

SEMI E39-0703 (Reapproved 1109)

객체 서비스 표준: 컨셉트, 동작, 서비스(OBJECT SERVICES STANDARD: CONCEPTS, BEHAVIOR, AND SERVICES)

이 표준은 기술적으로 Global Information & Control Committee 에 의해 승인되었으며, 해당 버전의 출판은 2009 년 9 월 4 일에 Global Audits and Reviews Subcommittee 에 의해 승인되었다. 1995 년에 초본이 출판되었고, 이전 버전은 2003 년 7 월에 발행되었다. 이번 출판본은 www.semi.org 에서 이용 가능하다.

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.

본 번역서는 참고용으로써, 영문원본과 한글번역본간에 차이가 발생할 경우, 영문원본을 우선으로 해야함을 알려드립니다. 본 번역서에 대해 문의사항이 있는 경우 krstandards@semi.org 로 문의바랍니다.

Table of Contents

1 목적 (Purpose).....	3
2 범위 (Scope).....	3
3 참조 문헌 및 표준 (Referenced Standards and Documents).....	3
4 용어 (Terminology).....	3
5 규약 (Conventions)	7
5.1 OMT 객체 정보 모델 (OMT Object Information Model).....	7
5.2 객체 속성 표시 (Object Attribute Representation).....	7
5.3 서비스 메시지 표시 (Service Message Representation).....	7
6 배경 (Background).....	9
7 적용 문헌 (Applicable Documents).....	9
8 기본 개념 및 동작 (Basic Concepts and Behavior).....	9
8.2 객체 속성 (Object Attributes).....	9
8.3 객체 정의 (Object Definition).....	12
9 객체 관계 (Object Relationships).....	12
10 추가 작업 (Additional Operations).....	16
11 서비스 및 시나리오 (Services and Scenarios).....	197
12 추가 서비스 (Additional Services).....	27
13 적용 (Applications).....	34
14 준수에 대한 필요조건 (Requirements for Compliance).....	35
Appendix 1	36

List of Tables

Table 1 상위 객체 속성 정의 (Top Object Attribute Definition)	11
Table 2 객체 서비스 (Object Services)	199
Table 3 객체 서비스의 매개 변수 디렉토리 (Object Services Parameter Dictionary)	19
Table 4 GetAttr 서비스 (GetAttr Service)	23
Table 5 SetAttr 서비스 (SetAttr Service)	23
Table 6 GetType 서비스 (GetType Service)	24
Table 7 GetAttrName 서비스 (GetAttrName Service)	25
Table 8 추가 객체 서비스 (Additional Object Services)	25
Table 9 추가 객체 서비스 매개 변수 디렉토리 (Additional Object Services Parameter Dictionary)	26
Table 10 생성 서비스 (Create Service)	26
Table 11 삭제 서비스 (Delete Service)	27
Table 12 첨부 서비스 (Attach Service)	27
Table 13 AttachSetAttr 서비스 (AttachSetAttr Service)	27
Table 14 분리 서비스 (Detach Service)	307
Table 15 재첨부 서비스 (Reattach Service)	308
Table 16 SupervisedObject 첨부 서비스 (AttachSupervisedObject Service)	308
Table 17 SupervisedObject (DetachSupervisedObject Service)	28
Table 18 ObjectAction 서비스 (ObjectAction Service)	30
Table 19 ObjectActionCompletion 알림 (ObjectActionCompletion Notification)	31
Table 21 GetServiceParameters	31

1 목적 (Purpose)

1.1 객체 서비스 표준(Object Services Standard (OSS))은 객체와 그 속성의 동작 및 데이터 기술을 설명하고자 전반적인 용어, 규약 및 표기법을 제공한다. 또한 객체 속성의 판독과 그 값의 설정 및 객체 내용의 요청을 위한 기본 서비스를 제공한다. 이 표준(Standard)은 중복되는 표현을 줄이고자 특정 객체를 정의한 다른 표준을 참조 한다.

2 범위 (Scope)

2.1 이 표준(Standard)의 범위는 다양한 public 객체를 위해 공통된 개념, 동작 및 서비스를 제공하는 것이다.

2.2 객체 모델은 다양한 표준(Standards)에 통용되며, 객체 서비스(Object Services)는 일반 객체를 정의하는 모든 표준에 사용될 수 있는 객체 관련 기본 정의를 비롯하여 객체 속성을 읽고 그 속성 값을 설정하는 데 필요한 기본 서비스를 제공한다. 이러한 서비스들은 객체 기반 데이터의 기본 관리를 허용한다.

