

SEMI E125-0308

装置自己記述(EqSD)の仕様

本仕様は、global Information & Control Committee で技術的に承認されている。現版は 2007 年 9 月 5 日、global Audits and Reviews Subcommittee にて発行が承認された。2007 年 10 月に www.semi.org で、そして 2007 年 11 月に CD-ROM で入手可能となる。初版は 2003 年 7 月発行、前版は 2007 年 3 月に発行された。

注意: SEMI E125 は、SEMI E125.1 の改訂を反映して、0308 の出版サイクルで更新された。

目次

- 1 目的
 - 1.1 仕様の目的
- 2 範囲
 - 2.1 範囲内
 - 2.2 範囲外
- 3 制限
 - 3.1 抽象モデル
 - 3.2 他の制限事項
- 4 参照スタンダード
 - 4.1 SEMI スタンダード
 - 4.2 ISO スタンダード
 - 4.3 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 文書および表記法
 - 6.2 用語
- 7 背景
 - 7.1 動機付け
- 8 概要
 - 8.1 装置メタデータの構成
 - 8.2 装置メタデータへのアクセス
- 9 装置メタデータ管理のためのインタフェース
 - 9.1 装置メタデータマネージャ (Equipment Metadata Manager)
 - 9.2 メタデータクライアント
- 10 装置のメタデータ
 - 10.1 装置のメタデータの管理
 - 10.2 装置の構造
 - 10.3 装置ノード
 - 10.4 パラメータ
 - 10.5 パラメータ型
 - 10.6 単位
 - 10.7 例外事象
 - 10.8 StateMachines
 - 10.9 ObjTypes
 - 10.10 SimpleEvent
- 11 準拠の要件
 - 11.1 装置自己記述の準拠

1 目的

1.1 仕様の目的

1.1.1 本仕様は、装置に存在する参照し得る変数、イベント、例外事象、物理的な装置の構成を示した記述をサプライヤが提供できるようにする方法を説明するものである。ソフトウェアシステムが利用できるこのような情報により、工場の自動システムに装置を統合する過程で手助けするための一具として使用できるものである。

2 範囲

2.1 範囲内

2.1.1 本文書では、サプライヤの装置によって提供される重要なデータ、イベント、例外事象を記述するためにサプライヤが使用するクラスを指定する。この仕様は、本質的に静的な情報のみを記述する(つまり、装置実行中に動的に変化しない情報)。本文書では、クライアントが使用してこの情報にアクセスできるインタフェースも指定する。

2.1.2 本仕様は、データ収集管理のためのSEMI仕様で定義されるデータ取得インタフェースをサポートするすべての半導体製造装置に適用される。

2.2 範囲外

2.2.1 本仕様では、装置の構成、インタフェース、使用可能なデータを記述する情報を取り出し、この情報の流れを維持するために必要な挙動を除き、装置に必要ないかなる新しい挙動も定義しない。

2.2.2 メタデータによって記述できる根本的な概念および挙動モデル(つまり、キャリア管理、プロセス/コントロールジョブ等)の詳細は、これらの概念専用の文書内で別途指定される。サプライヤがそのような概念を実装したという事実およびクライアントが

この実装とそれが生成できるいかなるデータも発見できるという事実を記述できることのみがこの仕様の範囲内である。

2.2.3 本仕様では、装置によって提供されるメタデータが直接人間によって読める必要はない。この情報をエンドユーザが簡単に理解できる形で整理し人間のユーザに提示するためのアプリケーションが記述されることが期待される。

注意: このスタンダードはその使用に関連したすべての安全問題を取り扱うことを意図していない。このスタンダードの使用者は、その責任において、適切な安全および健康上の作業方法を確立し、また使用前に規制上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 抽象モデル

3.1.1 この仕様は、抽象モデルにすぎない。この仕様を実装技術に結びつけるには、付属仕様を作成しなければならない。

3.2 他の制限事項

3.2.1 本仕様は、本仕様が定義する情報を受け取るクライアントを特定する方法に使用される機構を定義するものではない。装置は要求を出すことがあるクライアント個々の求める場合に沿った通信を確立していて、それらのクライアントに、通知を送るために必要な場合がすべてわかるものと仮定する。

3.2.2 本仕様は、クライアントを識別あるいは認証するために使用される機構を定義するものではない。クライアントは一意的に識別でき、装置は異なるクライアントから発せられる要求を区別することができるものと仮定する。

4 参照スタンダードおよび文書

4.1 SEMI スタンダード

SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)

SEMI E39 — Object Services Standard: Concept, Behavior, and Services

SEMI E120 — Provisional Specification for the Common Equipment Model (CEM)

4.2 ISO スタンダード¹

ISO 639:1988 — Code for the representation of names of languages

ISO 3166-1:1997 — Codes for the representation of names of countries and their subdivisions – Part 1: Country codes

ISO 8601 — Representations of dates and times, 1988-06-15

ISO 8601 Draft Revision — Representations of dates and times, draft revision, 2000

