

SEMI E125.1-0414

EQUIPMENT SELF DESCRIPTION(EqSD)의 SOAP 바인딩 규격

본 표준은 Information & Control Global Technical Committee 의 기술 승인을 받았다. 본 에디션은 Global Audits and Reviews Subcommittee 가 2013 년 12 월 23 일 간행을 승인했다. 2014 년 4 월부터 www.semiviews.org 및 www.semi.org 에서 이용 가능하다. 초판은 2005 년 3 월에 간행되었으며 2010 년 7 월에 이전 에디션이 발행되었다.

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

주의: 본 한글본은 참고용으로, 영문원본과 한글본간에 차이가 발생할 경우, 영문원본을 우선으로 해야합니다. 본 한글본에 대한 문의사항이 있는 경우 krstandards@semi.org 로 문의바랍니다.

목차

1 목적 (Purpose).....	6
1.1 SOAP 구현 매핑.....	6
2 범위 (Scope).....	6
3 제한 (Limitations).....	6
3.1 없음.....	6
4 참조 표준 및 문서 (Referenced Standards and Documents).....	6
4.1 SEMI 표준 및 안전 가이드라인.....	6
4.2 OMG 표준.....	6
4.3 W3C 표준.....	6
4.4 기타 표준.....	7
5 용어 (Terminology).....	7
5.1 약자 및 약어.....	7
5.2 정의.....	7
6 규범 (Conventions).....	8
6.1 UML 을 XML 스키마로 변환.....	8
6.2 XML 스키마 및 WSDL 파일 문서화.....	9
6.3 다이어그램을 이용한 XML 스키마 문서화.....	9
6.4 XML 스키마 샘플.....	11
6.5 UML 을 WSDL 로 변환.....	12
7 SEMI E125 UML 을 XML 스키마 및 WSDL 에 매핑 (Mapping of SEMI E125 UML to XML Schema and WSDL).....	13
7.1 WSDL 구조.....	13
7.2 EquipmentMetadataManager.....	14

7.3 MetadataClient	49
8 메시지 전송 (Message Transport)	52
RELATED INFORMATION 1	53

그림 목차

그림 1 XML 스키마 다이어그램의 예	11
그림 2 샘플 XML	12
그림 3 XML 스키마 및 WSDL 파일 구조	13
그림 4 SEMI E125 의 WSDL 및 XML 스키마 파일	14
그림 5 EquipmentMetadataManager 를 WSDL portType 에 매핑	16
그림 6 전역 요소 GetUnitsRequest	18
그림 7 전역 요소 GetUnitsResponse	19
그림 8 전역 요소 GetTypeDefinitionsRequest	20
그림 9 전역 요소 GetTypeDefinitionsResponse	21
그림 10 전역 요소 GetStateMachinesRequest	22
그림 11 전역 요소 GetStateMachinesResponse	23
그림 12 complexType 전역 요소 StateMachineType	24
그림 13 전역 요소 GetSEMIObjTypesRequest	27
그림 14 전역 요소 GetSEMIObjTypesResponse	28
그림 15 complexType 전역 요소 StateMachineInstanceType	30
그림 16 전역 요소 GetExceptionsRequest	33
그림 17 전역 요소 GetExceptionsResponse	34
그림 18 전역 요소 GetEquipmentStructureRequest	35
그림 19 전역 요소 GetEquipmentStructureResponse	36
그림 20 전역 요소 GetEquipmentNodeDescriptionsRequest	37
그림 21 전역 요소 GetEquipmentNodeDescriptionsResponse	38
그림 22 complexType 전역 요소 EquipmentNodeDescriptionType	39
그림 23 complexType 전역 요소 ParameterType	42
그림 24 GetLatestRevisionRequest	46
그림 25 전역 요소 GetLatestRevisionResponse	46
그림 26 전역 요소 NotifyOnRevisionsRequest	48
그림 27 전역 요소 NotifyOnRevisionsResponse	48
그림 28 전역 요소 MetadataRevisedNotification	52

표 목차

표 1 변환 테이블의 예	9
표 2 연산 입/출력 인수 변환 테이블의 예	9

표 3 스키마/WSDL 문서 설명 테이블의 예	9
표 4Altova XMLSPY 스키마 다이어그램 심볼	10
표 5 인터페이스 WSDL 포트 유형 테이블의 예	12
표 6 인터페이스 WSDL 바인딩 테이블 예	12
표 7 연산 바인딩 테이블 예	12
표 8 PortType 연산 테이블 예	12
표 9 XML 스키마	14
표 10 EquipmentMetadataManager PortType 정의	15
표 11 EquipmentMetadataManager 바인딩 정의	15
표 12 EquipmentMetadataManager 포트 유형	16
표 13 EquipmentMetadataManager 바인딩	17
표 14 GetUnits 연산 바인딩	17
표 15 GetUnits PortType 연산	18
표 16 EquipmentMetadataManager 바인딩	18
표 17 GetUnits 연산의 출력 인수 변환 테이블	19
표 18 GetTypeDefinitions 연산 바인딩	19
표 19 GetUnits PortType 연산	20
표 20 EquipmentMetadataManager 바인딩	20
표 21 GetTypeDefinitions 연산의 출력 인수 변환 테이블	21
표 22 GetStateMachines 연산 바인딩	21
표 23 GetStateMachines PortType 연산	22
표 24 EquipmentMetadataManager 바인딩	22
표 25 GetStateMachines 연산의 출력 인수 변환 테이블	25
표 26 SEMI E125 StateMachine → StateMachineType 변환 테이블	25
표 27 SEMI E125 State → StateType 변환 테이블	25
표 28 SEMI E125 Transition → TransitionType 변환 테이블	26
표 29 SEMI E125 Event → EventType 변환 테이블	26
표 30 GetSEMIObjTypes 연산 바인딩	26
표 31 GetSEMIObjTypes PortType 연산	27
표 32 EquipmentMetadataManager 바인딩	27
표 33 GetSEMIObjTypes 연산의 출력 인수 변환 테이블	28
표 34 SEMI E125 SEMIObjType → SEMIObjTypeType 변환 테이블	29
표 35 SEMI E125 StateMachineInstance → StateMachineInstanceType 변환 테이블	30
표 36 파라미터 참조 예	31
표 37 SEMI E125 EventToParameterMap → EventToParameterMapType 변환 테이블	32
표 38 GetExceptions 연산 바인딩	32
표 39 GetExceptions PortType 연산	32

