

SEMI E134.1-0710

データ収集管理（DCM）の SOAP バインディングに関する仕様

NOTICE: This translation is a REFERENCE COPY ONLY. If differences should exist between the English version and a translation in any other language, the English version is the official and authoritative version.

免責事項： この SEMI スタンドアードは、投票により作成された英語版が正式なものであり、日本語版は日本の利用者各位の便宜のために作成したものです。万が一英語と日本語とに差異がある場合には英語版記載内容が優先されます。

SEMI スタンドアード日本語翻訳版をご利用にあたっての注釈を本文の末尾に記載しております（「すべきである」「しなければならない」について等）。

本スタンドアードは、Information & Control Global Technical Committee で技術的に承認されている。現版は 2010 年 4 月 30 日、global Audits and Reviews Subcommittee にて発行が承認された。2010 年 6 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 2010 年 3 月に発行された。

目次

- 1 目的
 - 1.1 SOAP 実装マッピング
- 2 範囲
 - 2.1 仕様範囲
- 3 制限
- 4 参照スタンドアードおよび文書
 - 4.1 SEMI スタンドアード
 - 4.2 OMG スタンドアード
 - 4.3 W3C スタンドアード
 - 4.4 その他のスタンドアード
- 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 ファイルの説明方法
 - 6.3 XML スキーマ図による説明
 - 6.4 XML スキーマの例
 - 6.5 UML から WSDL への変換
- 7 SEMI E134 の UML の XML スキーマおよび WSDL へのマッピング
 - 7.1 WSDL の構成
 - 7.2 DataCollectionManager
 - 7.3 DCPConsumer
- 8 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 XML スキーマおよび WSDL ファイルの構成
- 図 4 SEMI E134 の WSDL および XML スキーマファイル
- 図 5 DataCollectionManager の WSDL portType へのマッピング
- 図 6 DefinePlanRequest

- ☒ 7 DataCollectionPlanType
- ☒ 8 EventRequestType
- ☒ 9 TraceRequestType
- ☒ 10 EventTriggerType
- ☒ 11 DefinePlanResponseType
- ☒ 12 InvalidPlanType
- ☒ 13 InvalidEventRequestType
- ☒ 14 InvalidTraceRequestType
- ☒ 15 GetDefinedPlanIdsRequest
- ☒ 16 GetDefinedPlanIdsResponse
- ☒ 17 GetPlanDefinitionRequest
- ☒ 18 GetPlanDefinitionResponse
- ☒ 19 ActivatePlanRequest
- ☒ 20 ActivatePlanResponse
- ☒ 21 GetActivePlanIdsRequest
- ☒ 22 GetActivePlanIdsResponse
- ☒ 23 DeactivatePlanRequest
- ☒ 24 DeactivatePlanResponse
- ☒ 25 DeletePlanRequest
- ☒ 26 DeletePlanResponse
- ☒ 27 GetParameterValuesRequest
- ☒ 28 GetParameterValuesResponse
- ☒ 29 GetObjTypeInstanceIdsRequest
- ☒ 30 GetObjTypeInstanceIdsResponse
- ☒ 31 GetCurrentPerformanceStatusRequest
- ☒ 32 GetCurrentPerformanceStatusResponse
- ☒ 33 GetCurrentDateTimeRequest
- ☒ 34 GetCurrentDateTimeResponse
- ☒ 35 NewDataNotification
- ☒ 36 EventReportType
- ☒ 37 ExceptionReportType
- ☒ 38 CollectedDataType
- ☒ 39 PerformanceWarningNotification
- ☒ 40 PerformanceRestoredNotification
- ☒ 41 DCPDeactivationNotification
- ☒ 42 DCPHibernationNotification
- ☒ 43 DCPDefinedNotification
- ☒ 44 DCPDeletedNotification

表リスト

- 表 1 変換表の例
- 表 2 操作入力／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 のポート型への変換表の例
- 表 6 インターフェイス WSDL バインディングへの変換表の例
- 表 7 操作バインディング表の例
- 表 8 PortType 操作表の例
- 表 9 SEMI E134 から SEMI E132 への権限のマッピング

表 10	ObjType イベント／例外／パラメータの表記方法
表 11	ObjType を使ったイベント要求の例
表 12	SEMI E134 のエラーコード
表 13	XML スキーマ
表 14	DataCollectionManager の PortType の定義
表 15	DataCollectionManager のバインディングの定義
表 16	DefinePlan の操作のバインディング
表 17	DefinePlan PortType の操作
表 18	DataCollectionManagement のバインディング
表 19	DefinePlan の操作の入力引数の変換表
表 20	DataCollectionPlanType の変換表
表 21	EventRequestType の変換表
表 22	ParameterRequestType の変換表
表 23	ExceptionRequestType の変換表
表 24	TraceRequestType の変換表
表 25	EventTriggerType の変換表
表 26	ConditionType の変換表
表 27	ParameterReference の変換表
表 28	ExceptionTriggerType の変換表
表 29	ParameterRequestType の変換表
表 30	DefinePlan の操作の出力引数の変換表
表 31	DCPDefinedType の変換表
表 32	InvalidPlanType の変換表
表 33	InvalidEventRequestType の変換表
表 34	InvalidExceptionRequestType の変換表
表 35	InvalidParameterRequestType の変換表
表 36	InvalidTraceRequestType の変換表
表 37	InvalidTriggerType の変換表
表 38	InvalidCondition の変換表
表 39	InvalidIntervalType の変換表
表 40	InvalidGroupSizeForIntervalType の変換表
表 41	InvalidCycleType の変換表
表 42	GetDefinedPlanIds の操作のバインディング
表 43	GetDefinedPlanIds の PortType の操作
表 44	DataCollectionManagement のバインディング
表 45	GetDefinedPlanIds の操作の出力引数の変換表
表 46	GetPlanDefinition の操作のバインディング
表 47	GetPlanDefinition の PortType の操作
表 48	DataCollectionManagement のバインディング
表 49	GetPlanDefinition の操作の入力引数の変換表
表 50	GetPlanDefinition の操作の出力引数の変換表
表 51	NoSuchPlanType の変換表
表 52	ActivatePlan の操作のバインディング
表 53	ActivatePlan の PortType の操作
表 54	DataCollectionManagement のバインディング
表 55	ActivatePlan の操作の入力引数の変換表
表 56	GetDefinedPlanIds の操作の出力引数の変換表
表 57	DCPActivatedType の変換表
表 58	GetActivePlanIds の操作のバインディング