4.3 その他のスタンダード

American National Standards Institute (ANSI) X3.135-1992 – Information Systems – Database Language – SQL; <http://www.ansi.org>²

Internet Engineering Task Force (IETF) RFC 1776 – Tags for the Identification of Languages, ed. H. Alvestrand. 1995; <http://www.ietf.org/rfc/rfc1766.txt>

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

Uniform Resource Name (URN) Syntax, IETF RFC 2141, May 1997; <http://www.ietf.org/rfc/rfc2141.txt>

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 この仕様内で使用される多くの略語および頭字語の説明は, SEMI のホームページ, <http://www.semi.org/> で入手可能な SEMI Comilation of Terms に記載されている可能性がある。ほとんどの場合, これらの用語はこの項には記載されていない。

5.1.2 *IDL (Interface Definition Language)* — Interface Definition Language。

5.1.3 *OMG (Object Management Group)* — オブジェクトマネジメントグループ。

5.1.4 *SQL (Standard Query Language)* — 標準照会言語。

5.1.5 *UML (Unified Modeling Language)* — 統一モデリング言語

5.1.6 *URN (Uniform Resource Name)* — Uniform Resource Name

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

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

2 American National Standards Institute, Headquarters: 1819 L Street, NW, Washington, DC 20036, USA. Telephone: 202.293.8020; Fax: 202.293.9287, New York Office: 11 West 42nd Street, New York, NY 10036, USA. Telephone: 212.642.4900; Fax: 212.398.0023; Website: <http://www.ansi.org>

5.2 定義

5.2.1 この仕様内で使用される多くの用語の定義と説明は、SEMIのホームページ、<http://www.semi.org/> で入手可能なSEMI Compilation of Termsに記載されている可能性がある。ほとんどの場合、これらの用語はこの項には記載されていない。

5.2.2 関連情報 8 には、UMLスタンダードから直接とったUML用語の有用な定義が記載されている。本文書で使用されるUML用語は、これらの定義に準拠している。必要に応じて、この項を参照すること。

5.2.3 メタデータ(metadata) — データを記述するために使用されるデータ。例えば、ツールが特定の条件で関連する複数の変数によってイベントを報告できる場合、そのイベントのメタデータは、そのイベントが生成される条件の説明、各変数の型と単位、イベント自体のIDを提供する。

6 表記法

6.1 文書および表記法

6.1.1 UMLモデル記述仕様(UML) — この仕様では、すべてのクラス、状態、オブジェクト、シーケンスダイアグラムの表現にUMLの表記法を使用する。これらの表記法の詳細については、UML仕様を参照すること。

6.1.2 属性表 — 次の表では、この仕様内で定義されるクラスの属性を一覧表示し記述するために使用される表の例を記載する。

表 1 属性表のフォーマット

属性名	定義	必須	形式
		Y または N	次のリストを参照。

6.1.2.1 形式 — 属性のデータ型を定義する。この列内のデータ型を記述するために使用される用語は、SEMI Compilation of Termsで定義されている、あるいは仕様の一部として記載される。SEMIの型名の意味の定義については、Compilation of Termsを参照すること。

6.1.3 UMLで表現する関連 — クラス間のUMLで言う関連を表現するために使用される仕掛けは、実装に依存する。本文書は、本質的に抽象を扱い、そのような仕掛けを指定したり、示唆したりしない。本仕様の実装を提供する関連スタンダードには、本文書で示すUMLの関連を表現するために使用される仕掛けが記述されていないなければならない。

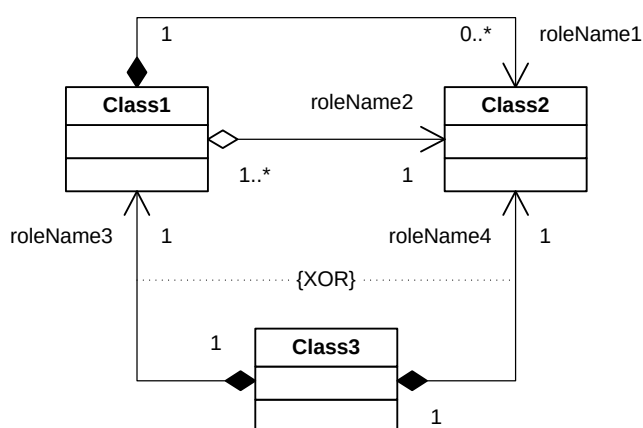


図 1
関連付けの表記

6.1.3.1 関連付けの記法 — 本文書では、関連を指定するすべてのクラス図において、UMLの集約ダイヤモンド記号の補足表記、役割名、各側の多重度、方向性を使用する。表記を見やすくするために補足表記のない関連