표 40 EquipmentMetadataManager 바인딩	33
표 41 GetExceptions 연산의 출력 인수 변환 테이블	34
표 42 SEMI E125 Exception → ExceptionType 변환 테이블	34
표 43 GetEquipmentStructure 연산 바인딩	34
표 44 GetEquipmentStructure PortType 연산	35
표 45 EquipmentMetadataManager 바인딩	35
표 46 GetEquipmentStructure 연산의 출력 인수 변환 테이블	36
표 47 GetEquipmentNodeDescriptions 연산 바인딩	36
표 48 GetEquipmentNodeDescriptions PortType 연산	37
표 49 EquipmentMetadataManager 바인딩	37
표 50 GetEquipmentNodeDescriptions 연산의 입력 인수 변환 테이블	37
표 51 GetEquipmentNodeDescriptions 연산의 출력 인수 변환 테이블	38
표 52 SEMI E125 EquipmentNodeDescription → EquipmentNodeDescriptionType 변환 테이블	40
표 53 SEMI E125 SimpleEvent → SimpleEventType 변환 테이블	40
표 54 SEMI E125 ExceptionReference → ExceptionReferenceType 변환 테이블	41
표 55 SEMI E125 SECSVarRef → SECSVarRefType 변환 테이블	42
표 56 SEMI E125 Parameter → ParameterType 변환 테이블	42
표 57 SEMI E125 Constraint → ConstraintType 변환 테이블	43
표 58 SEMI E125 AssociatedParameter → AssociatedParameterType 변환 테이블	43
표 59 SEMI E125 Units → UnitType 변환	43
표 60 SEMI E125 UnitConfigArray → UnitConfigArrayType 변환	44
표 61 SEMI E125 UnitConfig → UnitConfigType 변환	44
표 62 SEMI E125 ParameterClassification → ParameterClassificationType 변환 테이블	44
표 63 SEMI E125 ReadWrite → ReadWriteType 변환 테이블	44
표 64 GetLatestRevision 연산 바인딩	45
표 65 GetLatestRevision PortType 연산	45
표 66 EquipmentMetadataManager 바인딩	45
표 67 GetLatestRevision 연산의 출력 인수 변환 테이블	46
표 68 NotifyOnRevisions 연산 바인딩	47
표 69 NotifyOnRevisions PortType 연산	47
표 70 EquipmentMetadataManager 바인딩	48
표 71 NotifyOnRevisions 연산의 입력 인수 변환 테이블	48
표 72 GetUnits 연산의 출력 인수 변환 테이블	48
표 73 XML 스키마	49
표 74 MetadataClient PortType 정의	49
표 75 MetadataClient 바인딩 정의	50
표 76 MetadataClient 포트 유형	50



표 77 MetadataClient 바인딩.....	51
표 78 MetadataRevised 연산 바인딩.....	51
표 79 MetadataRevised PortType 연산.....	51
표 80 MetadataRevised 바인딩.....	51
표 81 MetadataReviseNotification 연산의 인수 변환 테이블.....	52

1 목적 (Purpose)

1.1 SOAP 구현 매핑

1.1.1 본 규격의 목적은 SEMI E125 규격에 따른 구현 매핑을 SOAP 1.2 프로토콜에 제공하는 것이다. 본 문서는 SEMI E125에 정의된 UML 클래스를 지원하기 위해 사용되는 XML 스키마 데이터 유형을 설명하고, SEMI E125에 지정된 인터페이스를 통해 정의한 작업을 지원하기 위해 사용되는 WSDL 포트 유형과 바인딩 정의에 대해 기술한다.

2 범위 (Scope)

2.1 본 문서의 범위는 EqSD 모델을 XML 스키마로 표현하고 해당 WSDL 포트 유형 및 바인딩을 설명하는 것으로 한정한다. 새로운 도메인 정보 또는 개념을 SEMI E125 모델에 추가하는 것은 이 문서의 범위가 아니다. 단, 활용 가능한 WSDL 또는 XML 스키마를 제공하기 위해 꼭 필요한 내용은 추가되었다.

2.2 본 규격은 SEMI E132 인증을 필요로 하며, SEMI E132 개념을 SEMI E125 규격에 적용하기 위한 필수 조건을 정의한다.

주의: SEMI 표준 및 안전 가이드라인은 이 문서를 사용하여 모든 안전 관련 문제를 해결할 수 있다고 주장하지 않는다. 문서 사용자는 적절한 안전 및 건강 보호 조치를 취하고 상기 문서를 사용하기 전 관련 규제 또는 기타 제한사항의 적용 여부를 확인할 책임이 있다.

3 제한 (Limitations)

3.1 없음.