2.3 본 문서에서 정의된 객체 서비스는 다른 표준(Standards)에서 제공되는 서비스가 포함되며, 또한 다른 표준(Standards)들과 독립적으로 제공된다.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the Documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

이 표준은 사용에 관련된 안전 사항의 명시를 의미하지 않는다. 사용자는 이 표준을 사용하기에 앞서 안전과 건강에 관련된 규제 방안을 수립하고, 규제나 기타 제한사항들에 대한 적용 방안을 규정할 책임이 있다.

3 참조 문헌 및 표준 (Referenced Standards and Documents)

3.1 ISO 표준(ISO Standard¹)

ISO 9595 — 공통적인 운영 정보 서비스 (CMIS)

3.2 그 외 문서(Other Documents)

James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen, Object-Oriented Modeling and Design, Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

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

별도의 표기가 없을 시, 인용된 모든 서류는 가장 최근 발행본이다.

4 용어 (Terminology)

4.1 다음의 기본 정의는 논리적인 순서로 제공된다.

4.2 정의 (Definitions)

4.2.1 필요조건 (Requirements)

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>

4.2.1.1 필수 준수 사항 (*fundamental compliance*) — 객체나 서비스 자원(resource)의 필수 요구사항에 대한 적합성.

4.2.1.2 필수 요구 사항 (*fundamental requirements*) — 표준 준수 사항을 반드시 충족해야 하는 정보와 동작들에 대한 요구사항. 필수 요구 사항은 특정 어플리케이션 영역, 객체, 혹은 서비스에 적용된다.

4.2.2 객체 (*Objects*)

4.2.2.1 객체 (*object*²) — 특정한 데이터 집합과 동작을 가진 개체(entity). 객체들은 물리적이거나 개념적이다.

4.2.2.2 표준화된 객체 (*standardized object*) — SEMI Standards 에서 공식적으로 정의하는 바이자, SEMI E39 의 필수적인 요구사항들을 준수하는 객체.

4.2.2.3 객체 모델 (*object model*) — 구조를 보여주는 객체들의 고정된 그래픽 모델 – 객체 ID, 객체 속성, 작업, 객체간의 관계.

4.2.2.4 객체 유형 (*object type*) — 유사 객체 그룹의 공식적 범주. 동의어: 객체 클래스(object class). 예: 장비, 웨이퍼, 캐리어.

4.2.2.5 객체 인스턴스 (*object instance*) — 객체 유형의 인스턴스. 객체 유형은 객체 인스턴스가 실제 객체인 경우 템플릿(template)과 같다. 예: 특정 공장에 설치된 실제적이고 특정한 광학 스텝퍼는 “광학 스텝퍼(Optical Stepper)” 유형의 예이다.

4.2.2.6 객체 속성 (*object attribute (attribute)*) — 객체에 대한 정보. “장비(equipment)” 객체 형식의 예: 제조사, 모델, 일련 번호.

4.2.2.6.1 속성은 가시성(public/private) 선택성(optionality), 접근(access)성에 따라 다양한 방법으로 분류된다:

- 필수적인 속성(*fundamental attribute*) — 표준 서비스에 대해 필수적 준수가 요구되는 속성.
- 선택적 속성 (*optional attribute*) — 하나 혹은 그 이상의 선택적인 표준 서비스에 대한 지원만이 요구되는 속성.
- 비공개 속성 (*private attribute*) — 내부적인 목적을 위해서만 사용되고 공개 서비스를 통해 알려지지 않는(보이지 않는) 속성.
- 공개 속성 (*public attribute*) — 요청에 의해 다른 개체들에 대한 서비스로 제공되는 현재 값을 가지고 보여지는 속성.
- 읽기 속성 (*read-only attribute (RO)*) — public 서비스를 통해 변경될 수 없다.
- 읽기/쓰기 속성 (*read/write attribute (RW)*) — public 서비스를 통해 변경될 수 있다.

4.2.2.7 속성 명칭(*attribute name*) — 속성을 식별하기 위해 사용되는 공식적인 이름. 공개(public) 속성의 이름(데이터 유형) 객체 정의에 포함되고 고유성을 띈다.

² The term “object” may refer to either an object class or an instance of a class.