は使用しない。詳細については、UML 1.4 の仕様書の 3.43 項を参照すること。すべての役割名はデフォルトでパブリックなので、可視記号は使用しない。

6.1.3.1.1 白抜きダイヤモンド記号の箇所は、先方のクラスのインスタンスが集約するクラス内で共有できることを示す。黒塗りのダイヤモンド記号の箇所は、先方のインスタンスが多くても 1 つの複合体であるクラスの一部であることを示す。集合／複合記号の部分全体における意味は、本仕様では重要ではない。詳細については、UML 1.4 の仕様書の 3.43.2.5 項および 3.48 項を参照すること。

6.1.3.1.2 ダイアグラム記号の箇所は場合によって、UML の“XOR”の制約を使用することがある。“XOR”の制約は、関連先のクラスの任意のインスタンスに関して多くの考え得る関連の中から、一時に 1 つだけのインスタンスを作成できることを示す。詳細については、UML 1.4 の仕様書の 3.42.5.1 項を参照すること。

6.1.4 関連表 — 以下に、本仕様で定義するクラス間の関連の一覧を記述するために使用される表の例を示す。

表 2 関連表のフォーマット

関連での役割名	定義	コメント

6.1.4.1 関連での役割名 — 指定する関連での役割の名称。

6.1.4.2 定義 — 関連の機能や目的を記述する。

6.1.4.3 コメント — 関連に関する任意の追加コメントや注記。

6.1.5 操作定義表 — 次の表に、本仕様で定義されるインタフェースの操作の一覧を記述するために使用される表の例を示す。

表 3 操作定義

操作	説明	タイプ
		次の箇条書きを参照。

6.1.5.1 操作 — 操作の名称を指定する。

6.1.5.2 タイプ — 操作メッセージの用途を指定する。本仕様では、要求-応答(RR)の用途のみが使用される。要求-応答メッセージは、発信者と受信者が関わるメッセージである。要求-応答の交換においては、発信者が受信者に単一の要求メッセージを送り、受信者は発信者にその要求に対する単一の応答を送り返す。送り切り(Fire-and-Forget) は送信者と受信者が関わるメッセージである。送り切りの通信では、発信者が受信者に一通のメッセージを送り、受信者は発信者に関連する応答を送り返さない。

6.1.6 操作の引数定義表 — 次の表に、本仕様で定義されるインタフェースの操作の引数を一覧で記述するために使用される表の例を示す。

表 4 操作の引数定義

引数	説明	種類	形式
		次の箇条書きを参照。	次の箇条書きを参照。

6.1.6.1 引数 — 引数の名称を指定する。

6.1.6.2 種類 — 引数が操作に対する「入力」、「出力」、「エラー」の引数の何れかを示す。「エラー」の引数は常に「出力」の引数として機能するが、操作が正しく完了しなかったことを示す。

6.1.6.3 形式 — 引数のデータ型を定義する。この列内のデータ型を記述するために使用される用語は、SEMI Compilation of Terms で定義されている、あるいは仕様の一部として記載される。SEMI の型名の意味の定義については、Compilation of Terms を参照すること。

6.2 用語

6.2.1 クラスと型— 本文書では「クラス」と「型」をUMLクラスを指すものとして同義に混用する。

7 背景

7.1 動機付け

7.1.1 本仕様は、さまざまなサプライヤの半導体装置を自動化工場の環境に統合する問題を解決する手助けをするためと、装置が提供し、広範囲の用途となり得る情報とデータを説明する標準的な仕組みを提供できるための両方の目的で設計されている。ユースケースの例いくつかは概要として、関連情報 1 を参照のこと。

7.1.2 本仕様では、装置の本質的な特徴をソフトウェアアプリケーションに教えるために必要な情報を定義し、サプライヤが伝えたいある特定の装置に固有な情報、あるいはSEMIスタンダードでは規定されていない情報による拡張的な特徴についても提供する。

7.1.3 装置メタデータは人間のエンドユーザにインタフェースを提供するようなアプリケーションに役立つ設計のもので、目的は装置の構成、使用可能なデータ項目、イベント、例外事象等を理解し、参照することである。さまざまな目的(診断、トラブルシューティング、プロセス制御等)にデータ収集を実施するために、いろいろな場合に、ユーザは装置のメタデータを参照する。

7.1.4 結果としてこの情報の検知や利用が人の介入の必要もなく自動でできることで、標準化された状態モデルとその他のデータの優位性に着目して書かれたアプリケーションにとっても利用価値が高い様、設計されている。例えば、特にあるタイプの装置の詳細がわかっている診断、状態の健全性の監視、あるいはサプライヤが提供する他のアプリケーションも(エンドユーザと)共通に、装置特有の構成およびデータを人手が介助しなくても探し当て、かつ利用できる。

8 概要

8.1 装置メタデータの構成

8.1.1 図 2 に本仕様に書かれている概念と、SEMI E120 共通装置モデルとの関係を示す。各メタデータ概念の目的の簡単な説明は、概要のみ本項で紹介する。関連するクラスの詳細については 10 章参照。