表 59	GetActivePlanIds の PortType の操作
表 60	DataCollectionManagement のバインディング
表 61	ActivatePlan の操作の出力引数の変換表
表 62	DeactivatePlan の操作のバインディング
表 63	DeactivatePlan の PortType の操作
表 64	DataCollectionManagement のバインディング
表 65	DeactivatePlan の操作の入力引数の変換表
表 66	DeactivatePlan の操作の出力引数の変換表
表 67	DeactivatePlanResponseType の変換表
表 68	DCPNotActiveType の変換表
表 69	DeletePlan の操作のバインディング
表 70	DeletePlan の PortType の操作
表 71	DataCollectionManagement のバインディング
表 72	DeletePlan の操作の入力引数の変換表
表 73	DeletePlan の操作の出力引数の変換表
表 74	DCPDeletedType の変換表
表 75	GetParameterValues の操作のバインディング
表 76	GetParameterValues の PortType の操作
表 77	DataCollectionManagement のバインディング
表 78	GetParameterValues の操作の入力引数の変換表
表 79	GetParameterValues の操作の出力引数の変換表
表 80	GetObjTypeInstanceIds の操作のバインディング
表 81	GetObjTypeInstanceIds の PortType の操作
表 82	DataCollectionManagement のバインディング
表 83	GetObjTypeInstanceIds の操作の入力引数の変換表
表 84	ObjTypeRequestType の変換表
表 85	GetObjTypeInstanceIds の操作の出力引数の変換表
表 86	ObjTypeResultType の変換表
表 87	ObjTypeInstanceType の変換表
表 88	ObjTypeRequestErrorType の変換表
表 89	GetCurrentPerformanceStatus の操作のバインディング
表 90	GetCurrentPerformanceStatus の PortType の操作
表 91	DataCollectionManagement のバインディング
表 92	GetCurrentPerformanceStatus の操作の出力引数の変換表
表 93	PerformanceStatusType の変換表
表 94	ActiveJobType の変換表
表 95	JobTypeEnumType に関して選択可能なジョブの種類を示す表
表 96	AffectedSystemType の変換表
表 97	GetCurrentDataTime の操作のバインディング
表 98	GetCurrentDataTime の PortType の操作
表 99	DataCollectionReportType のバインディング
表 100	GetCurrentDataType の操作の出力引数の変換表
表 101	XML スキーマ
表 102	DCPConsumer の PortType の定義
表 103	DCPConsumer のバインディングの定義
表 104	NewData の操作のバインディング
表 105	NewData の PortType の操作
表 106	DCPConsumer のバインディング
表 107	NewDataNotification の操作の入力引数の変換表

表 108	DataCollectionReportType の変換表
表 109	EventReportType の変換表
表 110	ExceptionReportType の変換表
表 111	TraceReportType の変換表
表 112	CollectedDataType の変換表
表 113	PerformanceWarning の操作のバインディング
表 114	PerformanceWarning の PortType の操作
表 115	DCPConsumer のバインディング
表 116	PerformanceWarning 操作の入力引数の変換表
表 117	PerformanceRestored 操作のバインディング
表 118	PerformanceRestored の PortType の操作
表 119	DCPConsumer のバインディング
表 120	PerformanceRestored の操作の入力引数の変換表
表 121	DCPDeactivation の操作のバインディング
表 122	DCPDeactivation の PortType の操作
表 123	DCPConsumer のバインディング
表 124	DCPDeactivation の操作の入力引数の変換表
表 125	DCPHibernation の操作のバインディング
表 126	DCPHibernation の PortType の操作
表 127	DCPConsumer のバインディング
表 128	DCPHibernation の操作の入力引数の変換表
表 129	DCPHibernatedType の変換表
表 130	DCPDefined の操作のバインディング
表 131	DCPDefined の PortType の操作
表 132	DCPConsumer のバインディング
表 133	DCPDefined の操作の入力引数の変換表
表 134	DCPDeleted の操作のバインディング
表 135	DCPDeleted の PortType の操作
表 136	DCPConsumer のバインディング
表 137	DCPDeleted の操作の入力引数の変換表

1 目的

1.1 SOAP 実装マッピング

1.1.1 本仕様の目的は、SEMI E134 データ収集管理仕様の SOAP 1.2 プロトコルに対する実装用マッピングを提供することである。本文書は、SEMI E134 が定義する UML クラスを具体化するために使用される XML スキーマデータ型の説明を行い、SEMI E134 で指定するインターフェイスが定義する操作を実現するために使用される、WSDL ポート型とバインディング定義についても説明する。

2 範囲

2.1 本書の範囲は、XML スキーマで DCM モデルを忠実に表現し、対応する WSDL ポート型へのバインディングを表現することである。SEMI E134 の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。

2.2 本仕様は SEMI E132 の認証が必要であり、SEMI E132 の概念を SEMI E134 仕様に適用するための要求条件を定義する。

注意： このスタンダードおよび安全ガイドラインはその使用に関連した全ての安全上の問題を取り扱うことを意図していない。この文書の使用者は、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項を決定し、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

4.1 SEMI スタンダードおよび安全ガイドライン

SEMI E120.1 — XML Schema for the Common Equipment Model

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

SEMI E125.1 — Specification for SOAP Implementation of Equipment Self Description

