

SEMI E134.1-0709

データ収集管理(DCM)の SOAP バインディングに関する仕様

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 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 2005 年 3 月発行, 前版は 2008 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 参照スタンダード
 - 4.1 SEMI スタンダード
 - 4.2 OMG スタンダード
 - 4.3 W3C スタンダード
 - 4.4 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 DCPCConsumer

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
- 図 7 DataCollectionPlanType
- 図 8 EventRequestType
- 図 9 TraceRequestType
- 図 10 DefinePlanResponseType
- 図 11 InvalidPlanType
- 図 12 InvalidEventRequestType
- 図 13 InvalidTraceRequestType

- ☒ 14 GetDefinedPlanIdsRequest
- ☒ 15 GetDefinedPlanIdsResponse
- ☒ 16 GetPlanDefinitionRequest
- ☒ 17 GetPlanDefinitionResponse
- ☒ 18 ActivatePlanRequest
- ☒ 19 ActivatePlanResponse
- ☒ 20 GetActivePlanIdsRequest
- ☒ 21 GetActivePlanIdsResponse
- ☒ 22 DeactivatePlanRequest
- ☒ 23 DeactivatePlanResponse
- ☒ 24 DeletePlanRequest
- ☒ 25 DeletePlanResponse
- ☒ 26 GetParameterValuesRequest
- ☒ 27 GetParameterValuesResponse
- ☒ 28 GetObjTypeInstanceIdsRequest
- ☒ 29 GetObjTypeInstanceIdsResponse
- ☒ 30 GetCurrentPerformanceStatusRequest
- ☒ 31 GetCurrentPerformanceStatusResponse
- ☒ 32 GetCurrentDateTimeRequest
- ☒ 33 GetCurrentDateTimeResponse
- ☒ 34 NewDataNotification
- ☒ 35 EventReportType
- ☒ 36 ExceptionReportType
- ☒ 37 CollectedDataType
- ☒ 38 PerformanceWarningNotification
- ☒ 39 PerformanceRestoredNotification
- ☒ 40 DCPDeactivationNotification
- ☒ 41 DCPHibernationNotification
- ☒ 42 DCPDefinedNotification
- ☒ 43 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 port type 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 operation のバインディング
- 表 17 DefinePlan PortType の operation

表 18	DataCollectionManagement のバインディング
表 19	DefinePlan の operation 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	ExceptionTriggerType の変換表
表 27	ParameterRequestType の変換表
表 28	DefinePlan の operation の出力引数の変換表
表 29	DCPDefinedType の変換表
表 30	InvalidPlanType の変換表
表 31	InvalidEventRequestType の変換表
表 32	InvalidExceptionRequestType の変換表
表 33	InvalidParameterRequestType の変換表
表 34	InvalidTraceRequestType の変換表
表 35	InvalidTriggerType の変換表
表 36	InvalidIntervalType の変換表
表 37	InvalidCycleType の変換表
表 38	GetDefinedPlanIds の operation のバインディング
表 39	GetDefinedPlanIds の PortType の operation
表 40	DataCollectionManagement のバインディング
表 41	GetDefinedPlanIds の operation の出力引数の変換表
表 42	GetPlanDefinition の operation のバインディング
表 43	GetPlanDefinition の PortType の operation
表 44	DataCollectionManagement のバインディング
表 45	GetPlanDefinition の Operation の入力引数の変換表
表 46	GetPlanDefinition の Operation の出力引数の変換表
表 47	NoSuchPlanType の変換表
表 48	ActivatePlan の operation のバインディング
表 49	ActivatePlan の PortType の operation
表 50	DataCollectionManagement のバインディング
表 51	ActivatePlan の operation の入力引数の変換表
表 52	GetDefinedPlanIds の operation の出力引数の変換表
表 53	DCPActivatedType の変換表
表 54	GetActivePlanIds の operation のバインディング
表 55	GetActivePlanIds の PortType の operation
表 56	DataCollectionManagement のバインディング
表 57	ActivatePlan の operation の出力引数の変換表
表 58	DeactivatePlan の operation のバインディング
表 59	DeactivatePlan の PortType の operation
表 60	DataCollectionManagement のバインディング
表 61	DeactivatePlan の operation の入力引数の変換表
表 62	DeactivatePlan の operation の出力引数の変換表
表 63	DeactivatePlanResponseType の変換表
表 64	DCPNotActiveType の変換表
表 65	DeletePlan の operation のバインディング
表 66	DeletePlan の PortType の operation
表 67	DataCollectionManagement のバインディング

