルは、SecurityAdmin インターフェイスのそれぞれの操作および XML エンコードスタイルについて必要な SOAP ヘッダと HTTP ヘッダを指定して、抽象 portType の定義を HTTP および SOAP にバインドする。

8.2.2.2 これらの文書は、本仕様の一部であり、本文書に付属しなければならない。これらの文書の内容が、本仕様の核心部分を構成する。

8.2.3 WSDL Port Type の概要

8.2.3.1 WSDLのSecurityAdmin のportTypeの定義は、SEMI E132.1 のXMLスキーマのデータ型を、SEMI E132 でSecurityAdminインターフェイスとして定義したUMLでの操作に対応して命名した操作に、入出力も伴う集合体に使うものとして構成する。WSDLのportType自体は、SEMI E132 のインターフェイスを基に命名され、WSDL portTypeの各操作は、インターフェイスとして定義された、対応するUMLでの操作に基づいて命名される。WSDLの各操作は、SEMI E132 で定義されるUML操作での入出力(要求と応答)に対応した 2 つのWSDLメッセージの定義で構成する。図 7 に、SEMI E132 のUMLインターフェイスとWSDL portTypeの定義の関係を示し、表 12 に表形式で示す。WSDLのメッセージと操作の定義は、§8.2.9 から§8.2.15 で説明する。

8.2.3.2 SecurityAdminのポート型(port type)の操作の詳細は、表 10 に記載されたwsdlファイルに定義されている。

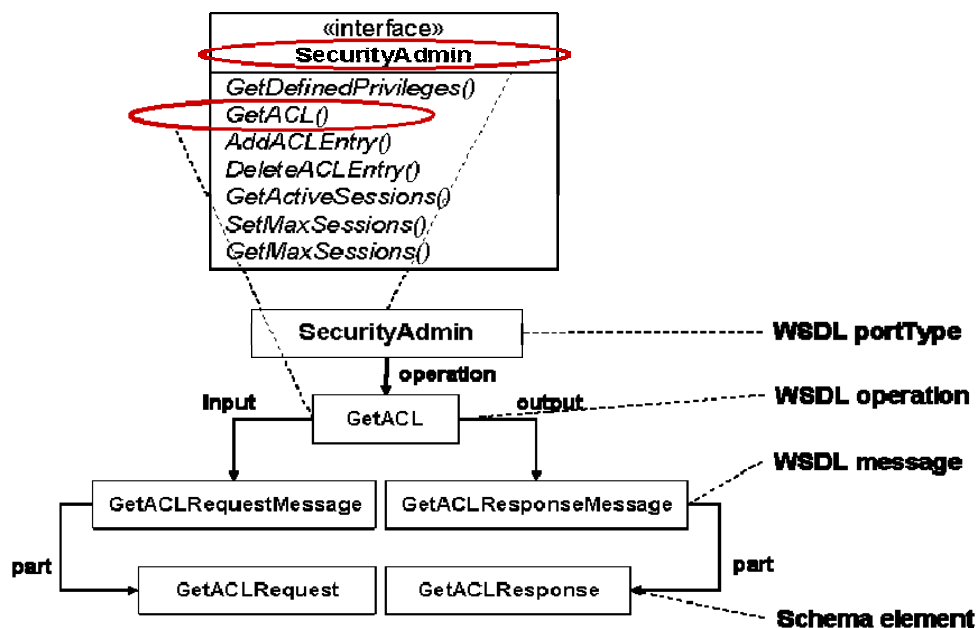


図 7

SecurityAdmin インターフェイスの WSDL の portType へのマッピング