SEMI E128 — Specification for XML Messaging

SEMI E132.1 — Specification for SOAP Implementation of Equipment Client Authentication and Authorization

SEMI E138 — XML Semiconductor Common Components

4.2 OMG スタンダード¹

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

4.3 W3C スタンダード²

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/>

¹ Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, U.S.A. Telephone: +1.781.444.0404; Fax: +1.781.444.0320; <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: +1.617.253.2613; Fax: +1.617.258.5999; <http://www.w3.org>

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

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>

4.4 その他のスタンダード

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

Secure Sockets Layer (SSL); <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>

注意： 別途指示のない場合、引用された全ての文書は最新版を指す。

5 用語

5.1 略語および頭字語

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*) — 統一モデルリング言語

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*) — ウェブサービス定義言語

5.1.5 *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。XML はオブジェクト指向の構造を表現することが可能である。

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*, スキーマ仕様の実際上の知識を有していることを前提とする (¶ 4.2 と ¶ 4.3 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 本文書で従うガイドラインの重要な部分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、SEMI E125 の 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 エレメント (または、指定された多重度が 1 より大きい場合は、複数のそのようなエレメント) にマッピングされる。
- 集約は、集約内容の型のエレメントとして格納するようにモデル化されるか、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モデル化される。
- 継承は、complexType に extension 要素を使用してマッピングする。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 の型の関連を定義する。
- XML エレメント／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, xsd: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 Attribute or Role Name	UML 名／型 UML Name/Type	XML エレメント／属性の区分 XML Element or Attribute	XML 名／型 XML Name/Type
friend	関連 (association)	エレメント (element)	Friend: HumanReferenceArray
employees	集約 (aggregation)	エレメント (element)	Employees: HumanArray
family	構成 (composition)	エレメント (element)	Family: HumanArray
name	文字列 (string)	エレメント (element)	Name: xsd:string

6.1.6 XML スキーマへの操作の変換

6.1.6.1 UML のインターフェイスクラスで定義されている操作から 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	形式	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 本仕様の共通の構成要素である 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

6.3.1 本文書は、記載する XML スキーマのデータ型やElementの定義を図で説明する。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、矩形状の箱が XML Element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, Child3, Child4 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 個の順序付きの要素列（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，順序付きの要素列（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 つ以上 2 つ以下の右側の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。

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，他で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，この詳細は図では示さなくても良い。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 Child1and 2Type 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 の順序付きの要素列である。
2. Child1 は，Child1a，1 つまたは複数の Child1b，（有無が任意の）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。

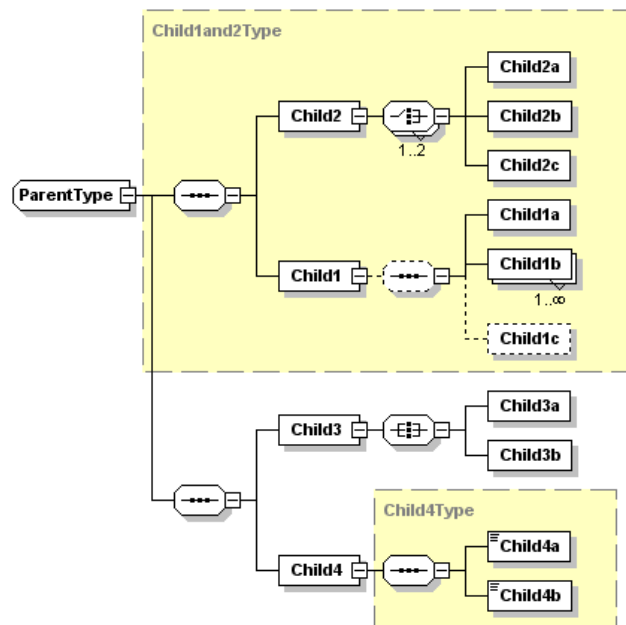


図 1
XML スキーマ図の例

6.4 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
上記の例の XML

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 インターフェイスに対する 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型の名>
SEMI E125 での操作名 → WSDL による操作	<UML インターフェイスでの操作名> → <WSDL ポート型の操作名>

表 6 インターフェイスに対する 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 HTTP 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 の操作表の例

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 E134 の UML からの XML スキーマおよび WSDL へのマッピング

7.1 WSDL の構成

7.1.1 SEMI E134 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 E134 の記載に即して、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 E134 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。

7.1.2 図 3 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 4 は、SEMI E134 専用に定義された XML スキーマおよび WSDL ファイルを示したものである。