4.2.2.8 객체 식별자 (*object identifier*) — 주어진 유형의 특정 객체 인스턴스 (인스턴스화) 를 고려한 다수의 정보 아이템 집합. 정해진 범주 내에서 그 객체의 다른 모든 인스턴스들로부터 인스턴스를 구별하게 하는 고유성을 가진다. Note: 객체는 하나 이상의 식별자를 가질 수 있고, 식별자는 단지 하나의 속성으로 이루어져 단순(*simple*)하거나 하나 이상의 속성으로 이루어져 복잡(*complex*)하기도하다. 예: 장비 제조사, 모델, 일련번호의 조합은 특정한 설비(*installation*)를 고유하게 식별하는 식별자의 역할을 한다.

4.2.2.9 객체 핸들 (*object handle*) — 내부적으로 사용되는 응용프로그램에 의해 부여된 정수 혹은 이진 식별자. Note: 핸들(*handle*)은 공개 속성(*public attribute*) 으로 사용될 수 있지만 일반적으로 서비스 사용자와 제공자 사이에서 현재 관계를 벗어나 지속적이거나 고유할 수 없다. 핸들(*handle*)의 지속성은 객체를 정의하는 표준에 의해 명시된다.

4.2.2.10 작업 (*operation*) — 객체에 의해 수행되거나, 내재된 기능(*Function*). 예: 장비인 경우 “run,” “stop,” “abort.”

4.2.2.11 집합 객체(*aggregation object*) — 다른 객체들로 구성된 객체. 요소 중 하나가 빠지게 된다면, 총합은 완전성(*intergrity*)을 어느정도 잃게 된다.

4.2.2.12 부품 객체 (*component object*) — 집합의 일부인 객체.

4.2.2.13 컨테이너 객체 (*container object*) — 다른 유형의 객체를 갖고 있는 객체. 내용은 순차적일 수도 있고 아닐 수도 있다.

4.2.2.14 내용 (*contents*) — 컨테이너(*container*) 내의 객체. 예: 카세트 안에 있는 웨이퍼들, 도서관에 있는 책.

4.2.2.15 주체 객체(*owner object*) — 집합, 컨테이너(*container*) 혹은 다른 객체의 관리자인 주체 객체는 다른 객체를 소유한다.

4.2.2.16 종속 객체(*owned object*) — 다른 객체로 구성 혹은 포함되거나 관리되는 종속 객체는 다른 객체에 소유된다.

4.2.2.16.1 객체는 다수의 주체(*owner*)를 갖는다. 예를 들어 이는 공유된 자원(*resource*)일 수 있다.

4.2.3 서비스 (*Services*)

4.2.3.1 범위 (*scope*) — 특정한 주체 객체로 시작하고 “소유(*Owns*)” 관계의 위계 순서에 따라 하향 진행되는, 하나 혹은 그 이상의 객체의 사양.

4.2.3.2 서비스 또는 메시지 서비스 (*service(or message service)*) — 제공자가 사용자에게 제공한 기능(*Function*)을 나타낸다. 서비스는 메시지 구조와 프로토콜의 정의를 제외하고, 매개변수(*parameters*) 리스트에서 서술한 서비스 기초 요소의 시퀀스(*sequence*)로 구성된다

4.2.3.2.1 서비스는 그 제공자에 의해 처리되거나 그렇지 않을 수도 있다. 일부 서비스의 호출은 자체 요구사항을 충족시키기 위해 다른 객체와 상호 작용해야 한다. 서비스가 그 전 이나 후 상태를 만족시키지 못한다면 예외가 발생된다. 서비스를 제공하는 객체는 장비 상태로 인한 적절치 못한 동작(*illegal action*)을 방지하기 위해 서비스 완료를 요구하거나 제한할 수 있다.

4.2.3.3 알림 서비스 (*notification service*) — 서비스 제공자에 의해 시작되고, 서비스 소비자(consumer)/가입자(subscriber)에게 전송된다. 기대되는 응답은 없다.³

4.2.3.3.1 알림은 두 개의 서비스 기초 요소로 이루어진다 – 송신자가 통신 장비로 전송하는 메시지와 통신 장비에서 수신자로의 지시(indication)이다.

4.2.3.4 요청 서비스 (*request service*) — 서비스 소비자에 의해 초기화된다. 요청은 제공자로부터 데이터 혹은 활동(작업)을 요구하며, 특정한 응답 메시지를 기대한다.

4.2.3.4.1 요청은 수신자 응답을 요구하는 메시지로 이루어지며, 그 기초 요소는 응답이 응답과 확인(*response* 와 *confirmation*)이라 불리는 추가적인 기초 요소들을 정의할 때 그 알림과 같다.