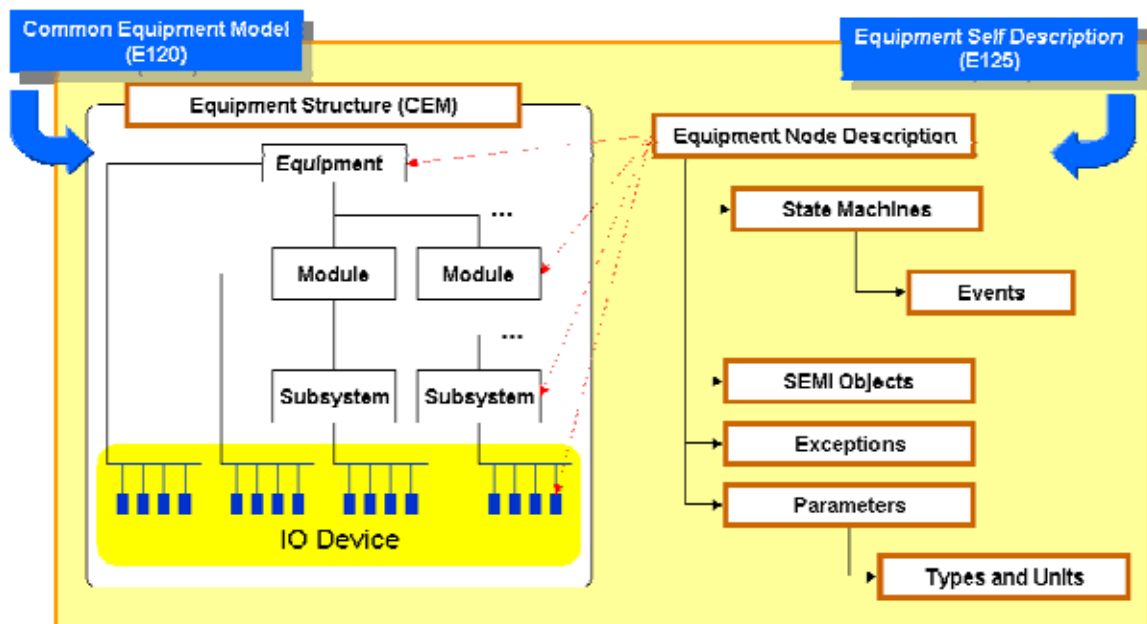


図 2
メタデータ概念の構造

8.1.2 装置ノードの記述 (Equipment Node Description)

8.1.2.1 本仕様では、イベント、例外事象、パラメータ、SEMI E39 で規定されるオブジェクトが、装置の物理構造として示される各ノードに関連あるものであることを示すことができるクラスを定義する。SEMI E120 は、装置の物理構造を表現するために使用するクラスを定義し、本仕様は、イベント、例外事象、パラメータ、生成されるオブジェクトとしてE120 の装置ノードに関連を持つクラスを定義する。装置ノードを説明するクラスは、¶10.3 で定義する。

8.1.3 パラメータ (Parameters)

8.1.3.1 本仕様は装置が提供する変数を表現するクラスを定義する。それら変数は“パラメータ”と呼ばれる。データの取得可能な型として表現できる概念なら何でも、パラメータを使ってモデル化可能である。例えば、SEMI E30 の装置定数、データ変数、状態変数の概念は、どれもパラメータで表現可能である。装置が生成できるデータはどの種類でも（つまり、測定装置の測定結果）パラメータで表現可能である。パラメータは、装置ノード、SEMI E39 規定のObjTypeまたは例外事象についてのみ定義できる。パラメータについては¶10.4 で説明する。

8.1.4 型と単位 (Type and Units)

8.1.4.1 パラメータを表現するため、本仕様は装置上で使用されるデータの型と単位を言い表す方法を与える。例えば、装置が整数、文字列、論理値の配列から構成される構造化されたデータ型を有する複数のパラメータを定義する場合、本仕様は、そのデータ型の構造を言い表す際に使用できるクラスを与える。数値データ型については、本仕様では、そのデータ型の値が持つ単位を言い表す際に使用できるクラスを与える。例えば、あるパラメータがマイクロファラッドで静電容量を表す場合、マイクロファラッドの単位自体と、そのパラメータの値がマイクロファラッド単位であることをどうやって示すかを本仕様を利用して言い表すことができる。パラメータの型の言い表し方は¶10.4.11 で定義し、単位の言い表し方は¶10.5.5 で定義する。

8.1.5 状態マシンとイベント (State Machine and Events)

8.1.5.1 本仕様では、装置が実装する状態マシン（対応する遷移とイベントを含む）を記述する際に使用できるクラスを定義する。これらのクラスを使用して、SEMI定義の如何なる状態マシンやサプライヤ定義の如何なる状態マシンも言い表すことができる。本仕様は、あるイベントが発生した時に、報告のなかで使用できるパラメータを言い表す際に使用できるクラスも定義する。状態マシンの言い表し方については、¶10.8 で定義する。