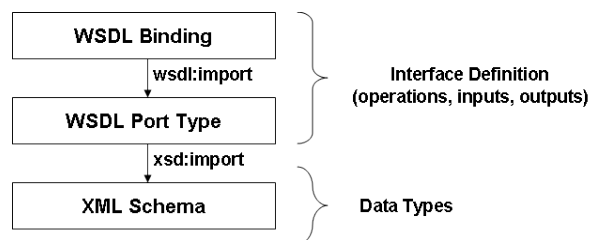


図 3

XML スキーマおよび WSDL ファイルの構成

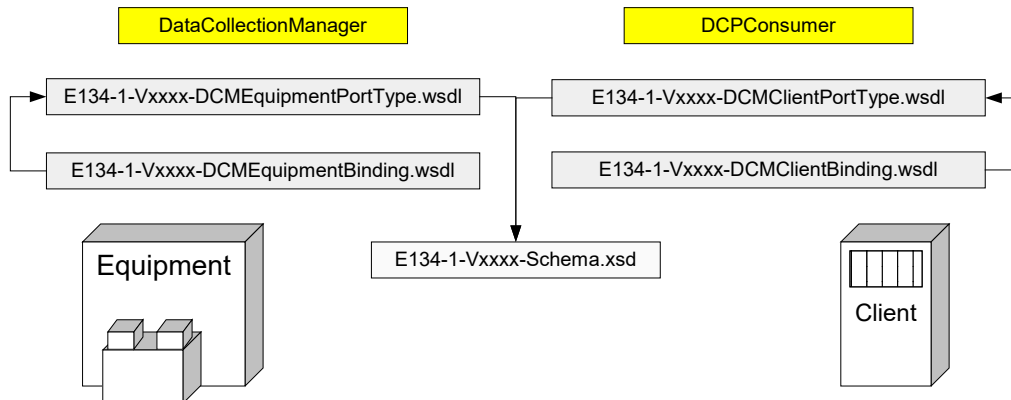


図 4
SEMI E134 の WSDL および XML スキーマファイル

7.2 DataCollectionManager

7.2.1 SEMI E132 Session の適用

7.2.1.1 DataCollectionManager は、認証と権限付与の手段として SEMI E132.1 を使用しなければならない。

7.2.1.2 データを収集するために SEMI E134 で定義されているインターフェイスを使用すると、装置のデータスループット全般に直接影響することがあり、装置本来の機能の実行能力に間接的に影響することがある。さらに、SEMI E134 のインターフェイスによって装置からアクセス可能なデータは、装置のサプライヤや装置を使用する工場にとっての専有情報であることがあり、工場内の権限が付与されていないアクセスに対して保護しなければならない。

7.2.1.3 このため、SEMI E134 の DataCollectionManager インターフェイスが定義するすべての操作は、SEMI E132.1 で指定されているような SSL 認証を有効にすることにより、HTTPS トランスポート (SSL 3.0 による HTTP 1.1) を使用して交換すれば良い。SEMI E132.1 が介在することになると、装置がサポートする SEMI E134 DataCollectionManager の何れの操作にも、クライアントとサーバの相互の SSL 認証を装置は要求する必要がある、HTTPS による伝送である場合を除き、何れの操作にも装置はアクセスを許可してはならない。

7.2.2 クライアントが装置と通信するための証明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、SEMI E132.1 が規定する装置構成設定を使って、SEMI E134 の DataCollectionManager インターフェイスに関して SSL 認証をユーザが無効にできる機能をサポートしなければならない。SSL 認証を無効にすると、装置にネットワーク経由でアクセスするすべてのアプリケーションに対して、SEMI E134 のデータへのアクセスを際限なしに許可することになるということをユーザは認識する必要がある。

7.2.2.1 SSL 認証が有効であるかどうかにかかわらず、要求に伴う SOAP エンベロープのヘッダ内で SEMI E128 のセッション識別子が有効でない場合、クライアントからの要求はすべて拒否しなければならない。SEMI E128 の仕様の通り、SEMI E128 のセッション識別子は、SyncHeader エlement として示す必要がある。クライアントが正当なセッション識別子を提示しない場合、装置は、SEMI E132.1 の仕様の通り、source 属性を「urn:semi-org:E132」としたエラーコード「6005」の、SEMI 共通エラーデータ型を使用して直ちにその要求を拒否しなければならない。要求に有効でない SEMI E128 セッション識別子が含まれている、あるいはセッション識別子がないことがわかった時点で、装置は、SEMI E134 の要求をそれ以上実行してはならない。

7.2.3 セッションおよびDCPの状態の依存関係

7.2.3.1 装置は、セッション単位で DCP アクティビティを保持しなければならない。2 つの別の SEMI E132 セッションが同じ DCP を起動しようとした場合、装置はどれのセッションについても DCP を起動しなければならない。同様に、アクティブな 1 つの SEMI E132 セッションが同一の DCP を 2 回起動しようとする、SEMI E134 の仕様の通り、装置は二番目の要求に対して DCPIsActive エラーで応答しなければならない。

7.2.3.2 ¶ 7.2.1.2 で説明しているように、装置はどれの SEMI E134 通信を送信する際にも、SEMI E132 EstablishSession 要求で得られる HTTPEndpoint 情報が示す URL を使用しなければならない。

7.2.3.3 コンシューマの SEMI E132.1 のセッションが何らかの理由によってクローズされると（意図的なクローズや、通信障害によるクローズ）、そのコンシューマが予め起動した DCP の何れからのデータ送信も停止しなければならない。これは依然として確立された状態の SEMI E132 セッションがある DCP からコンシューマへのデータ送信に影響を与えてはならない。

7.2.3.3.1 ここで、複数のコンシューマを抱える DCP が 1 つでもある場合、装置は SEMI E134 DataCollectionPlan 状態モデルの遷移 5 が規定するアクションをとらなければならない。これらの DCP は ACTIVE なままであるが、装置は ACTIVE であるコンシューマのリストからそのコンシューマのセッションを削除し、削除した DCP の状態を INACTIVE にしなければならない。他のコンシューマに対してはデータの送信は継続して行われる。

7.2.3.3.2 同じく、コンシューマが 1 つだけある DCP が 1 つでもあると、装置は E134 DataCollectionPlan 状態モデルの遷移 8 が規定するアクションをとらなければならない。装置は ACTIVE であるコンシューマを並べたリストからそのコンシューマのセッションを削除し、その DCP の状態を INACTIVE に変更しなければならない。

7.2.3.4 DCP が休止状態から戻るとき、休止状態になる前にその DCP の起動を要求したコンシューマが少なくとも 1 つはあって、その SEMI E132 のセッションが健在である場合、装置はその DCP を ACTIVE 状態に遷移させ、その健在な SEMI E132 セッションがある各コンシューマへのデータ送信を再開しなければならない。

7.2.3.5 DCP が休止状態から戻るときに、健在な SEMI E132 セッションがあるコンシューマがない場合、装置はその DCP を INACTIVE 状態に遷移させなければならない。

7.2.4 SEMI E132.1 を使用した認証および権限付与

7.2.4.1 装置は、SEMI E134 の権限レベルを SEMI E132 の権限を以って実装しなければならない。

7.2.4.2 装置は、SEMI E134 の権限レベルを表 9 に従って実装しなければならない。ここに記載した権限のフォーマットは、urn:semi-org:priv.<priv-name>である。