4 참조 표준 및 문서 (Referenced Standards and Documents)

4.1 SEMI 표준 및 안전 가이드라인

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 — Specification for Equipment Self Description (EqSD)

SEMI E128 — Specification for XML Messaging

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

SEMI E138 — XML Semiconductor Common Components

SEMI E145 — Classification for Measurement Unit Symbols in XML

4.2 OMG 표준¹

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 표준

¹ Object Management Group, Inc., 250 First Ave. Suite 100, Needham, MA 02494, USA. Telephone: 1.781.444.0404; Fax: 1.781.444.0320;
<http://www.omg.org>

Extensible Markup Language (XML) 1.0 (Fourth Edition) — W3C, 2006 년 8 월 16 일: <http://www.w3.org/TR/2006/REC-xml-20060816/>

Namespaces in XML (Second Edition) — W3C, 2006 년 8 월 16 일: <http://www.w3.org/TR/2006/REC-xml-names-20060816/>

XML Schema Part 0: Primer (Second Edition) — W3C, 2004 년 10 월 28 일: <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

XML Schema Part 1: Structures (Second Edition) — W3C, 2004 년 10 월 28 일: <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

XML Schema Part 2: Datatypes (Second Edition) — W3C, 2004 년 10 월 28 일: <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) — W3C, 2007 년 4 월 27 일: <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>

SOAP Version 1.2 Part 2: Adjuncts (Second Edition) — W3C, 2007 년 4 월 27 일: <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 기타 표준

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

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

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

주의: 별도 명시된 경우를 제외하고 인용된 모든 문서는 최신 출판본을 기준으로 한다.

5 용어 (Terminology)

5.1 약자 및 약어

5.1.1 SOAP — Simple Object Access Protocol(단순 객체 접근 프로토콜)

5.1.2 UML — Unified Modeling Language(통합 모델링 언어)

5.1.3 W3C — World Wide Web Consortium(월드와이드웹 컨소시엄)

5.1.4 WSDL — Web Service Description Language(웹 서비스 기술 언어)

5.1.5 XML — eXtensible Markup Language(확장 가능한 마크업 언어)

5.2 정의

² World Wide 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.w3c.org>

5.2.1 *Unified Modeling Language(UML: 통합 모델링 언어)* — Booch, Rumbaugh 및 Jacobson 이 개발한 객체 지향적 설계 및 보기 표현을 위한 표기법(notation). 이들은 세 사람의 표기법과 다른 표기법의 특징을 통합하여 어디에나 사용 가능한 단일 객체 지향적 표기법인 UML 을 개발했다.

5.2.2 *eXtensible Markup Language(XML: 확장 가능한 마크업 언어)* — 문서 및 통신에서 컨텍스트 및 콘텐츠 용량이 큰 데이터를 표현하기 위해 사용되는 마크업 언어. XML 은 SGML 을 확장한 문서 지향적 마크업 언어다. W3C 가 인터넷에 사용하기 위해 개발한 것이다.

6 규범 (Conventions)

6.1 UML 을 XML 스키마로 변환

6.1.1 사용자는 UML, XML, WSDL, SOAP 및 스키마 규격에 대한 실용적 지식을 갖고 있어야 한다(¶ 4.2 및 4.3 참조). 본 문서는 이 주제들에 대한 별도의 학습 정보를 제공하지 않는다.

6.1.2 본 문서는 SEMI E121 에 규정된 XML 가이드라인을 준수한다.

6.1.3 본 문서에서 준수하는 핵심 가이드라인을 요약하면 다음과 같다.

- SEMI E125 에 정의된 UML 클래스의 속성은 장비의 메타데이터 설명 전송 효율을 향상시키기 위해 보통 XML 속성으로 표현된다. 단, 데이터 구조에 해당하는 UML 속성 또는 XML 속성으로 표현할 수 없는 구문들은 예외이다. CDATA(non-parsed character data)를 포함하는 텍스트 값에도 이 규범이 적용되지 않을 수 있다.
- 합성 연관(Composition association)은 하나의 XML 요소로 매핑된다(지정된 배수가 1 보다 큰 경우 여러 XML 요소로 매핑).
- 집합(Aggregation)은 집합에 포함된 요소 또는 대상 유형의 하나 이상의 고유 속성에 대한 참조의 배열로 매핑된다.
- 상속(Inheritance)은 complexType 확장을 사용해 모델링한다. UML 모델에 추상 기본 클래스(abstract base class)에 대한 참조가 있는 경우, 파생 가능한 유형별로 선택 컴포지터(choice compositor)를 사용해 유형 대체를 통한 임의 확장이 인스턴스 문서에 포함되지 않도록 한다.

6.1.4 UML 클래스를 XML 스키마 유형으로 변환하는 방법은 표 1 과 같이 표 형식을 사용해 설명한다. UML 클래스에 정의된 연산은 WSDL 연산으로 변환된다.

6.1.5 변환 테이블 열의 머리말 설명

- 속성 또는 역할 이름 — 속성의 경우 속성의 이름이 여기에 표시된다. 연관(집합 또는 합성 포함)의 경우 UML 다이어그램의 역할 이름이 여기에 표시된다. 합성에는 보통 역할 이름이 할당되지 않는다. 따라서 'none'으로 표시된다.
- UML 이름/유형 — 속성의 경우 UML 속성의 데이터 유형이 여기에 표시된다. 연관의 경우 연관의 유형이 여기에 표시된다. 가능한 연관 유형은 '합성', '집합' 또는 기본적으로 '연관'이다. UML 은 이 세 가지 유형의 연관을 정의한다.