8.1.6 SEMIのオブジェクト (SEMI Objects)

8.1.6.1 本仕様は、装置が実装する如何なるSEMI E39 のObjTypeを言い表すときに使用できるクラスを定義する。SEMI ObjTypeの状態マシンとイベントは、状態マシンを言い表す際に使用されるクラスと同じクラスを使用して言い表される。ObjTypeの属性は、パラメータと型を言い表す際に使用されるクラスと同じクラスを使用して言い表される。ObjTypeのサービスの言い表し方はサポートされない。SEMIのObjTypeの言い表し方は、¶10.8 で定義する。

8.1.7 例外事象 (Exceptions)

8.1.7.1 本仕様は、装置が例外事象として検出し報告できるエラー状態の存在を言い表すときに使用できるクラスを定義する。例外事象は、SEMIスタンダードE30 のすべてのアラームや装置サプライヤ定義の例外事象でクライアントに報告できるものを言い表すことができる。例外事象の言い表し方は、¶10.7 で定義する。

8.2 装置メタデータへのアクセス

8.2.1 装置メタデータは、工場での個々の装置毎の設置に直接、関わるものである。メタデータへのアクセスは、個々の装置の設置で可能な操作を集めたものとして用意される。このインタフェースについては、¶9.1 で説明する。

9 装置メタデータ管理のためのインタフェース

9.1 装置メタデータマネージャ (Equipment Metadata Manager)

9.1.1 図 3 に装置メタデータの内容にアクセスするために使用されるインタフェースを示す。

«interface» EquipmentMetadataManager
GetUnits() : Unit[] GetTypeDefinitions() : ParameterTypeDefinition[] GetStateMachines() : StateMachine[] GetSEMIObjTypes() : SEMIObjType[] GetExceptions() : Exception[] GetEquipmentStructure() : EquipmentElement[] GetEquipmentNodeDescriptions(in equipmentNodeIds : String[]) : NodeDescriptionResult GetLatestRevision() : MetadataRevision NotifyOnRevisions(in notification : NotificationRequest)

図 3

装置メタデータにアクセスするためのインタフェース

9.1.2 EquipmentMetadataManager の操作

表 5 EquipmentMetadataManager の操作の定義

操作	説明	型
GetUnits	装置が提供するすべての単位に間するメタデータを取り出す。	RR
GetTypeDefinitions	装置が提供するすべての型定義を取り出す。	RR
GetStateMachines	装置が提供するすべての状態マシンに関するメタデータを取り出す。	RR
GetSEMIObjTypes	装置が提供するすべての SEMI ObjType のメタデータを取り出す。	RR
GetExceptions	装置が提供するすべての例外事象のメタデータを取り出す。	RR
GetEquipmentStructure	装置が提供するすべての構造に関するメタデータを取り出す。	RR
GetEquipmentNodeDescriptions	要求されたすべての装置のノード関連に関するメタデータを取り出す。	RR
GetLatestRevision	装置メタデータが変更された最後の日時を取り出す。	RR
NotifyOnRevisions	メタデータが変更されたら通知するように装置に要求する。	RR

9.1.2.1 GetUnits — 装置はこの要求を受け付けた際に、装置が提供する単位に関わるメタデータすべてを応答として返さなければならない。

9.1.2.1.1 GetUnits 操作の引数

表 6 GetUnits の引数の定義

引数	説明	種類	形式
units	装置に定義されているすべての単位。	out	¶10.6 にあるような Unit 型の要素を適当な順序でリストにまとめたもの。

9.1.2.2 GetTypeDefinitions — 装置はこの要求の受信により、装置が提供する型定義のメタデータすべてを応答として返さなければならない。

9.1.2.2.1 GetTypeDefinitions 操作の引数

表 7 GetTypeDefinitions の引数の定義

引数	説明	種類	形式
typesDefinitions	装置が提供する型定義、すべて。	out	¶10.5 にあるような ParameterTypeDefinition 型の要素を適当な順序でリストにまとめたもの。

9.1.2.3 *GetStateMachines* — 装置はこの要求の受信により、装置が提供する状態マシンのメタデータすべてを応答として返さなければならない。

9.1.2.3.1 *GetStateMachines* 操作の引数

表 8 *GetStateMachines* の引数の定義

引数	説明	種類	形式
stateMachines	装置が提供する状態マシンのメタデータすべて。	out	¶10.8 にあるような <i>StateMachine</i> 型の要素を適当な順序でリストにまとめたもの。

9.1.2.4 *GetSEMIObjTypes* — 装置はこの要求の受信により、装置が提供するOSS *ObjType*メタデータを応答として返さなければならない。