表 9 SEMI E134 から SEMI E132 への権限のマッピング

SEMI E134 の権限レベル	SEMI E132.1 PrivilegeId のエントリ
ManageOnlyAuthoredDCPs	urn:semi-org:priv.ManageOnlyAuthoredDCPs
UseAnyDCP	urn:semi-org:priv.UseAnyDCP
ManageAnyDCP	urn:semi-org:priv.ManageAnyDCP

7.2.5 SEMI E125 のメタデータを使用した発生源、イベント、例外事象、パラメータの認識方法

7.2.5.1 発生源は意味のある SEMI E120 のロケータ値、および SEMI E120.1 LocatorType を使用して命名しなければならない。

7.2.5.2 SEMI E125 のイベントは SEMI E125 のイベントの「id」属性を使って区別されるが、その値は、SEMI E134 の DataCollectionPlan を構成する SEMI E134 の EventRequest の「eventId」属性の値が用いられる。

7.2.5.3 SEMI E125 の例外事象は SEMI E125 の例外事象の「id」属性を使って区別されるが、その値は、SEMI E134 DataCollectionPlan を構成する SEMI E134 の ExceptionRequest の「exceptionId」属性の値が用いられる。

7.2.5.4 SEMI E125 のパラメータは SEMI E125 のパラメータの「name」属性を使って区別されるが、その値は、SEMI E134 DataCollectionPlan を構成する SEMI E134 の ParameterRequest の「parameterName」属性の値が用いられる。

7.2.5.5 イベント、例外事象、またはパラメータが SEMI E39 で言う objType で定義される場合、それらのソースを定義する為に次の表記方法を使用すること。これらの定義は、（同一タイプの）インスタンスが確定していない場合の情報とインスタンスが確定している場合の情報を区別します。

表 10 ObjType イベント／例外／パラメータの表記方法

SEMI E125 項目	表記方法
ObjType が属性（未確定のインスタンス）	objType:<typeName>;objAttr:<parameter name>
ObjType がイベント（未確定のインスタンス）	objType:<typeName>;<eventId>
ObjType が例外事象（未確定のインスタンス）	objType:<typeName>;<exceptionId>
ObjType が属性（確定のインスタンス）	objInst:<typeName>;<objId>;<parameterName>
ObjType がイベント（確定のインスタンス）	objInst:<typeName>;<objId>;<eventId>
ObjType が例外事象（確定のインスタンス）	objInst:<typeName>;<objId>;<exceptionId>

7.2.5.6 上記の表に登場する <typeName> と <objId> は、次のフォーマットでなければならない。
「typeName」は、対象となるオブジェクトタイプに対する E39 による objType 名でなければならない。
「objId」は、そのインスタンスの SEMI E39 による objId 属性の値でなければならない。

7.2.5.7 確定していない場合の ObjType のインスタンス（表 10 の 1～3 行を参照）－対象となる例外事象、イベント、パラメータ（属性）発生させる SEMI E39 による具体的なオブジェクト id を DCP の作成者が事前に知らないあるいは知ることができない場合、この形態が使用される。これは、ほとんどの場合「過渡的な」SEMI E39 のオブジェクト、つまり動的に作成され、装置の動作中に破棄される SEMI E39 のオブジェクトの場合である。SEMI スタンダードに登場する SEMI E39 の過渡的オブジェクトの例には、SEMI E40 の PROCESSJOB ObjType, SEMI E87 の Carrier ObjType などがある。

7.2.5.7.1 「未確定のインスタンス」の表現形態をとる event/exceptionId で構成される Event/ExceptionRequest を含む DCP と言うことは、指定された ObjType で、SEMI E39 が適用できる任意のオブジェクトが、要求されたイベントや例外を生成する時には何時でも、装置は Event/ExceptionReport を生成しなければならないということを意味する。同じ ObjType に複数のインスタンスが存在し、それぞれが要求されたイベントまたは例外を生成する場合、装置は、SEMI E39 のそれぞれのオブジェクトに対して別々に Event/ExceptionReport を生成しなければならない。

7.2.5.7.2 「未確定のインスタンス」の表現形態をとる EventRequest は、同一 ObjType である SEMI E39 オブジェクトの属性を要求する「未確定のインスタンス」の表現形態をとる ParameterRequest を含めても構わない。そのような ParameterRequest では、ある SEMI E39 オブジェクトに関して EventReport に合わせて添えられる ParameterValue は、要求されたイベントを発生させたものと同一の SEMI E39 のオブジェクトに関するものでなければならない。例については、表 11 を参照すること。

7.2.5.7.3 `eventId` を「未確定のインスタンス」とする表現形態をとる `EventRequest` には「未確定のインスタンス」の表現形態をとる、異質の `ObjType` の属性に対する `ParameterRequest` を含んではならない。「未確定のインスタンス」の表現形態による `EventRequest` には、「確定したインスタンス」の表現形態による `ParameterRequest` を含んではならない(¶ 7.2.5.5 参照)。`eventId` を「未確定のインスタンス」とする表現形態を使ってはいない `EventRequest` には、「未確定のインスタンス」の表現形態をとる `ParameterRequest` を全く含んではならない。装置は、このような誤りがある `EventRequest` を含んだ DCP を拒否しなければならないが、それは、「DefinePlan」操作の際に、定義されている `InvalidEventRequest` タイプを使って行われる。

7.2.5.7.4 装置は、パラメータを「未確定のインスタンス」とする表現形態をとる `GetParameterValues` 要求を何れも拒否しなければならない。装置は、理由として「NoSuchParameter」が付いた「NoValue」を応答することにより、パラメータが定義されていないかのように、そのような要求を取り扱わなければならない。

7.2.5.7.5 `ObjType` の「未確定のインスタンス」を参照する構文を `TraceRequest` と共に使用することは有効ではない。「未確定のインスタンス」構文を含む `TraceRequest` は、`invalidContext` を「true」に設定した `InvalidParameterRequest` 構造を使用して拒否しなければならない。