- XML 요소 및 속성 – UML 속성 또는 연관 표현에 사용되는 XML 구문(Construct)의 유형을 제공한다.
- XML 이름/유형 — 변환된 XML 구문의 이름 및 데이터 유형을 제공한다. 유형은 빌트인 유형(예: xsd:string) 또는 XML 스키마에 정의된 named 유형일 수 있다.

표 1 변환 테이블의 예

속성 또는 역할 이름	UML 이름/유형	XML 요소 또는 속성	XML 이름/유형
friend	association	element	Friend: HumanReferenceArray
employees	aggregation	element	Employees: HumanArray
family	composition	element	Family: HumanArray
name	string	element	Name: xsd:string

6.1.6 연산을 XML 스키마로 변환

6.1.6.1 UML 인터페이스 클래스에 정의된 연산을 XML 스키마 유형으로 변환하는 방법은 0 와 같이 표 형식을 사용해 문서화한다. 설명 중인 연산의 입/출력 인수(argument) 각각이 하나의 열에 표시된다.

표 2 연산 입/출력 인수 변환 테이블의 예

인수 이름	형식	XML 요소 또는 속성	XML 이름/유형
<추상 명세(abstract specification)의 인수 이름>	<추상 명세의 인수 형식>	<인수가 요소 또는 속성으로 모델링되는지 여부>	<속성/요소의 네임스페이스 한정 이름(namespace-qualified name)>

6.2 XML 스키마 및 WSDL 파일 문서화

6.2.1 본 문서에는 규격의 핵심 요소인 XML 스키마와 WSDL 보충 파일이 포함되어 있다. 이 문서들은 각각 0 과 같이 표 형식을 사용해 설명한다.

표 3 스키마/WSDL 문서 설명 테이블의 예

보충 파일 이름	<.xsd 또는 .wsdl 파일 이름>
대상 네임스페이스	<스키마 또는 WSDL 파일에 정의된 대상 네임스페이스>
들여오거나 참조된 네임스페이스	<스키마 또는 WSDL 파일이 들여오거나 사용하는 네임스페이스>
설명	<스키마 또는 WSDL 파일의 목적/기능에 대한 간략한 설명>

6.3 다이어그램을 이용한 XML 스키마 문서화

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

6.3.1 본 문서는 문서에 포함된 XML 스키마 데이터 유형과 요소 정의를 그래픽으로 표현한다. XML 스키마 그래픽 표기법은 별도의 표준이 없고 XML 도구별로 각자의 표기법을 사용한다. 본 문서에서는 Altova Corporation 의 XMLSpy 가 제공하는 표기법을 사용한다. 0 에 나와있는 샘플 XML 스키마 다이어그램은 이후 이 문서에서 사용되는 스키마 그래픽 표기법을 설명하기 위한 것이다.

6.3.2 이 다이어그램에서 직사각형 상자는 XML 요소 정의를 나타낸다. owner 요소들은 왼쪽에, owner 요소에 포함된 요소들은 오른쪽에 표시된다. 샘플 다이어그램의 경우 ParentType 에 Child1, Child2, Child3 및 Child4 가 포함되어 있다. 반면 Child2 에는 Child2a, Child2b 및 Child2c 가 포함되어 있다. 추가 심볼(8 면체 상자)은 시퀀스 또는 선택을 의미한다. 이 심볼들에 대한 설명은 0 를 참조한다.

표 4Altova XMLSPY 스키마 다이어그램 심볼

	오른쪽 요소들이 순서대로 연결된 시퀀스가 필요하다는 뜻이다. 각 요소의 기수(cardinality)는 1 이다. (sequence)
	오른쪽 요소들이 순서대로 연결된 시퀀스를 옵션으로 추가할 수 있다는 뜻이다. 각 요소의 기수는 1 이다. (sequence)
	오른쪽 요소들이 무작위로 연결된 시퀀스가 필요하다는 뜻이다. 각 요소의 기수는 1 이다. (all)
	오른쪽 요소 중에서 반드시 선택해야 한다는 표시다. 오른쪽에 정확히 1 개 또는 2 개의 요소만 존재해야 한다. (choice)
	PCDATA(parsed character data)가 포함된 요소를 나타낸다.
	다른 곳에서 정의된 그룹 또는 요소에 대한 참조를 나타낸다.

6.3.3 실선을 사용한 그래픽은 필수 요소를 의미한다. 점선은 옵션 요소를 의미한다. 우측 하단의 숫자 또는 범위는 기수를 나타낸다. 기본 기수는 1 이다.

6.3.4 다이어그램을 간소화하거나 특정 부분에 집중하기 위해 일부 세부사항은 숨겨질 수 있다. 8 면체 심볼은 오른쪽 끝에 작은 정사각형이 있다. 상자에 마이너스 부호 '-'가 있으면 모든 세부사항이 표시된다. 상자에 플러스 부호 '+'가 있으면 이 심볼 오른쪽에 있는 세부사항이 모두 숨겨진다. 예의 경우 숨겨진 세부사항이 없다.

6.3.5 노란색(또는 단색 인쇄본인 경우 회색) 상자는 정의된 다른 유형을 사용한다는 표시다. 즉, Child4 는 'Child4Type' 유형이다. Child4Type 은 Child4a 및 Child4b 를 정의한다. 이러한 세부사항은 다이어그램에는 표시되지 않을 수 있다. 객체 지향적 상속은 보통 XML 에서 유형 확장으로 표현된다. 그림 1 에서 ParentType 은 Child3 및 Child4 가 포함된 시퀀스를 추가하여 Child1and2Type 을 확장한다.

6.3.6 그림 1 을 보면 다음과 같은 추가 정보를 알 수 있다.