9.1.2.4.1 *GetSEMIObjTypes* 操作の引数

表 9 *GetSEMIObjTypes* の引数の定義

引数	説明	種類	形式
objTypes	装置が提供する SEMI <i>ObjType</i> メタデータすべて。	out	¶10.8 にあるような SEMI <i>ObjType</i> 型の要素を適当な順序でリストにまとめたもの。SEMI <i>ObjType</i> を提供しない場合は、このリストは空でなければならない。

9.1.2.5 *GetExceptions* — 装置はこの要求の受信により、装置が提供する例外事象のメタデータすべてを応答として返さなければならない。

9.1.2.5.1 *GetExceptions* 操作の引数

表 10 *GetExceptions* の引数の定義

引数	説明	種類	形式
exceptions	装置が提供する例外事象のメタデータすべて。	out	¶10.7 にあるような <i>Exception</i> 型の要素を適当な順序でリストにまとめたもの。

9.1.2.6 *GetEquipmentStructure* — 装置はこの要求の受信により、装置が提供する装置構造に関するメタデータすべてを応答として返さなければならない。

9.1.2.6.1 *GetEquipmentStructure* 操作の引数

表 11 *GetEquipmentStructure* の引数の定義

引数	説明	種類	形式
equipmentStructure	装置が提供する装置構造に関するメタデータすべて。	out	SEMI 120 型の <i>Link</i> もしくは <i>EquipmentElement</i> から収集された型の構造体をしたデータ。装置を全体的に示す。

9.1.2.7 *GetEquipmentNodeDescriptions* — 装置は、この要求の受信により、装置に定義されている装置のノードを言い表す要求された全部のメタデータを返さなければならない。その要求に認識されない装置のノードの指定子が含まれている場合、装置は認識された指定子のノード説明のみを返す。認識されない指定子は応答とともに返されなければならない。

9.1.2.7.1 *GetEquipmentNodeDescriptions* 操作の引数

表 12 *GetEquipmentNodeDescriptions* の引数の定義

引数	説明	種類	形式
equipmentNodeIds	要求された装置ノード説明の id。	in	それぞれ有効な SEMI E120 ロケータ値 (E120 関連情報 2 参照) に等しいテキストのリスト。空のリストの場合、装置はすべての装置ノードに対してノードの記述を返さなければならない。
equipmentNodeDescriptions	装置ノード説明のメタデータすべて。	out	<i>NodeDescriptionResult</i> 型の構造体をしたデータ。

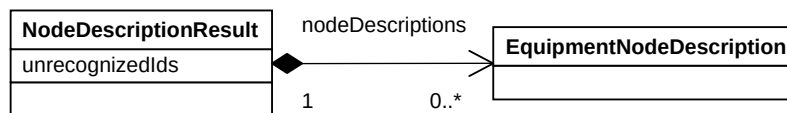


図 4

装置メタデータへのアクセスのためのインタフェース

9.1.2.8 *NodeDescriptionResult* — このクラスは要求された *EquipmentNodeDescription* エレメントのリストとその要求によって提供される認識されない装置ノードidを提供する。

9.1.2.8.1 *NodeDescriptionResult* 属性定義表

表 13 *NodeDescriptionResult* 属性定義

属性名	定義	形式
unrecognizedIds	最初の要求で提供された認識されない装置ノードの指定子。	<i>GetEquipmentNodeDescriptions</i> 要求で提供される認識されない装置ノードの指定子に等しいテキストのリスト。

9.1.2.8.2 *NodeDescriptionResult* 関連定義表

表 14 *NodeDescriptionResult* 関連定義

関連の役割名	定義	コメント
nodeDescriptions	要求された装置ノード説明のメタデータ。	<i>EquipmentNodeDescription</i> 型の 0 以上のエレメント。装置は、認識された装置ノード指定子のすべてに対してノード説明のメタデータを返さなければならない。要求の中に同じノード指定子が一度以上現れる場合、装置は対応するノード説明をひとつのみ返す。

9.1.2.9 *GetLatestRevision* — 装置は、この要求の受信により、メタデータの内容が変更された最終の日時を返さなければならない。装置は最終の変更日時を不揮発性メモリに保存しなければならない。装置は偽りの改訂日時、すなわち改訂日時は変更されたが、入手可能なメタデータに変更が生じなかった場合は改訂日時を生成してはならない。

9.1.2.9.1 *GetLatestRevision* 操作の引数

表 15 *GetLatestRevision* の引数の定義

引数	説明	種類	形式
revision	メタデータが変更された最終の日時。	out	¶9.1.2.11 にあるような <i>MetadataRevision</i> 型の構造体をしたデータ。

9.1.2.10 *NotifyOnRevisions* — 装置はこの要求の受信により、要求を行ったクライアントに対するメタデータ変更の発生通知を、クライアントから与えられた操作を示す引数に従って、可能か不可能にしなければならない。装置はクライアントの選択内容を不揮発性メモリに保存しなければならない。

9.1.2.10.1 *NotifyOnRevisions* 操作の引数

表 16 *NotifyOnRevisions* の引数の定義

引数	説明	種類	形式
notification	通知に関するクライアントの選択を指定する	in	¶9.1.2.12 にあるような <i>NotificationRequest</i> 型の構造体をしたデータ。