7.2.5.8 確定している `ObjType` インスタンス (表10の4～6行を参照) – 対象となる例外事象、イベント、パラメータ (属性) を発生させる SEMI E39 のオブジェクト id が何であるかを DCP の作成者が事前に知っている場合、この表現形態が使用される。これは、ほとんどの場合「静的な」SEMI E39 のオブジェクト、つまり装置初期化中に一旦作成されると、装置動作中、あるいは場合によってはシャットダウン後であっても保持される SEMI E39 のオブジェクトの場合である。SEMI E39 の静的オブジェクトの例としては SEMI スタンドードには SEMI E90 の `SubstLoc` `ObjType`, SEMI E116 の `EPTTracker` `ObjType` などがある。

7.2.5.8.1 「確定のインスタンス」の表現形態をとる `event/exceptionId` で構成される `Event/ExceptionRequest` を含む DCP と言うことは、その確定している SEMI E39 オブジェクト (しかもそのオブジェクトのみ) が要求されたイベントや例外事象を発生する時に装置は `Event/ExceptionReport` を作成しなければならないということを意味する。同じ `ObjType` の複数のインスタンスが存在し、それぞれが要求されたイベントまたは例外事象を発生させる場合、装置は、要求された SEMI E39 オブジェクトのみに関しての `Event/ExceptionReport` を作成しなければならない。

7.2.5.8.2 「確定のインスタンス」の表現形態をとる `EventRequest` は、`EventRequest` に添えられる同一の SEMI E39 オブジェクトの属性を要求する「確定のインスタンス」の表現形態をとる `ParameterRequest` を含めても良い。そのような `ParameterRequest` では、そのオブジェクトの `EventReport` に合わせて添えられる `ParameterValue` は、要求されたイベントを発生したものと同一の SEMI E39 のオブジェクトに関するものでなければならない。例については、表 11 を参照すること。

7.2.5.8.3 「確定のインスタンス」の表現形態をとる `EventRequest` には、異なる `ObjType` の属性や、同じ `ObjType` の別のインスタンスに関する「確定のインスタンス」の表現形態をとる `ParameterRequest` を含んではならない。この表現形態をとる `EventRequest` は、「未確定のインスタンス」の表現形態をとる `ParameterRequest` を含んではならない。装置はこの様な誤りをがある `EventRequest` で構成される DCP を拒否しなければならないが、それは「DefinePlan」操作の際に、その操作に定義されている `InvalidEventRequest` タイプを使って行われる。

7.2.5.9 次の例は、タイプ「PROCESSJOB」の 2 つのオブジェクトおよびタイプ「SubstLoc」の 2 つのオブジェクトのものである。「PROCESSJOB」オブジェクトは、Process Job State の状態遷移#3 に伴うイベント (PJST3) を発生するものとする。「SubstLoc」オブジェクトは、Substrate Location State Model の状態遷移 #1 に伴うイベント (SLSMT1) を発生するものとする。最初の「PROCESSJOB」オブジェクトは「PJID1」のオブジェクト ID をとり、2 番目のオブジェクトは「PJID2」のオブジェクト ID をとるものとする。最初の「SubstLoc」オブジェクトは「Chamber1」に位置するオブジェクトの ID をとりし、2 番目のオブジェクトは「Chamber2」に位置するオブジェクトの ID をとるものとする。

表 11 ObjType を使ったイベント要求の例

イベント要求例	実行結果例
Event: objType:PROCESSJOB:PJST3 Parameter: objType:PROCESSJOB:ObjID	装置は、「PJID1」の Process Job State の状態遷移#3 のイベントが発生したときも、「PJID2」でそれが発生したときもイベント報告を送信する。 「PJID1」に対するイベント報告に添えられるパラメータ値は「PJID1」の「ObjID」の値である。 「PJID2」に対するイベント報告に添えられるパラメータ値は「PJID2」の「ObjID」の値である。
Event: objInst:SubstLoc:Chamber1:SLSMT1 Parameter: objInst:SubstLoc:Chamber1:SubstID	装置は、「Chamber1」で Substrate Location State Model の状態遷移#1 が発生したときにイベント報告を送信する。イベント報告に添えられるパラメータ値は「Chamber1」内に位置する「SubstID」の値である。
Event: objType:SubstLoc:SLSMT1 Parameter: objInst:SubstLoc:Chamber1:SubstID	イベントはタイプが「SubstLoc」である未確定のインスタンスに対するものであるが、パラメータは確定したインスタンス「Chamber1」に対するものなので、これはエラーとなる。
Event: objInst:SubstLoc:Chamber1:SLSMT1 Parameter: objType:SubstLoc:SubstID	イベントは確定しているインスタンス「Chamber1」に対するものであるが、パラメータはタイプが「SubstLoc」である未確定のインスタンスに対するものなので、これはエラーとなる。
Event: objType:PROCESSJOB:PJST3 Parameter: objType:SubstLoc:SubstID	イベントはタイプが「PROCESSJOB」である不特定のインスタンスに対するものであるが、パラメータはタイプが「SubstLoc」である不確定のインスタンスに対するものなので、これはエラーとなる。イベントとパラメータが同じ objType でなければならない。

7.2.6 共通エラー型の使用方法

7.2.6.1 次の表 12 に、SEMI E134 定義のエラーから上記の共通エラーコードへのマッピングを記載する。次の SEMI E134 のマッピングされたエラーコードは、共通構成要素規定で定義されている共通エラーコードへの追加である。SEMI E134 のエラーコードは、共通エラータイプ構造体内に「urn:semi-org:E134」の「発生源」名を指定して示さなければならない。

7.2.6.2 SEMI E134.1 は、SEMI E134 の Unauthorized Operation クラスを SEMI E132.1 の UnauthorizedOperation クラスとして実装し、SEMI E132.1 のエラーコード「6005」および共通構成要素のエラー構造体内の「urn:semi-org:E132」権限が付与されていない操作の発生源を返す。