表 68 DeletePlan の operation の入力引数の変換表
表 69 DeletePlan の operation の出力引数の変換表
表 70 DCPDeletedType の変換表
表 71 GetParameterValues の operation のバインディング
表 72 GetParameterValues の PortType の operation
表 73 DataCollectionManagement のバインディング
表 74 GetParameterValues の operation の入力引数の変換表
表 75 GetParameterValues の operation の出力引数の変換表
表 76 GetObjTypeInstanceIds の operation のバインディング
表 77 GetObjTypeInstanceIds の portType の operation
表 78 DataCollectionManagement のバインディング
表 79 GetObjTypeInstanceIds の operation の入力引数の変換表
表 80 ObjTypeRequestType の変換表
表 81 GetObjTypeInstanceIds の operation の出力引数の変換表
表 82 ObjTypeResultType の変換表
表 83 ObjTypeInstanceType の変換表
表 84 ObjTypeRequestErrorType の変換表
表 85 GetCurrentPerformanceStatus の operation のバインディング
表 86 GetCurrentPerformanceStatus の PortType の operation
表 87 DataCollectionManagement のバインディング
表 88 GetCurrentPerformanceStatus の operation の出力引数の変換表
表 89 PerformanceStatusType の変換表
表 90 ActiveJobType の変換表
表 91 JobTypeEnumType に関して選択可能なジョブの種類を示す表
表 92 AffectedSystemType の変換表
表 93 GetCurrentDataTime の operation のバインディング
表 94 GetCurrentDataTime の PortType の operation
表 95 DataCollectionReportType のバインディング
表 96 GetCurrentDataTime の operation の出力引数の変換表
表 97 XML スキーマ
表 98 DCPConsumer の PortTypeh の定義
表 99 DCPConsumer のバインディングの定義
表 100 NewData の operation のバインディング
表 101 NewData の PortType の operation
表 102 DCPConsumer のバインディング
表 103 NewDataNotification の operation の入力引数の変換表
表 104 DataCollectionReportType の変換表
表 105 EventReportType の変換表
表 106 ExceptionReportType の変換表
表 107 TraceReportType の変換表
表 108 CollectedDataType の変換表
表 109 PerformanceWarning の operation のバインディング
表 110 PerformanceWarning の PortType の operation
表 111 DCPConsumer のバインディング
表 112 PerformanceWarning 操作の入力引数の変換表
表 113 PerformanceRestored 操作のバインディング
表 114 PerformanceRestored の PortType の operation
表 115 DCPConsumer のバインディング
表 116 PerformanceRestored の operation の入力引数の変換表
表 117 DCPDeactivation の operation のバインディング

表 118	DCPDeactivation の portType の operation
表 119	DCPConsumer のバインディング
表 120	DCPDeactivation の operation の入力引数の変換表
表 121	DCPHibernation の operation のバインディング
表 122	DCPHibernation の PortType の operation
表 123	DCPConsumer のバインディング
表 124	DCPHibernation の operation の入力引数の変換表
表 125	DCPHibernatedType の変換表
表 126	DCPDefined の operation のバインディング
表 127	DCPDefined の PortType の operation
表 128	DCPConsumer のバインディング
表 129	DCPDefined の operation の入力引数の変換表
表 130	DCPDeleted の operation のバインディング
表 131	DCPDeleted の PortType の operation
表 132	DCPConsumer のバインディング
表 133	DCPDeleted の operation 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 1.1 プロトコルに対する実装用マッピングを定めることである。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のポートタイプとバインディングの定義についても説明する。