4.2.3.5 서비스 자원 (*service resource*) —기능의 특정 영역 내에서 하나 혹은 그 이상의 논리적 서비스 집단.

4.2.4 데이터 유형 (*Data Type*)

4.2.4.1 형식 (*form*) — 데이터의 유형: 양수, 무부호 정수, 정수, 소수점, 계수, 불 대수, 문자, 형식화된 문자, 구조, 리스트, 순서화된 리스트.

4.2.4.2 양의 정수 (*positive integer*) — 모든 양의 정수 값을 취한다. 메시징 프로토콜은 가능한 값의 범위에 제한을 둘 수 있다.

4.2.4.3 무부호 정수 (*unsigned integer*) — 양의 정수 혹은 영(zero)의 값을 취한다. 메시징 프로토콜은 가능한 값의 영역에 제한을 둘 수 있다.

4.2.4.4 정수 (*integer*) — 음수 혹은 무부호 정수의 값을 취한다. 메시징 프로토콜은 가능한 값의 영역에 제한을 둘 수 있다

4.2.4.5 소수점 (*floating point*) — 양수 혹은 음수의 실수 값을 취한다. 메시징 프로토콜은 가능한 값의 영역에 제한을 둘 수 있다.

4.2.4.6 계수 (*enumerated*) — 가능한 값에 대해 하나의 유한한 집합을 취한다. 이러한 값들은 논리적 이름이 주어질 수 있지만 소수점을 제외하고 단일-아이템의 데이터 형식으로 나타낼 수 있다.

4.2.4.7 불대수 (*Boolean*) — TRUE 또는 FALSE 와 같이 하나 혹은 두 개의 가능한 값을 취한다.

4.2.4.8 텍스트 (*text*) — 텍스트 문자열. 메시징 프로토콜은 길이나 ASCII 설명 같은 제한을 부과할 수 있다.

4.2.4.9 형식화된 텍스트 (*formatted text*) — 제한된 형식을 가진 텍스트 문자열. 이는 위치(position)나 특수 문자의 사용 혹은 둘 모두를 제한 할 수 있다.

4.2.4.10 구조 (*structure*) — 혼합 데이터 형식일 수 있는 특정한 아이템의 집합으로, 특정한 배열로 이루어지는 복잡한 구조를 뒀다.

4.2.4.11 리스트 (*list*) — 동일한 형태 (위의 형태 중에 하나)를 가지는 하나 혹은 그 이상의 집합.

³ This does not preclude the implementation of a response message in the message protocol.

4.2.4.12 순서 리스트 (*ordered list*) — 중요한 순서로 아이템이 나타나는 순서에 대한 리스트.

5 규약 (Conventions)

- 처음 소개될 때 굵은 글씨체로 정의된 용어.
- 속성 이름과 같이 공식적으로 예약된 텍스트 문자열은 밑줄을 친다.

5.1 OMT 객체 정보 모델 (*OMT Object Information Model*) — 객체 모델은 객체 모델링과 설계 (*Object-Oriented Modeling and Design*), James Rumbaugh가 개발한 객체 모델링 기술(OMT)⁴을 이용하여 기술된다.⁵

5.1.1 이 표기에 대한 개요와 용어에 대한 개략적인 논의는 별첨으로 제공된다.

5.2 객체 속성 표시 (*Object Attribute Representation*) — 표준화 된 객체에 대한 객체 정보 모델은 아래의 제목을 가진 속성 정의 표로 설명된다:

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Rqmt</i>	<i>Form</i>
속성에 대한 공식적인 텍스트 명칭	포함된 정보에 대한 설명	RO or RW	Y or N	(참조 § 4.2.4)

5.2.1 접근(access) 열은 서비스 사용자의 속성에 대한 접근(access)을 나타내기 위해 RO(읽기만) 혹은 RW(읽고 쓰기)를 사용한다.

5.2.2 이 속성이 서비스에 대한 기본 요건을 만족하기 위해서 지원되어야 한다면, 요구사항 (Reqmt) 란에 “Y” 혹은 “N”으로 표시된다.

5.2.3 상단 표는 속성의 데이터 유형을 보여주기 위해 사용된다. (정의는 § 4.2.4 을 참조)

5.3 서비스 메시지 표시 (*Service Message Representation*)

5.3.1 서비스 자원 정의 (*Service Resource Definition*) — 서비스 자원(resource) 정의 표는 다음의 표에서 보여지는 것처럼, 주어진 서비스그룹에 대한 특정한 메시지 집합을 정의한다:

<i>Message Service Name</i>	<i>Type</i>	<i>Description</i>
메시지 이름	N or R	서비스 목적.