表 12 SEMI E134 のエラーコード

SEMI E134 のエラー	共通エラーコード
Operation not authorized.	6000 (SEMI E132.1 の定義通り)
InvalidPlan.	8000
NoSuchPlan.	8001
DCPIsActive.	8002
DCPNotActive.	8003

7.2.7 XML スキーマおよび WSDL ファイル

7.2.7.1 DataCollectionManager インターフェイスについて本仕様で使う XML スキーマおよび WSDL は次の文書に記載されている。

表 13 XML スキーマ

ファイル名	E134-1-V0710-Schema.xsd
ターゲットネームスペース	urn:semi-org:xsd.E134.V0710.DCM
インポートされる／参照されるネームスペース	urn:semi-org:xsd.E138.V0709.ccs urn:semi-org:xsd.E132-1.V0310.auth http://www.w3.org/2001/XMLSchema
説明	このファイルは DataCollectionManager および DCPCConsumer の両方のインターフェイスによって使用される SEMI E134 のすべてのデータタイプとエレメントを定義する。

表 14 DataCollectionManager の PortType の定義

ファイル名	E134-1-V0710-DCMEquipmentPortType.wsdl
ターゲットネームスペース	urn:semi-org:ws.E134.V0710.DCMEquipmentPortType
インポートされる／参照されるネームスペース	urn:semi-org:xsd.E128.V0706.xms urn:semi-org:xsd.E134.V0710.DCM http://schemas.xmlsoap.org/wsdl/ http://www.w3.org/2001/XMLSchema
説明	このファイルは、SEMI E134 の XML スキーマで定義されるデータタイプに基づいて、DataCollectionManager 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 15 DataCollectionManager のバインディングの定義

ファイル名	E134-1-V0710-DCMEquipmentBinding.wsdl
ターゲットネームスペース	urn:semi-org:ws.E134.V0710DCMEquipmentBinding
インポートされる／参照されるネームスペース	urn:semi-org:ws.E134-V0710DCMEquipmentPortType http://schemas.xmlsoap.org/wsdl/soap12/ http://schemas.xmlsoap.org/wsdl/
説明	このファイルは、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。

7.2.7.2 これらの文書は、本仕様の一部であり、本文書に付属すべきものである。これらの文書の内容が、本仕様の中核部分を構成する。

7.2.8 WSDL のポート型 (port type) の概要

7.2.8.1 WSDL の DataCollectionManager portType の定義は、SEMI E134.1 の XML スキーマのデータ型を、SEMI E134、¶ 9.1 で DataCollectionManager 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、入出力も伴う集合体を使うものとして構成する。WSDL の portType 自体は、SEMI E134 のインターフェイスを基に命名され、WSDL portType の各操作には、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 E134 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 5 に、SEMI E134 の UML インターフェイスと WSDL portType の定義の関係を示す。WSDL のメッセージと操作の定義は、¶¶ 7.2.10～7.2.19 で説明する。

7.2.8.2 DataCollectionManager のポート型 (port type) の操作の詳細は、表 14 の wsdl ファイルに定義されている。

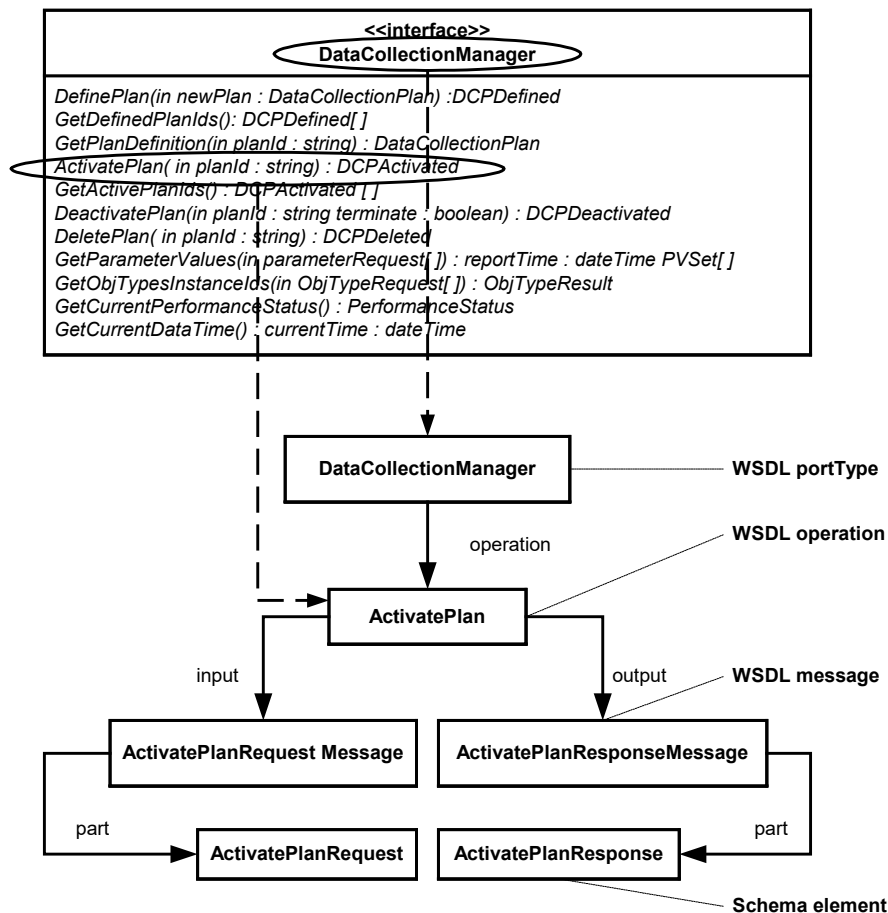


図 5
DataCollectionManager の WSDL portType へのマッピング

7.2.9 WSDL のバインディングの概要

7.2.9.1 WSDL の DataCollectionManager バインディングの定義は、対象となる portType (SEMI E134.1 の SOAP と HTTP) に使用するメッセージと伝送プロトコル、および使用されるインターフェイスの形態 (SEMI E134.1 の文書スタイル) を規定するものである。WSDL の各 portType 操作について、バインディングでは、使用するすべての SOAP ヘッダ、それらが必要か否か、その操作に使用する Content-Type ヘッダ「action」の値、使用する XML の表現を (正確に) 定義する。各 portType の操作のバインディングの定義は、¶¶ 7.2.10~7.2.19 で説明する。