2 範囲

2.1 仕様範囲

2.1.1 本書の範囲は、XMLスキーマによるDCMモデルの忠実な表現、対応するWSDLのポートタイプやバインディングである。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。

2.1.2 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。

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: このスタンダードはその使用に関連した全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

4.1 SEMI スタンダード

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

SEMI E121 — Guide for Style & 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

Unified Modeling Language (UML) Specification, Version 1.4, OMG Specification 01-09-67;
http://www.omg.org/technology/documents/modeling_spec_catalog.htm

4.3 W3C スタンダード²

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

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

¹ Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, U.S.A., Telephone: 781.444.0404; Fax: 781.444.0320; Website: <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; Website: <http://www.w3.org>

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

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

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

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

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/>

4.4 その他のスタンダード

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

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

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ウム

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 (*UML (Unified Modeling Language)*) — 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 (*XML (eXtensible Markup Language)*) — 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表現物の転送効率を改善するために、SEMI E125 にある UML によるクラスの属性は XML での属性として一般に表現する。この変換方法の例外は、XML 属性として表現できないデータ構造や、その他の構成様式をもつ UML での属性であり、構文解析されない文字データ(CDATA)を含む必要性が起り得る文字の内容値もこの中に入る。
- 構成関係を伴う関連は、XML 要素 1 個(または、指定された多重度が 1 より大きい場合は、複数のそのような要素)にマッピングする。

- 集約は、要素として内容に収めるか、あるいは集約の対象となるものの型の属性のうち要素をユニークに個体識別する属性1個以上からなる配列としてモデル化される。
- 継承は複合型の **extension** 要素を使用してマッピングする。UML モデル内で抽象ベースクラスへの参照が発生する場合には、型の置換による結果的に勝手な拡張がインスタンスドキュメントで起きないように制限するために、派生する可能性がある型各々の **choice** による合成が使用される。

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記載する。UMLでのクラスにおいて定義された操作は、WSDLのoperationに変換される。

6.1.5 変換表の列のヘッダの説明

- 属性名／役割名 — 属性の場合は、その属性名をこの欄に記入する。関連(集約や構成)の場合は、UML図の役割名をこの欄に記入する。構成関係はしばしば役割名が割り当てられない。その場合は「特記せず」と記載する。
- UML 名／型 — 属性の場合は、UML 属性のデータ型をこの欄に記入する。関連の場合は、関連の種類がこの欄に記入する。あり得る関連の種類は「構成」、「集約」、一般「関連」である。UML では、これらの 3 種類の関連を定義している。
- XML 要素／属性の区分 — UML 属性または関連を表現するためにとられる XML 構造の種類をリスト表記する。
- XML 名／型 — 変換結果として得られる XML 構造の名前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 スキーマで新たに定義する名前付き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 要素／属性の区分	XML 名／型
friend	association	element	Friend: HumanReferenceArray
employees	aggregation	element	Employees: HumanArray
family	composition	element	Family: HumanArray
name	string	element	Name: xs: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からXMLスキーマの型への操作に対して定められる変換方法は、表 2 に示す表形式を使用して説明資料とする。対象とする操作の取り得る入／出力引数、各々に1行を充てるものとする。

表 2 操作入力／出力引数の変換表の例

引数名	形態	XML 要素／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>	<引数が要素か属性の何れとしてモデル化されるか>	<ネームスペースで有効な属性／要素の名称>

6.2 XMLスキーマとWSDLファイルの説明方法

6.2.1 本仕様に共通に登場する構成要素であるXMLスキーマとWSDLファイルが本書に付随するものである。その各々に関する説明資料は、表 3 に示す表形式に従って記載するものとする。

表 3 スキーマ／WSDL 文書記述表の例

ファイル名	<xsd or .wsdl の拡張子を取るファイル名>
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>
インポートされる／参照されるネームスペース	<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>