5.3.1.1 유형(Type)은 N = 알림(Notification) 혹은 R = 요구(Request) 둘 중 하나 일 수 있다.

5.3.1.2 알림 유형 메시지는 서비스 제공자에 의해 초기화되고 서비스 사용자 혹은 요청자로부터 응답을 기대하지 않는다.

⁴ For a full description of the Object Modeling Technique, see Object-Oriented Modeling and Design, Rumbaugh, Blaha, Premerlani, Eddy and Lorenson, Prentice Hall, 1991.

5.3.1.3 메시지는 서비스 사용자 혹은 요청자에 의해 초기화된다. 이러한 요청(Request) 메시지는 제공자에게 데이터나 활동을 요청하며, 특정한 응답 메시지를 기대한다(메시지 내용에 대한 추정은 없음).

5.3.2 서비스 매개변수 디렉토리 (*Service Parameter Dictionary*) — 다음의 서비스 매개변수(parameter) 디렉토리는 하나 혹은 그 이상에 대한 매개변수(parameters)를 정의한다:

<i>Parameter</i>	<i>Form</i>	<i>Description</i>
매개변수(parameter) X	데이터 유형	X 로 불리는 매개변수(parameter)는 A 안의 B 이다.

5.3.2.1 서비스의 각 매개변수(parameter)는 테이블에서 행으로 나타낸다. 첫 번째 열은 매개변수(parameter)의 이름을 나타내는데, 이는 초기 통신의 형식과 내용을 설명한다.

5.3.2.2 형식(Form) 열은 매개변수(parameter)에 포함된 데이터의 유형을 보여주기 위해 사용된다. (정의는 § 4.2.4 참조)

5.3.2.3 서비스 매개변수 디렉토리(*Service Parameter Dictionary*) 표에서 설명(Description) 열은 매개변수(parameter)의 의미, 추정할 수 있는 값, 다른 매개변수(parameter)들과의 연관성을 기술한다.

5.3.2.4 대다수의 매개변수(parameter)가 “XxxList,”로 정의되는 것을 방지하기 위해, 이 문서는 “Xxxx(리스트)”로써 리스트를 참조하는 규약을 도입한다. 이 경우, 리스트의 정의가 아닌 가변적인 Xxx 가 주어진다. “list”는 동일한 데이터 유형의 0 또는 그 이상의 그룹 집합을 나타낸다. 요청과 응답이 모두 사용되는 경우, 요청의 리스트 순서는 응답에 포함된다. 리스트는 0 개의 요소가 특별히 허용되지 않는 한 최소한 하나의 요소를 포함해야 한다.

5.3.3 서비스 메시지 정의 (*Service Message Definition*) — 다음의 서비스 메시지 정의 표는 서비스에서 사용되는 매개변수(parameter)를 정의한다:

<i>Parameter</i>	<i>Req/Ind</i>	<i>Rsp/Conf</i>	<i>Description</i>
매개변수 X	(하단 참조)	(하단 참조)	서비스에 대한 설명.

5.3.3.1 Req/Ind 와 Rsp/Conf 열은 메시지 방향으로 매개변수(parameter)를 연결한다. 이 메시지는 ‘Request’로 불리는 개시자가 이 메시지를 전송하며, 수신자는 이 메시지를 요청 또는 ‘Indication’라고 칭한다. 수신자는 ‘Response’를 전송할 수 있으며, 최초 송신자는 이를 ‘Confirmation’이라고 칭한다.

5.3.3.2 다음의 코드는 Req/Ind 와 Rsp/Conf 열에서 나타나고, 매개변수(parameters)의 정의에서 사용된다. (예를 들어, 각 매개변수(parameter)가 각각의 방향에서 사용되는 방법):

- “M” — 필수 매개변수(Mandatory Parameter): 유효한 값을 제공해함.
- “C” — 조건 매개변수(Conditional Parameter): 일부 상황에서 정의될 수 있으며, 값은 선택적이거나 다른 매개변수(parameter)값에 의해 값이 결정될 수 있다.

- “U” — 사용자 정의 매개변수(parameter)
- “-” — 사용되지 않는 매개변수(parameter)
- “=” — (응답에서만 사용됨) 응답에서 이 매개변수(parameter)의 값은 (만약 정의된다면) 그 초기 매개변수(parameter)값과 반드시 일치해야 함.