7.2.9.2 DataCollectionManager のバインディングの操作の詳細は、表 15 の wsdl ファイルに定義されている。すべての SEMI E134 の操作で使われている SOAP 動作は、バインディングターゲットネームスペース、続いてコロンとオペレーションネームを使用する。(例, soapAction = <binding target namespace>:<operation-name>)

soapAction = “urn:semi-org:ws.E134.V0710.DCMEquipmentBinding:<operation-name>”

7.2.10 WSDL のサービスの概要

7.2.10.1 本仕様では、WSDL サービスの定義は行わない。WSDL サービスの定義は、クライアントが対象となるインターフェイスにアクセスできるエンドポイントを定義する。そのような情報は、装置が工場のネットワークに接続されるまでは決定できないので、本仕様の範囲外である。工場内でクライアントが SEMI E134 の装置サービスを見つける仕掛けを定めることは、ユーザの責任である。

7.2.11 DefinePlan

7.2.11.1 WSDL の操作のバインディング

7.2.11.1.1 バインディング wsdl ファイルの「DefinePlan」の操作のバインディングを参照のこと。便宜のため、次の表にバインディング操作に関する重要な情報を示す。

表 16 DefinePlan の操作のバインディング

SOAPAction	<binding target namespace>:DefinePlan
入力ヘッダ (WSDL メッセージ, 必須)	SyncHeaderMessage, 必須
出力ヘッダ (WSDL メッセージ, 必須)	SyncHeaderMessage, 必須

7.2.11.2 WSDL の操作

7.2.11.2.1 ポート型 (port type) wsdl ファイルでの「DefinePlan」 portType の操作の定義を参照のこと。便宜のため、次の表にポート型の操作に関する重要な情報を示す。

表 17 DefinePlan の portType の操作

入力メッセージ名	DefinePlanRequest
出力メッセージ名	DefinePlanResponse

7.2.11.3 WSDL のメッセージ

7.2.11.3.1 対応するポート型 (port type) wsdl ファイルの「SyncHeaderMessage」, 「DefinePlanRequest」, 「DefinePlanResponse」のメッセージの定義を参照すること。便宜のため、次の表に重要な情報を示す。

表 18 DataCollectionManagement のバインディング

メッセージ名 → スキーマエレメント名	SyncHeaderMessage → xms:SyncHeader DefinePlanRequest → dcm:DefinePlanRequestType DefinePlanResponse → dcm:DefinePlanResponseType
---------------------	--

7.2.11.4 XML スキーマの型

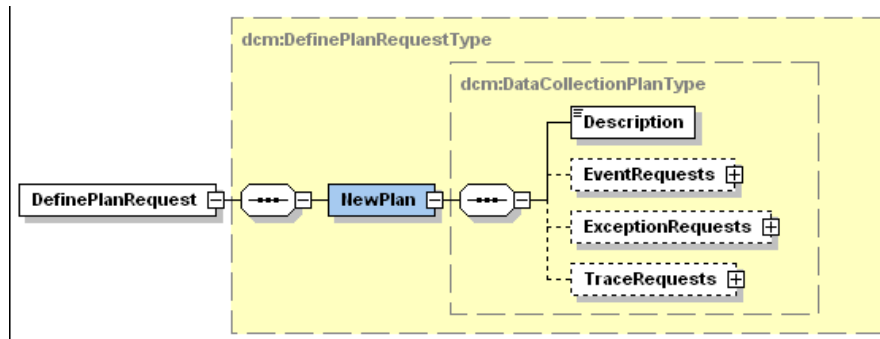


図 6
DefinePlanRequest

7.2.11.4.1 *DefinePlanRequest* — このグローバルエレメントは、DefinePlan 要求の SOAP ボディを表す一番外側の収容体としての役割を持つ。SEMI E134 が定義するように、この操作には、1つの入力引数、即ち、定義しようとする新たな DCP がある。この引数は、名称「NewPlan」とするエレメントの complexType 「DataCollectionPlanType」1個にモデル化される。

表 19 DefinePlan の操作の入力引数の変換表

引数名	形態	XML エレメント／属性 の区分	XML 名／型
newPlan	型が DataCollectionPlan である構造体データ	エレメント	newPlan: dcm:DataCollectionPlanType

7.2.11.4.2 *DataCollectionPlanType* — このグローバル complexType は、SEMI E134 の Data Collection Plan クラスの XML スキーマを表す。次の表で、UML クラスのこの XML スキーマ complexType へのマッピングを示す。

表 20 DataCollectionPlanType の変換表

属性／役割名	UML の型	XML エレメント／属性 (／参照) の区分	XML 名／型
name	テキスト	属性	name: xsd:string
id	テキスト	属性	id: xsd:string
intervalInMinutes	正の整数 ≥ 0	属性	intervalInMinutes: xsd:int
isPersistent	ブーリアン	属性	isPersistent: xsd:boolean
description	テキスト	エレメント	description: xsd:string
eventRequests	型が EventRequest である構造体データのリスト	エレメント	eventRequest: dcm:EventRequestType
exceptionRequests	型が ExceptionRequest である構造体データのリスト	エレメント	exceptionRequests: dcm:ExceptionRequestType
traceRequests	型が TraceRequest である構造体データのリスト	エレメント	traceRequests: dcm:TraceRequestType