ファイル名	<xsd or .wsdl の拡張子を取るファイル名>
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 XMLスキーマ図による説明

6.3.1 本文書は、含まれるXMLスキーマのデータ型、および要素の定義を図を以って表現する。XMLスキーマの標準とされる図示方法存在しないが、さまざまな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 に、本文書のここ以降で使用されるスキーマの図解の基本事項を説明するために、XMLスキーマ図の例を示す。

6.3.2 本図は矩形形状の箱がXML要素の定義を示す。オーナー要素は左側に置かれ、オーナー要素内に格納される要素を右側に置くものとする。この図の例では、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。意味を付加する記号(8 角の箱)は、シーケンスや選択を示す。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、順序付きの要素列で、要素の有無は任意。(sequence)
	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枝を示し、存在は必須である。1 つ以上 2 つ以下の右側の要素(左図の場合)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要素を示す。
	他の箇所所で定義されているグループや要素への参照を示す。

6.3.3 実線を使用した図形は必須の要素を示し、破線は任意の要素を示す。右下隅の数字は、個数を示す。指定なき場合の個数は 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ことを示す。箱内がプラスの記号「+」である場合は、その記号の右側の詳細部分は全く明らかにされていないことを示す。下図の例では、詳細は明らかにされている。

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、図には示さなくても良い。オブジェクト指向方法論に関する継承は、ここでは普通、XMLでは型のextension要素によって表現する。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。

- Child1and2Type は、2 つの項目 Child2 と Child1 の順序付き要素列である。
- Child1 は、Child1a, 1 つまたは複数の Child1b, (有無は任意の)Child1c の有無が任意の順序付きの要素列を格納する。
- Child2 の各々は、次の Child2a, Child2b, Child2c からの 1 つまたは 2 つを選んで格納する。
- Child3 は、Child3a と Child3b からなる順序が任意の要素列を格納する。

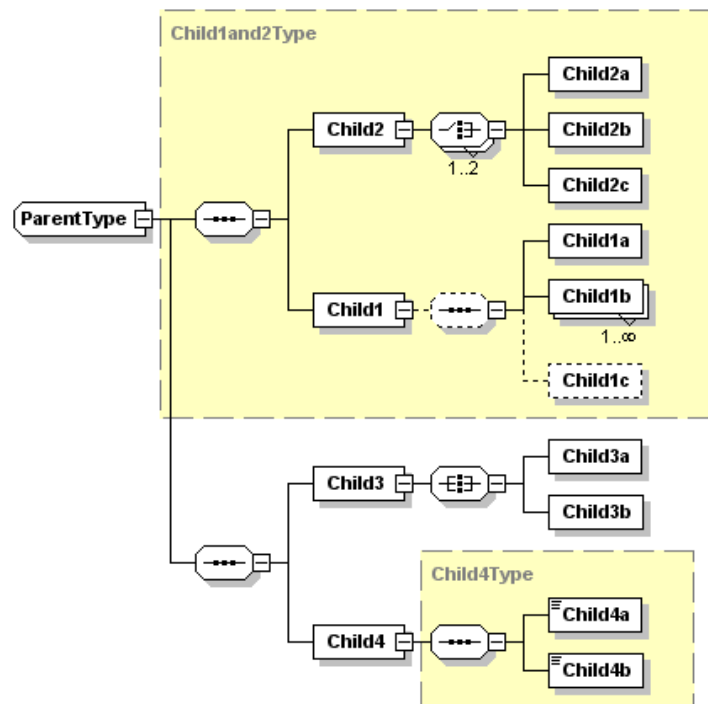


図 1
XML スキーマ図の例

6.4 XMLスキーマの例

6.4.1 図 1 に示した例の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

※: 6.3.5 の説明は、図 2 とは一致していません。
次版にて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、抽象規定中で定義される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。次の表は注目するUMLインターフェイスクラスについてWSDLのport typeとバインディングの定義の、記述言語表現に使用される表現方法を示す。