6 배경 (Background)

6.1 다양한 서비스 표준들을 개발 중에 장비 업체들이 공통적으로 객체와 관련된 용어와 서비스를 요구했다는 것을 알 수 있었다. 본래 각 제안은 객체 속성의 값을 얻기 위해 서비스용 메시지를 포함했으며 일부는 속성 값들을 설정하기 위해 서비스용 메시지를 제공했다.

6.1.1 이는 다른 개별적인 서비스 표준에서 사용된 정의와 서비스를 제공할 수 있는 표준의 규정으로 불필요한 중복을 없애기 위해 결정되었다.

7 적용 문헌(Applicable Documents)

SEMI Standards Factory Automation Software Volume⁵

ISO/TR 8509:1987 — Information Processing Systems, Open Systems Interconnection – Service Conventions⁶

William Stallings, Networking Standards, “A Guide to OSI, ISDN, LAN, and MAN Standards,” Addison-Wesley Publishing Company

8 기본 개념 및 동작(Basic Concepts and Behavior)

8.1 이 섹션은 public 객체에 공통의 개념과 동작을 정의한다.

8.1.1 객체 지향 분석은 객체 지향 기술과 전통적인 기술을 사용하여 실행되는 응용프로그램에 대해 통용되는 도구이며, 객체 정보 모델은 직관적으로 이해될 수 있는 관계를 설명하는 효과적인 메소드(method)를 제공한다.

8.1.2 객체 지향 기술의 용어에 대한 간략한 논의는 별첨에서 제공된다.

8.2 객체 속성 (*Object Attributes*) — 객체 속성(*object attribute*)은 주어진 객체 유형의 모든 인스턴스가 가지는 데이터 값이다. 통신 목적을 위해, public 속성은 해당 객체에 고유한 이름(name)을 할당한다. 객체 속성은 ¶ 8.1.1 에서 정의된 텍스트의 사용을 준수하는 열거된 숫자 값 혹은 따로 보존된 속성 이름(*attribute name*) 중 하나를 참조할 수 있다.

8.2.1 *Public* 속성(*public attribute*)은 공식적인 서비스(*public services*)를 통해 요구되거나 참조된다. 다른 속성들은 특정한 실행에서 존재하지만, *private* 속성(*private attributes*)으로 불리며 public 서비스(*public services*)를 통해 보이지 않는다.

⁵ These documents may be obtained from SEMI, 3081 Zanker Road, San Jose, CA 95134, USA; <http://www.semi.org>

⁶ International Organization for Standardization, ISO Central Secretariat, 1 rue de Varembé, Case postale 56, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.749.01.11; Fax: 41.22.733.34.30; <http://www.iso.ch>

8.2.2 특정 속성들은 특정 객체를 위한 기본 원칙(*fundamental*)으로서 지정된다. 이는 명시된 객체를 사용하거나 참조하는 모든 실행으로 지원되며, 속성은 필수가 아닌 선택(*optional*)이다.

8.2.3 텍스트 사용 (*Use of Text*) — 텍스트로 정의된 값들은, ASCII 로 주어졌을 때, 객체 속성 값, 객체 이름, 서비스 매개변수(parameters)들 포함한 제약들로 제한된다.

8.2.3.1 ASCII 에서 텍스트는 물음표 “?”, 별표 “*”, 물결표 “~” 제외하고, 20₁₆ 과 7E₁₆ 사이의 텍스트로 제한된다.

8.2.3.2 물음표와 별표가 필터와 검색에서 “wild characters”로 사용을 위해 보존되는 반면, 물결표는 내부적으로 사용하고자 위해 공간을 전환할 수 없는 시스템을 허용하기 위해 보존된다.

8.2.3.3 특정 내용에 사용된 텍스트는 추가 제한을 받는다.

8.2.3.4 별도 언급이 없다면 텍스트가 비교의 목적으로 사용될 경우에는 중요하지 않지만, 가독성(readability) 개선하기 위해 사용되며 필요한 경우 언제든지 보존되어야 한다.

8.2.4 객체 유형 (*Object Type*) — 객체는 항상 자체 유형을 인식하고 있으므로 객체 유형에 대한 속성 ObjType 이 모든 public 객체에 대해 요구된다. ObjType 은 객체의 공식적인 분류를 포함하는 텍스트 문자열이며, 이 텍스트 문자열은 “greater than” 부호 “>” 와 콜론 “:” 이 제외된 추가적인 제약과 함께 ¶ 8.1.1 에서 정의된 규약을 준수하며, 공백으로 시작되거나 끝나지 않아야 한다.