9.1.2.10.2 通知を受信するには、クライアントはnotification引数の‘enable’属性を‘true’に設定してこの操作を呼び出さなければならない。各クライアントの変更通知に関する当初の挙動は「通知せず」になっている。装置の通信状態が有効である場合に、このような通知を受信するクライアントをもはやアクセスできないと判断できるならば、装置は自動的にクライアントへの通知を不可能にしなければならない。この場合、クライアントが通知を再度、可能にするにはこの操作を呼び出さなければならない。

9.1.2.10.3 与えられたnotification引数の‘enable’属性が‘true’である場合、メタデータの変化が検出されると、装置は、クライアントがこの操作を呼び出した後に最初に検出された変化から直ちに、クライアントへの通知を（すでに実行中でなければ）開始すること。装置の通信が出来ない状態である時に変化が検出された場合、装置は通信できるようになった最初の機会を捉えて、対象となるクライアントに通知すること。

9.1.2.10.4 与えられるnotification引数の‘enable’属性が‘false’である場合、装置がクライアントへの変更通知の送信をその時に実行中であった場合は、送信を中止すること。同じクライアントから重複する有効／無効要求があった場合は、装置はそれを無視すること（§ 3 の制限を参照すること）。

MetadataRevision
revisionDate

図 5
Metadata Revision クラス

9.1.2.11 MetadataRevision — このクラスは最終のメタデータの変更の日時を示す。

9.1.2.11.1 MetadataRevision 属性定義表

表 17 MetadataRevision 属性定義

属性名	定義	形式
revisionDate	装置のメタデータが変更された最終の日時。	テキスト、ISO 8601 に従ったフォーマット。実装により 8601 に加えてフォーマットの制約を加える場合がある場合がある。

NotificationRequest
enable

図 6
Notification Request

9.1.2.12 NotificationRequest — このクラスはメタデータのクライアントの(変更発生時の)通知の選択を示す。

9.1.2.12.1 NotificationRequest 属性定義表

表 18 NotificationRequest 属性定義

属性名	定義	形式
enable	変更通知に関するクライアントの選択。	論理値。True ならばクライアントの選択はメタデータの変更時通知。False ならばメタデータ変更時非通知。

9.2 Metadata Client

9.2.1 図 7 はクライアントがメタデータ変更通知を受け取るためにクライアントがサポートすべきインタフェースと引数のデータ型を示す。

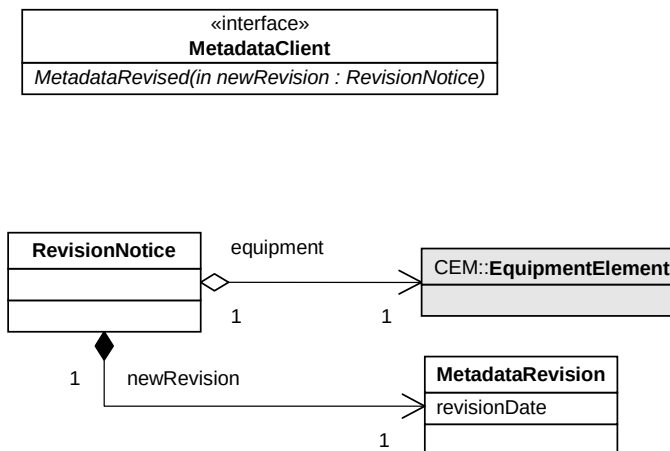


図 7

Metadata Client インタフェース

9.2.2 MetadataClient の操作

表 19 MetadataClient の操作定義

操作	説明	型
MetadataRevised	メタデータが最近変わったことをクライアントに通知する。	FF

9.2.2.1 MetadataRevised — クライアントがEquipmentMetadataManagerインタフェースの“NotifyOnRevisions”操作で前もって通知するよう、要求していれば、装置はメタデータの変更が判ったときに通知を送る。

9.2.2.1.1 MetadataRevised 操作の引数

表 20 MetadataRevised 引数の定義

引数	説明	種類	形式
revision	装置がメタデータの変更に気付いた日時。	in	9.2.2.2にあるような RevisionNotice 型の構造体をしたデータ。

9.2.2.2 RevisionNotice — このクラスはメタデータが変更された装置ならびにメタデータ変更の日時をもたらす。

9.2.2.2.1 RevisionNotice の関連定義表

表 21 RevisionNotice の関連定義

関連の役割名	定義	コメント
equipment	装置を全体として表すルート装置ノード。	SEMI E120 の EquipmentElement 型から派生する要素で、ただ 1 つだけとする。
newRevision	メタデータが変更された日時。	MetadataRevision 型の構造体をしたデータ。