表 5 インターフェイスの WSDL の port type への変換表の例

クラス名	<抽象規定での UML インターフェイス名>
WSDL の port type 名	<WSDL 内で使用される portType 名>
SEMI E125 の操作 → WSDL の操作	<UML インターフェイスからの操作名> → <WSDL の portType の operation 名>

表 6 インターフェイスの WSDL バインディングへの変換表の例

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 のバインディングスタイル	<RPC または文書形式>
SOAP のトランスポート	<URI を定義するトランスポート>

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operation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	<HTTP ヘッダの SOAPAction の値で、該当操作に使われるもの>
入力ヘッダ (WSDL メッセージ, 必要性)	<入力ヘッダおよびヘッダが必要かどうかを示す設定する WSDL メッセージ名, ならびに、その入力ヘッダの必要性>
出力ヘッダ (WSDL メッセージ, 必要性)	<SOAPAction WSDL メッセージ名, ならびに、その出力ヘッダの必要性>

表 8 PortType の操作表の例

入力メッセージ名	<WSDLPort Type の operation で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	<WSDLPort Type の operation で出力として使用される WSDL メッセージ名>

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

7.1.1 SEMI E134 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。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して名前が付けられる。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のエンコードスタイルおよびHTTPヘッダの使用方法を定義する目的で、各portTypeのWSDLバインディングの定義が用いられる。SEMI E134 のすべてのWSDLインターフェイスでは、グローバルな要素の定義を使ってXMLスキーマファイル内で定義されているSOAPヘッダとボディの完全形で、文書／文字形式のエンコーディングを使用する。

7.1.2 図 3 はWSDLバインディング、portTypeの定義の関係、および各インターフェイスに使用されるXMLスキーマの型を示す。portTypeの定義は各operationで使うXMLスキーマの型を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 専用に対して定義されるXMLスキーマおよびWSDLファイルを示したものである。

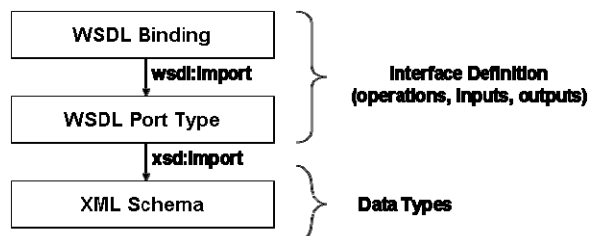


図 3

XML スキーマおよび WSDL ファイルの構成

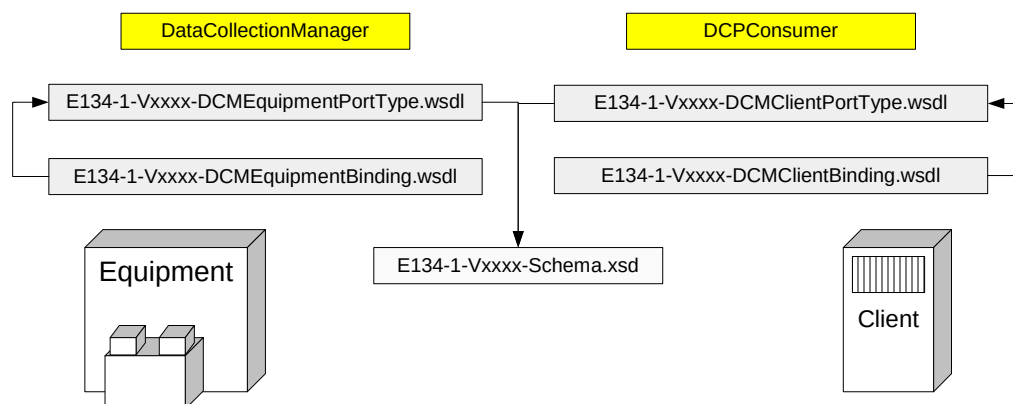


図 4

SEMI E134 の WSDL および XML スキーマファイル

7.2 DataCollectionManager

7.2.1 SEMI E132 のセッションの適用

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ポイン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操作はすべて、SEMI E132.1 で指定されているようにSSL認証を有効として、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ことになる。SEMI E132.1 に関して有効となると、装置がサポートするすべてのSEMI E134 DataCollectionManager操作について、クライアントとサーバの相互のSSL認証を要求する必要がある。装置は、HTTPSトランスポート経由のもの以外は何れの操作へのアクセスをも提供してはならない。