1. Child1and2Type 은 두 항목, Child2 와 Child1 이 순서대로 연결된 시퀀스다.
2. Child1 은 Child1a, 1 개 이상의 Child1b 및/또는 Child1c 이 순서대로 연결된 시퀀스(옵션)를 갖는다.
3. Child2 에는 Child2a, Child2b 및 Child2c 중 1 개 또는 2 개를 선택하는 선택이 포함되어 있다.

4. Child3 에는 Child3a 과 Child3b 이 무작위로 연결된 시퀀스가 있다.

6.4 XML 스키마 샘플

6.4.1 0 의 예에 사용된 샘플 XML 스키마는 아래에 제시되어 있다. XML 구문 및 시맨틱스(semantics)에 대한 자세한 설명은 ¶ 4.3 에 참조된 XML 문서에서 확인할 수 있다.

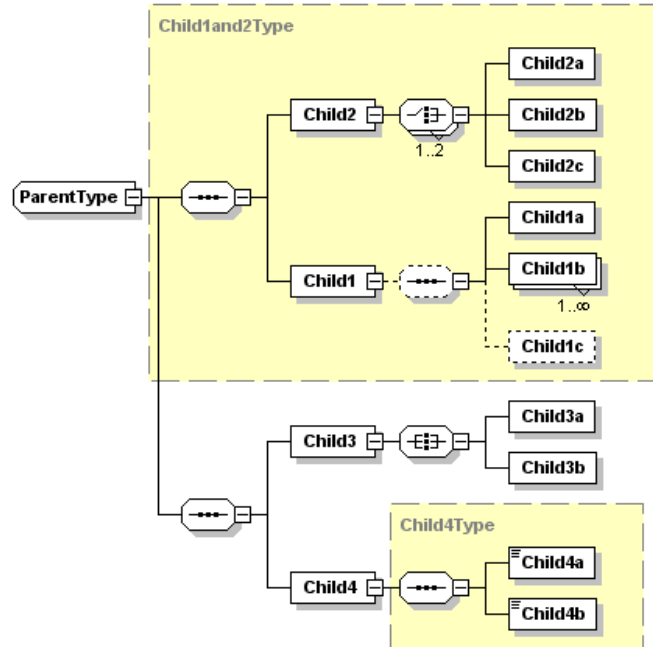


그림 1
XML 스키마 다이어그램의 예

```
<xsd:element name="Parent">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Child1"/>
      <xsd:complexType>
        <xsd:sequence minOccurs="0">
          <xsd:element name="Child1a"/>
          <xsd:element name="Child1b" maxOccurs="unbounded"/>
          <xsd:element name="Child1c" minOccurs="0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Child2">
  <xsd:complexType>
    <xsd:choice maxOccurs="2">
      <xsd:element name="Child2a"/>
      <xsd:element name="Child2b"/>
      <xsd:element name="Child2c"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Child3">
  <xsd:complexType>
```

```

<xsd:all>
  <xsd:element name="Child3a"/>
  <xsd:element name="Child3b"/>
</xsd:all>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

```

그림 2
샘플 XML

6.5 UML 을 WSDL 로 변환

6.5.1 보통 추상 명세(abstract specification)에 정의된 UML 인터페이스 클래스는 portType 정의 형식으로 WSDL 로 변환된다. 각 portType 정의는 해당하는 WSDL 바인딩 정의를 포함한다. 다음 표는 UML 인터페이스 클래스의 WSDL 포트 유형 및 바인딩 정의 문서화에 사용된 규범을 보여준다.

표 5 인터페이스 WSDL 포트 유형 테이블의 예

클래스 이름	<추상 명세의 UML 인터페이스 이름>
WSDL 포트 유형 이름	<WSDL 에 사용된 포트 유형 이름>
SEMI E125 연산 → WSDL 연산	<UML 인터페이스의 연산 이름> → <WSDL 포트 유형 연산 이름>

표 6 인터페이스 WSDL 바인딩 테이블 예

SEMI E125 클래스 이름	<추상 명세의 UML 인터페이스 이름>
WSDL 바인딩 이름	<WSDL 에 사용된 바인딩 이름>
SOAP 바인딩 스타일	<RPC 또는 문서>
SOAP 전송	<URI 식별 전송>

6.5.2 UML 인터페이스 클래스에 정의된 연산은 각각의 연산 정의 및 바인딩을 갖는다. 연산은 0 및 0 과 같이 표 형식을 사용해 설명한다.

표 7 연산 바인딩 테이블 예

SOAPAction	<이 연산에 사용할 Content-Type 머리말 'action' 값>
입력 머리말(WSDL 메시지, 필수 여부)	<입력 머리말을 제공하는 WSDL 메시지의 이름 및 머리말이 필수인지 여부>
출력 머리말(WSDL 메시지, 필수 여부)	<출력 머리말을 제공하는 WSDL 메시지의 이름 및 머리말이 필수인지 여부>

표 8 PortType 연산 테이블 예

입력 메시지 이름	<WSDL 포트 유형 연산에 입력으로 사용된 WSDL 메시지 이름>
출력 메시지 이름	<WSDL 포트 유형 연산에 출력으로 사용된 WSDL 메시지 이름>

7 SEMI E125 UML 을 XML 스키마 및 WSDL 에 매핑 (Mapping of SEMI E125 UML to XML Schema and WSDL)

7.1 WSDL 구조

7.1.1 SEMI E125 의 각 인터페이스 정의는 WSDL portType 및 바인딩 정의에 매핑된다. WSDL portType 정의의 이름은 SEMI E125 에 나와있는 인터페이스와 그 연산의 이름을 따서 짓는다. 각 portType 의 WSDL 바인딩 정의는 각 연산의 SOAP 1.2 엔벨로프(envelope) 내용을 지정하고 해당 XML 인코딩 스타일과 HTTP 머리말 사용을 정의하기 위해 사용된다. 모든 SEMI E125 WSDL 인터페이스는 문서/문자 인코딩을 사용한다. 전역 요소(global element) 정의를 통해 XML 스키마 파일에 정의된 SOAP 문자열과 본문 내용 일체를 인코딩에 적용한다.