8.2.4.1 표준화된 객체를 위한 객체 유형(ObjType 값)은 보존되며, 보존된 유형은 개별적인 표준에 한정된다.

8.2.5 객체 식별자 (*Object Identifier*) — 모든 객체 인스턴스는 동일한 유형의 객체에 대해 모든 다른 인스턴스로부터 그 인스턴스를 고유하게 구별하는 속성을 가진다. 객체는 하나 이상의 식별자를 가지며, 식별자는 객체 정의의 일부로 정의된다.

8.2.5.1 ObjID 는 모든 공개(public) 객체에 대해 텍스트-기반의 필수적인 객체이며, 일부 유형의 객체 식별자의 역할을 위해 알려진 속성 형식을 제공한다. 텍스트 문자열은 “greater than” 부호 “>” 와 콜론 “:” 이 제외된 추가적인 제약과 함께 ¶ 8.1.1 에서 정의된 통신을 준수하며 공백으로 시작되거나 끝나지 않아야 한다.

8.2.5.2 다른 객체에 대한 책임이 있는 운영자, 집합체, 컨테이너(container)와 같은 객체의 관점에서 볼 때(¶ 9.1 and ¶ 9.2 참조), 다른 객체들에 대한 ObjType 와 ObjID 의 조합은 고유하다. 예를 들어, 만약 여러 개의 프로세스 챔버가 하나의 진공 펌프를 가진다면, 각 프로세스 챔버는 최대 하나의 “Pump:Vacuum.”의 조합 ObjType:ObjID> 에 대해 직접 접근권(access)을 가질 것이다. 그러나 조합이 다른 챔버와 구별되도록 요구하지 않는다. 호스트는 다른 집합체를 가지는 이들의 속성으로 여러 개의 진공 펌프들을 구별한다.

8.2.5.3 각 객체에 대한 객체 정의는 일반적인 접근법(access)으로 사용되는 특정한 식별자를 명시한다. 일반적으로 식별자로서 숫자 값을 가지는 식별자를 사용하는 객체에 대해(아래에 설명된 “handle,”과 같은), ObjID 의 값은 그 속성의 값을 설정하는 숫자의 텍스트 문자열이다.

8.2.5.4 객체는 다른 식별자와 ObjID 를 가진다.

8.2.6 객체 핸들 (*Object Handle*) — 객체 핸들(*handle*)은 객체를 생성하는 응용프로그램에 의해 할당되는 숫자(numeric) 값을 가지는 객체이다. 핸들(*handle*)은 객체의 식별자로 사용될 수 있으며, 일반적으로 이는 내부 사용 목적이고 객체의 공식적 정의에서 public 객체로써 정의 혹은 정의 되지 않을 수도 있다.

8.2.6.1 사용되는 핸들(*handle*)은 특정한 서비스 자원(resource)에 대해 서비스 제공자와 서비스 사용자간의 단일 결합의 경우와 특정 내용 내에서만 보장한다. 핸들은 그 연계 내 혹은 그 이상 지속적일 수 있다. 핸들(*handle*)의 지속성은 객체의 정의의 부분으로써 명시되고 OSS의 범위를 넘어서나.

8.3 객체 정의 (*Object Definition*) — 객체 정의는 § 5.3에서 기술된 속성 정의 표를 포함한다.

8.3.1 모든 public 객체의 상위유형인 객체는 상위객체(*top object*)라고 불린다. 상위객체의 속성과 작업은, Figure 1에서 보여지는 것처럼, 모든 공식적 객체들에 대한 필수적인 요구다. 즉, 모든 객체는 객체 속성 *ObjType* 과 텍스트 문자열에 응답하고, 모든 public 인스턴스들은 속성을 얻으며(*get attributes*) 속성을 설정하는(*set attributes*) 작업에 응답한다.

SUPERSEDED

Top
ObjType: text ObjID: text
get attributes set attributes

Figure 1
상위 객체(Top Object)

8.3.2 객체에 대한 공식적 정의는 객체의 public 속성을 정의하는 표를 포함한다. 하단의 Table 1 에서 표시하는 상위 객체에 대한 객체 속성 정의 표(Object Attribute Definition)에서 보는 바와 같이 모든 public 객체에서 요구되는 두 개의 필수 속성을 포함한다.