7.2.2 装置と通信するために、装置やクライアントに証明機構をインストールすることが実際的あるいは実現可能ではないような統合環境、開発環境、テスト環境もあるであろう。そのような環境についてのみ、現実的な配慮として、装置は、SEMI E132.1 の仕様にあるように、装置構成情報を操作して、SEMI E134 の規定するDataCollectionManagerインターフェイスに関して、SSL認証機能をユーザが無効にできるような用意が必要である。SSL認証を無効にすると、ネットワーク経由で装置にアクセスするアプリケーションの何れもが、SEMI E134 のデータへの無制限のアクセスを与えることになるということをユーザは認識しておかなければならない。

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エンベロープヘッダ内の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要素で示す必要がある。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、エラーコードが「6005」でソースが「urn:semi-org:E132」のSEMIの共通エラーデータを示す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。要求に有効ではないSEMI E128 の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、装置は何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、装置は何れの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、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。

7.2.5.8 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スタンダードには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じの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

SEMI E134 のエラー	共通エラーコード
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-V0709-Schema.xsd
ターゲットネームスペース	urn:semi-org:xsd.E134.V0709.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.V1107.auth http://www.w3.org/2001/XMLSchema
説明	このファイルは DataCollectionManager および DCPConsumer 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要素を定義する。

表 14 DataCollectionManager の PortType の定義

ファイル名	E134-1-V0709-DCMEquipmentPortType.wsdl
ターゲットネームスペース	urn:semi-org:ws.E134.V0709.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.V0709.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-V0709-DCMEquipmentBinding.wsdl
ターゲットネームスペース	urn:semi-org:ws.E134.V0709DCMEquipmentBinding
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-V0709DCMEquipmentPortType http://schemas.xmlsoap.org/wsdl/soap/ 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、本仕様の核心部分を構成する。

7.2.8 WSDL の port type の概要

7.2.8.1 WSDLのDataCollectionManager 用のportTypeの定義は、SEMI E134.1 のXMLスキーマのデータ型を、SEMI E134, 9.1, のDataCollectionManagerインターフェイスとして定義したUMLで表現される操作に対応して定義された操作とその入出力集合として体系づける。WSDLのportType自体には、SEMI E134 のインターフェイスを基に命名され、WSDL portTypeの各operationには、インターフェイスについ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operationは、SEMI E134 で定義されるUMLでの操作に関しての入力および出力(要求と応答)に対応した 2 つのWSDLメッセージの定義として構成される。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operationの定義は、7.2.10～7.2.19 で説明する。