7.1.2 0 은 WSDL 바인딩 및 portType 정의와 각 인터페이스에 사용된 XML 스키마 유형의 관계를 보여준다. portType 정의는 XML 스키마 'import' 명령문을 사용해 각 연산에 사용된 XML 스키마 유형을 들여온다. WSDL 바인딩 정의는 WSDL 'import' 명령문을 사용해 portType 정의를 들여온다. 0 는 SEMI E125 에 정의된 XML 스키마와 WSDL 파일을 보여준다.

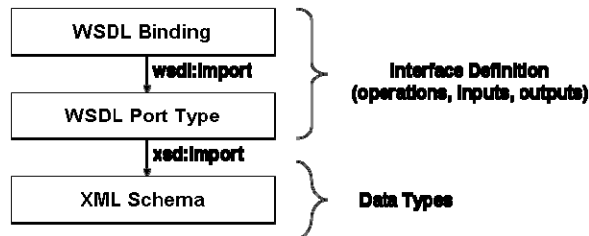


그림 3

XML 스키마 및 WSDL 파일 구조

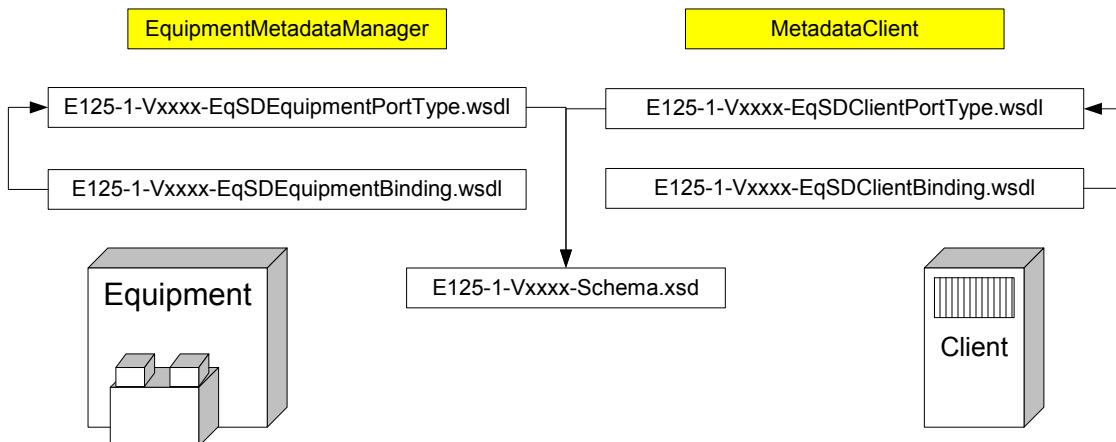


그림 4

SEMI E125의 WSDL 및 XML 스키마 파일

7.2 EquipmentMetadataManager

7.2.1 SEMI E132 세션 적용

7.2.1.1 SEMI E125에는 장비 메타데이터 접근에 대한 별도의 권한이 규정되어 있지 않다. 하지만 SEMI E125의 정보는 장비 공급업체의 소유이므로 설비 내에서 허가 없이 접근하지 못하도록 해야 한다.

7.2.1.2 이런 이유로 SEMI E125 EquipmentMetadataManager 인터페이스에서 정의한 모든 연산은 SEMI E132.1에 규정된 대로 SSL 인증을 활성화하여 HTTPS 전송(SSL 3.0을 통해 HTTP 1.1 전송)을 통해서 교환할 수 있다. SEMI E132.1을 통해 활성화된 경우 장비는 지원하는 모든 SEMI E125 연산에 대해 클라이언트-서버 상호 SSL 인증을 요구해야 한다. 또한 장비는 HTTPS 전송을 제외하고 어떤 연산에 대해서도 접근을 허용해서는 안 된다.

7.2.2 장비와 통신하기 위해 장비 및/또는 클라이언트 인증서를 설치하는 것이 어렵거나 불가능한 통합, 개발 및/또는 테스트 환경이 있을 수 있다. 이러한 환경의 현실적인 문제를 감안해 장비는 사용자가 SEMI E132.1에 지정된 장비 환경설정을 사용해 SEMI E125 EquipmentMetadataManager 인터페이스에 대한 SSL 인증을 비활성화할 수 있도록 지원해야 한다. 사용자는 SSL 인증을 비활성화하면 네트워크를 통해 장비에 접근이 가능한 모든 애플리케이션이 모든 SEMI E125 데이터에 제한 없이 접근할 수 있음을 명심해야 한다.

7.2.2.1 SSL 인증 활성화 여부와 관계 없이 장비는 요청에 첨부된 SOAP 엔벨로프(envelope) 머리말에 유효한 SEMI E128 세션 식별자가 없는 클라이언트의 요청은 모두 거부해야 한다. SEMI E128 세션 식별자는 SEMI E128에 규정된 대로 SEMI E128 SyncHeader 요소를 통해 제공해야 한다. 클라이언트가 인증된 세션 식별자를 제공하지 않는 경우, 장비는 SEMI Common Error 데이터 유형을 사용해 요청을 즉시 거부하고, SEMI E132.1에 규정된 대로 오류 코드 '6005'와 소스 'urn:semi-org:E132'를 출력해야 한다. SEMI E132 세션 식별자가 유효하지 않거나 세션 식별자를 제공하지 않는 SEMI E125 요청의 경우, 장비는 이 요청을 더 이상 처리해서는 안 된다.