Table 1 상위 객체 속성 정의
(Top Object Attribute Definition)

Name	Definition	Access	Req	Form
ObjType	객체 유형.	RO	Y	Text
ObjID	객체 식별자.	RO	Y	Text

8.3.3 접근법(access)은 OSS 서비스를 통해 속성의 값을 읽고 쓰는 기능을 나타낸다. 객체 유형과 식별자는 OSS 서비스를 통해 변경되지 않지만 서비스 사용자가 객체(인스턴스)를 생성할 때 객체의 식별자에 대한 값을 할당하는 것은 가능하며, 이는 객체와 제공되는 서비스에 의존한다.

9 객체 관계(Object Relationships)

9.1 이 절은 통신을 제한함으로 혹은 관계에 대한 정보를 통해 통신에 영향을 미치는 객체 간의 관계를 설명한다.

9.2 객체 계층 구조(Object Hierarchy) — 일부 객체는 그들이 정의될 때 컨테이너 객체(container objects)나 집합 객체(aggregation objects)로 명시된다.

9.2.1 집합 객체는 집합의 구성요소라 불리는 다른 객체들로 구성되며, 이는 Figure 2 에서 설명하고 있다. 구성 요소(components)는 하나 혹은 그 이상의 다른 유형일 수 있으며 동일한 유형일 수도 있다.

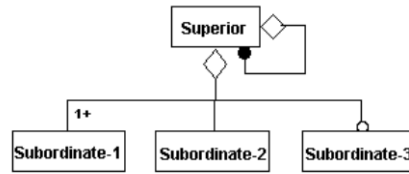


Figure 2

두 개의 구성 요소 유형을 가진 집합 (Aggregation with Two Component Types)

9.2.2 컨테이너(container) 객체는 동일하거나 다른 유형의 객체를 포함하지만 이들로 구성되지는 못한다.

9.2.3 Figure 3 은 “Contains”라 불리는 관계와 연계된 두 개의 객체 유형을 보여준다. 검정색의 원은 Type-1 이 Type-2 의 영(zero) 혹은 그 이상의 객체 포함하는 것을 나타낸다. 일반적인 컨테이너(container) 유형의 예는 파일 디렉토리이다.

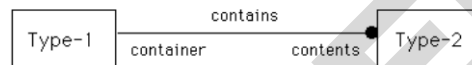


Figure 3

컨테이너와 내용들 (Container and Contents)

9.2.4 집합체는 특정 구성요소가 빠져있다면 완전성(integrity)의 일부를 잃지만, 컨테이너(container)는 내용물이 없더라도 완벽한 완전성(integrity)을 가진다. 자동차는 그 자체가 집합체인 수많은 구성 요소들의 집합체이며, 또한 제거할 수 있거나 구성 요소로써 간주되지 않는 내용물(운전자, 승객, 내용물)을 가진다. 컨테이너(container) 객체의 다른 예는 리스트, 사전, 도서관등이며, 내용물이 없는 컨테이너(container)는 비었다고(empty) 말한다.

9.2.5 집합체 혹은 컨테이너(container)는 상위객체(*superior object*)라 불리고 그것의 구성요소 혹은 내용물은 하위객체 (*subordinate objects*)라고 불린다.

9.2.6 “is composed of” (역할: 집합/구성요소)과 “contains” (역할: 컨테이너(container)/내용물) 와의 관계뿐만 아니라, 공장과 컨트롤(control) 시스템에서 일반적으로 발생하는 다른 계층적 관계는 “supervises”(역할: 관리자/관리되는)이다. 이는 관리되는 객체가 관리자로부터 지시의 일부 혹은 전체를 수용하는 컨트롤(control) 관계에서 나타난다. 다시 말해, 관리자는 관리될 수 있는 책임을 지게 된다. 관리자는 보통 종속자와 “is composed of” 혹은 “contains” 관계를 가진다.

9.3 범위와 소유권(*Scope and Ownership*)— 집합체, 컨테이너(container), 관리자의 계층 관계는 때때로 구체적인 객체를 지정하도록 요구된다. 이러한 관계는, Figure 4 에서 보여지는 것과 같이 다중 단계 트리(tree) 구조를 가진다. 한 단계에서 종속 객체는 다음 하위 단계에서 상위 객체일 수도 있다. 최 상위 단계에서 상위 객체는 트리(tree)의 루트(*root*)라고 불린다.

NOTE 1: Figure 4 는 OMT 표기법으로 나타내지 않았다.