7.2.8.2 DataCollectionManagerのport typeのoperationの詳細は、表 14 のwsdlファイルに定義されている。

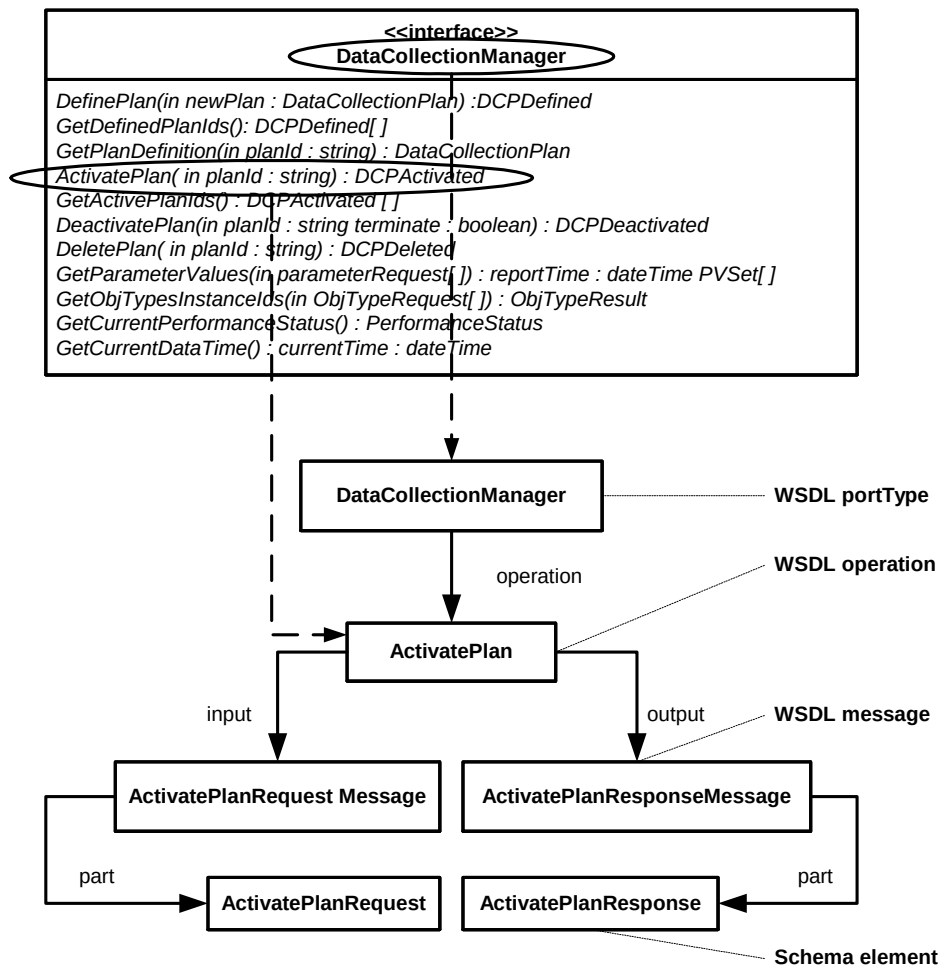


図 5
DataCollectionManager の WSDL portType へのマッピング

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、該当する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スタイル (SEMI E134.1 の文書スタイル)を指定して行う。WSDLの各portTypeのoperationについて、バインディングでは、使用されるすべてのSOAPヘッダ、その必要性、その操作に使用するHTTP SOAPActionヘッダの値、使用するXMLエンコーディング (文字)を定義する。各portTypeのoperation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 Action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, SOAP Action = <binding target namespace>:<operation-name>)

SOAP Action = urn:semi-org:ws.E134.V0709.DCMEquipmentBinding:<operation-name>

7.2.10 WSDLのサービスの概要

7.2.10.1 本仕様は、WSDLのサービスの定義は行わない。WSDLのサービスの定義は、クライアントは利用可能なインターフェイスにはどこにアクセスすれば良いかを示すアドレスを定義する。そのような情報は、装置が工場の

ネットワークにインストールされるまでは決定できないので、本仕様の取り扱い範囲外である。ユーザの責任で、クライアントが工場内でSEMI E134 の定める装置サービスを見つける仕掛けを決めるものである。

7.2.11 DefinePlan

7.2.11.1 WSDLのoperation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すること。便利なように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 operation のバインディング

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

7.2.11.2 WSDLのoperation

7.2.11.2.1 port type wsdlファイルでの “DefinePlan” のportTypeのoperationの定義を参照すること。便利なように次の表にport typeの操作に関する重要な情報を示す。

表 17 DefinePlan の portType の operation

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。便利なように、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 → スキーマ要素名	SyncHeaderMessage → xmlns:SyncHeader DefinePlanRequest → dcm:DefinePlanRequestType DefinePlanResponse → dcm:DefinePlanResponseType
------------------	--