7.2.3 Common Error 유형 사용

7.2.3.1 본 규격에서는 SEMI E125 전용 에러 코드를 별도로 정의하지 않는다. 단, 'source' 이름이 'urn:semi-org:E125'인 Common Error 유형은 계속 사용한다.

7.2.4 XML 스키마 및 WSDL 파일

7.2.4.1 본 규격에서 EquipmentMetadataManager 인터페이스를 위해 정의한 XML 스키마 및 WSDL은 다음 보충 파일에 저장되어 있다.

표 9 XML 스키마

보충 파일 이름	
E125-1-V0710-Schema.xsd	

보충 파일 이름	E125-1-V0710-Schema.xsd
대상 네임스페이스	urn:semi-org:xsd.E125-1.V0710.esd
들여오거나 참조된 네임스페이스	urn:semi-org:xsd.E120-1.V0310.CommonEquipmentModel urn:semi-org:xsd.E138.V0709.ccs http://www.w3.org/2001/XMLSchema
설명	이 파일은 EquipmentMetadataManager 및 MetadataClient 인터페이스가 사용하는 모든 E125 데이터 유형 및 요소를 정의한다.

표 10 EquipmentMetadataManager PortType 정의

보충 파일 이름	E125-1-V0710-EqSDEquipmentPortType.wsdl
대상 네임스페이스	urn:semi-org:ws.E125-1.V0710.EqSDEquipmentPortType
들여오거나 참조된 네임스페이스	urn:semi-org:xsd.E128.V0706.xms urn:semi-org:xsd.E125-1.V0710.esd http://schemas.xmlsoap.org/wsdl/ http://www.w3.org/2001/XMLSchema
설명	이 파일은 SEMI E125 XML 스키마에 정의된 데이터 유형을 기반으로 EquipmentMetadataManager 인터페이스의 모든 입/출력 메시지와 연산을 정의한다.

표 11 EquipmentMetadataManager 바인딩 정의

보충 파일 이름	E125-1-V0710-EqSDEquipmentBinding.wsdl
대상 네임스페이스	urn:semi-org:ws.E125-1.V0710.EqSDEquipmentBinding
들여오거나 참조된 네임스페이스	urn:semi-org:ws.E125-1.V0710.EqSDEquipmentPortType http://schemas.xmlsoap.org/wsdl/soap12/ http://schemas.xmlsoap.org/wsdl/
설명	이 파일은 각 연산에 필수 SOAP 및 HTTP 머리말과 XML 엔코딩 스타일을 지정하여 추상 portType 정의를 HTTP 및 SOAP 에 바인딩한다.

7.2.4.2 이 보충 파일들은 본 규격의 일부로 함께 제공된다. 이 문서들의 내용은 본 사양의 핵심 요소를 구성한다.

7.2.5 WSDL 포트 유형 개요

7.2.5.1 EquipmentMetadataManager WSDL portType 정의는 SEMI E125.1 XML 스키마 데이터 유형들을 하나의 named 연산 집합으로 조직화한다. 집합 내의 연산에는 SEMI E125 의 EquipmentMetadataManager 인터페이스에 정의된 UML 연산에 해당하는 입력과 출력이 포함된다. WSDL portType 자체는 SEMI E125 인터페이스의 이름을 따서 짓고, 각 WSDL portType 연산은 인터페이스에 정의된 해당 UML 연산의 이름을 따서 짓는다. 각 WSDL 연산은 SEMI E125 에 정의된 UML 연산의 입력과 출력(요청 및 응답)에 해당하는 2 개의 WSDL 메시지 정의로 구성된다. 0 는 SEMI E125 UML 인터페이스와 WSDL portType 정의의 관계를 보여준다. 0 는 이 관계를 표 형식으로 보여준다. WSDL 메시지 및 연산 정의는 §§ 7.2.8~7.2.16 에 설명되어 있다.