7.2.11.4 XMLスキーマの型

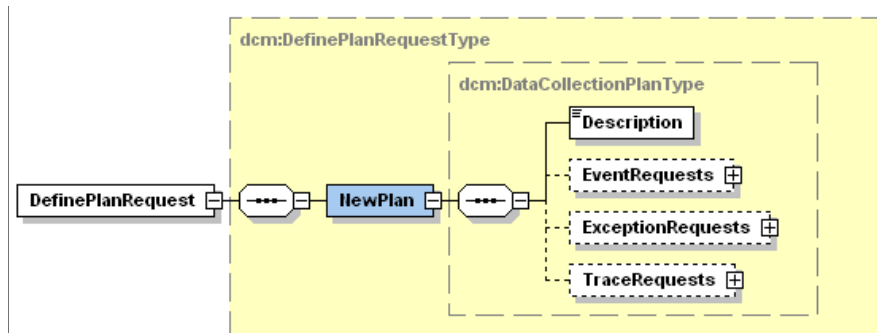


図 6

DefinePlanRequest

7.2.11.4.1 DefinePlanRequest - このグローバル要素は、DefinePlan要求のSOAPのボディを表現する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、複合型 “DataCollectionPlanType” の “NewPlan” という名前の要素 1 個にモデル化される。

表 19 DefinePlan の operation の入力引数の変換表

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

7.2.11.4.2 *DataCollectionPlanType* — このグローバルな複合型は、SEMI E134 のData Collection PlanクラスのXMLスキーマ表現である。次の表にUMLのクラスのこのXMLスキーマへの複合型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 要素／属性 (または参照) の区分	XML 名／型
name	Text	Attribute	name: xsd:string
id	Text	Attribute	id: xsd:string
intervalInMinutes	正の整数>=0	Attribute	intervalInMinutes: xsd:int
isPersistent	Boolean	Attribute	isPersistent: xsd:boolean
description	Text	Element	description: xsd:string
eventRequests	型が EventRequest である構造体データのリスト	Element	eventRequest: dcm:EventRequestType
exceptionRequests	型が ExceptionRequest である構造体データのリスト	Element	exceptionRequests: dcm:ExceptionRequestType
traceRequests	型が TraceRequest である構造体データのリスト	Element	traceRequests: dcm:TraceRequestType

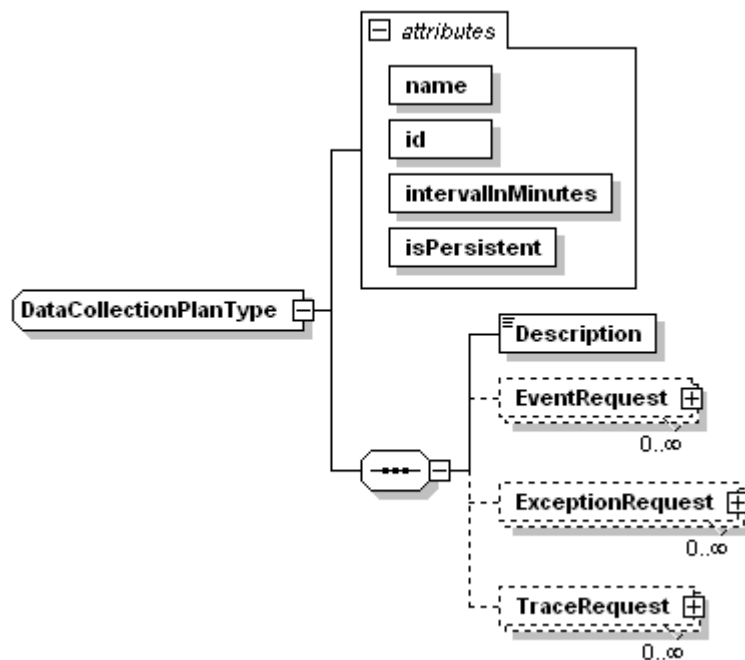


図 7

DataCollectionPlanType

7.2.11.4.3 *EventRequestType* — このグローバルな複合型は、SEMI E134 のEvent RequestクラスのXMLスキーマ表現である。次の表にUMLのクラスのこのXMLスキーマの複合型へのマッピングを示す。