7.2.5.2 EquipmentMetadataManager 포트 유형 연산의 세부사항은 0 의 wsdl 파일에 정의되어 있다.

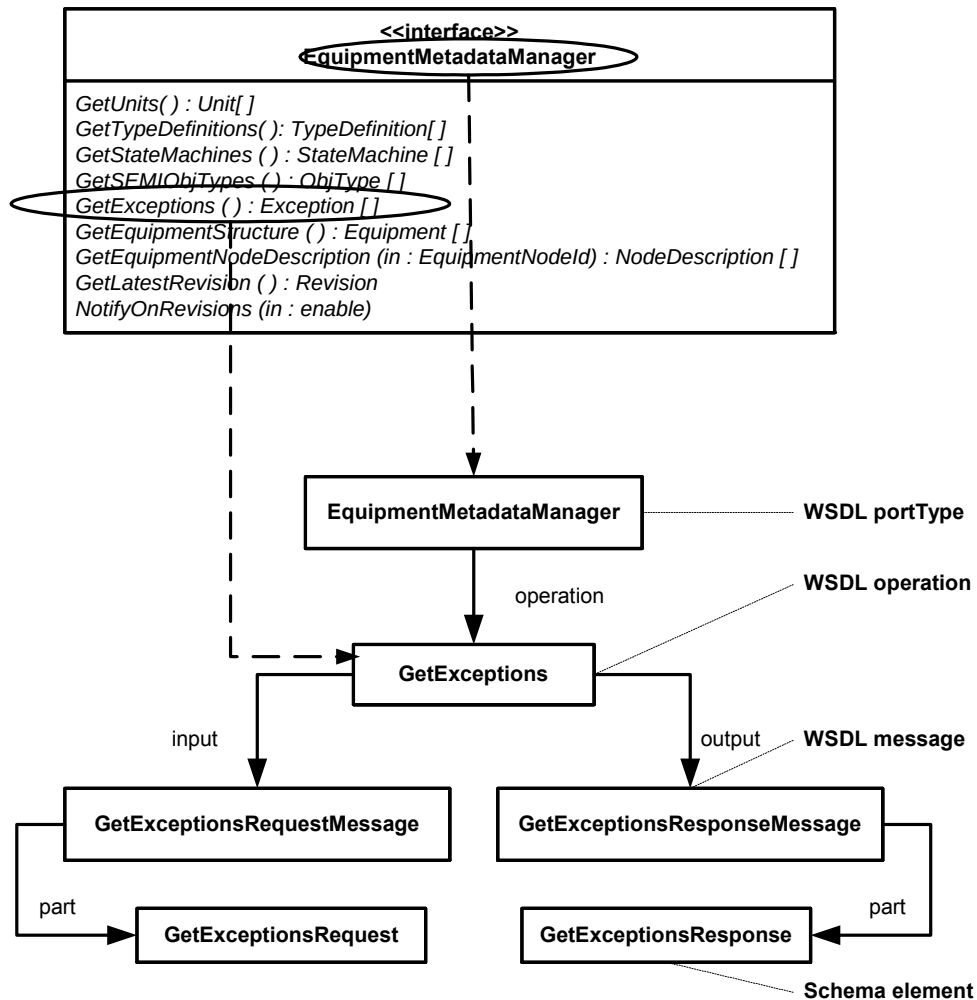


그림 5

EquipmentMetadataManager 를 WSDL portType 에 매핑

표 12 EquipmentMetadataManager 포트 유형

SEMI E125 클래스 이름	EquipmentMetadataManager
WSDL 포트 유형 이름	EquipmentMetadataManager
SEMI E125 연산 → WSDL 연산	GetUnits → GetUnits GetTypeDefinitions → GetTypeDefinitions GetStateMachines → GetStateMachines GetSEMIObjTypes → GetSEMIObjTypes GetExceptions → GetExceptions GetEquipmentStructure → GetEquipmentStructure GetEquipmentNodeDescriptions → GetEquipmentNodeDescriptions GetLatestRevision → GetLatestRevision NotifyOnRevisions → NotifyOnRevisions

7.2.6 WSDL 바인딩 개요

7.2.6.1 WSDL EquipmentMetadataManager 바인딩 정의는 해당 portType(SEMI E125.1 의 경우 SOAP 및 HTTP)에 사용할 메시지 및 전송 프로토콜과 인터페이스 스타일(SEMI E125.1 의 경우 문서 스타일)을 지정한다. 이 설정은 0 에 나와있다. 바인딩은 각 WSDL portType 연산에 사용되는 SOAP 머리말, SOAP 머리말이 필수인지 여부, 해당 연산에 사용할 Content-Type 머리말 'action' 값, 그리고 사용할 XML 인코딩(문자 인코딩)을 정의한다. 각 portType 연산의 바인딩 정의는 §§ 7.2.8~7.2.16 에 설명되어 있다.

7.2.6.2 EquipmentMetadataManager 바인딩 연산의 세부사항은 0 의 wsdl 파일에 정의되어 있다. 모든 EquipmentMetadataManager 연산에 사용된 SOAP 작업은 0 의 바인딩 대상 네임스페이스에 콜론(:)과 연산 이름을 붙여 사용한다(예: soapAction = <바인딩 대상 네임스페이스>:<연산 이름>).

soapAction = 'urn:semi-org:ws.E125.V0710.EqSDEquipmentBinding:<연산 이름>'

표 13 EquipmentMetadataManager 바인딩

SEMI E125 클래스 이름	EquipmentMetadataManager
WSDL 바인딩 이름	EquipmentMetadataManagerBinding
SOAP 바인딩 스타일	document
SOAP 전송	http://schemas.xmlsoap.org/soap/http

7.2.7 WSDL 서비스 개요

7.2.7.1 본 규격은 WSDL 서비스 정의를 제공하지 않는다. WSDL 서비스 정의는 클라이언트가 해당 인터페이스에 접근할 때 사용하는 엔드포인트(endpoint)를 정의하고 찾을 수 있는 방법을 제공한다. 장비가 공장 네트워크에 설치될 때까지는 이 정보를 확인할 수 없기 때문에 본 규격의 기술 범위에서 벗어난다. 클라이언트가 공장의 SEMI E125 장비 서비스를 찾을 수 있는 메커니즘을 정의하는 것은 사용자의 책임이다.

7.2.8 GetUnits

7.2.8.1 WSDL 연산 바인딩

7.2.8.1.1 'GetUnits' 연산 바인딩은 바인딩 wsdl 파일을 참조한다. 편의를 위해 아래 표에 바인딩 연산의 핵심 정보를 제공한다.

표 14 GetUnits 연산 바인딩

SOAPAction	<바인딩 대상 네임스페이스>:GetUnits
입력 머리말(WSDL 메시지, 필수 여부)	SyncHeaderMessage, required
출력 머리말(WSDL 메시지, 필수 여부)	SyncHeaderMessage, required

7.2.8.2 WSDL PortType 연산