9.2.2.2.2 Equipment — 装置のルートノードをRevisionNoticeにあてがうことが、装置のルートノード のメタデータの変化を意味すると解釈してはならない。これは、どの装置に変更があったかをクライアントが決定する手段に過ぎないものとして用意される。

10 装置のメタデータ

10.1 装置のメタデータの管理

10.1.1 装置構成の変更の結果、本文書の9.1.2.9 またはSEMI共通装置モデルのクラスを使用して説明される情報に変更がある場合は、装置は変更時にそれに従ってメタデータの内容を更新しなければならない。クライアントが変更通知を要求した場合(9.1.2.9 および9.1.2.10 参照)、装置は変更時もしくは通信ができる状態になり次第、求めたクライアントに通知する。これには、工場に導入された後の装置の構成変更が含まれる。確実に装置メタデータをすべての観点で最新の状態にして、工場が使用できるようにすることが、装置メタデータを維持するという点で重要である。

10.2 装置の構造

10.2.1 装置の物理的な構造は、SEMI共通装置モデルに従って記述しなければならない。サプライヤは、(少なくとも)稼働率トラッキング、プロセス制御、技術サポートが可能となるように、必要かつ実用的な範囲について装置の構成要素のモデルを提示しなければならない。

10.3 装置ノード

10.3.1 装置ノードの説明

10.3.1.1 図 8 に装置階層構造上の要素(即ちノード)で使用可能なパラメータ、イベント、ObjType、例外事象を言い表す方法を示す。これらが説明できることにより、注目するノードが、出し得るイベント、例外事象、ObjType、パラメータのすべての起源とみなされることを示している。ある装置ノードの関連するパラメータ、例外事象、ObjType、イベントは何れも、他の装置ノードに関連するイベント、例外事象、objType、パラメータとは独立している。

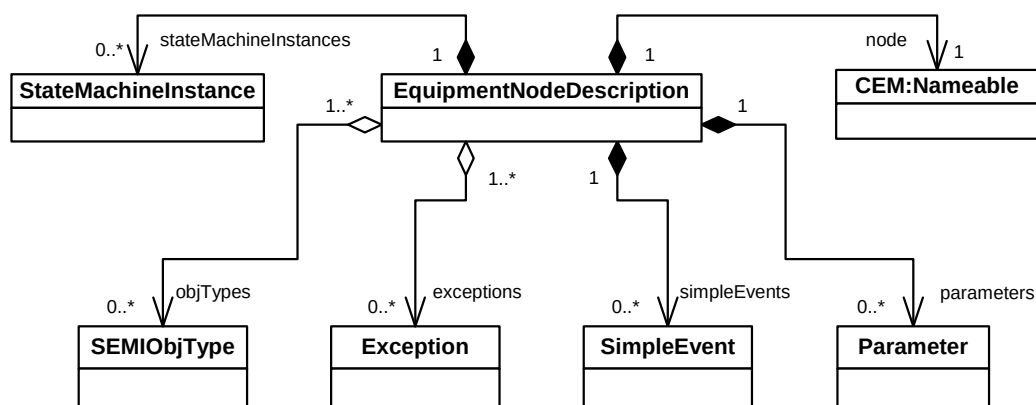


図 8
装置ノードの記述

10.3.1.2 *EquipmentNodeDescription* — このクラスを使用して装置の階層構造の個々の要素を特定し、そのノードを起源とする StateMachineInstances, パラメータ, 例外事象, ObjType, SimpleEvents を言い表す。これは、StateMachineInstance クラス(10.8.6), SEMIObjType クラス(10.8.5), Exception クラス(10.7), Parameter クラス(10.4) および SimpleEvent クラス(10.10) を使って行われる。ある装置ノード 1 つに EquipmentNodeDescription は 1 つだけが用意されなければならない。装置エレメントは、SEMI E120 の規定の共通装置モデルで定義されているように、Nameable を継承する何らかの具象クラスでよい。

10.3.1.3 EquipmentNodeDescription 関連定義表

表 22 EquipmentNodeDescription 関連定義

関連の役割名	定義	コメント
stateMachineInstances	この装置ノードによってサポートされる stateMachineInstances の順不同のリスト。	型 StateMachineInstance を要素とするリストであり、¶10.8.6 で説明する。パラメータがないイベントも含め、ノードが発生し得る各々のイベントにマップが 1 つずつ用意されること。
objTypes	この装置ノードに関連付けられる SEMIObjType の順不同のリスト。	型 SEMIObjType を要素とするリストであり、¶10.8.5 で説明する。
exceptions	この装置ノードによって報告できる例外の順不同のリスト。	¶10.7 で説明する Exception を要素とするリスト。
parameters	この装置ノードによって提供される型の区別のあるパラメータの順不同のリスト。	型 Parameter を要素とするリストであり、¶10.4 で説明する。SEMI スタンドardsの変数を表現するパラメータはすべて、¶10.5.4 および¶10.4.10 に従い型と名前を付けること。各パラメータ名は、このノードの中では一意であること。パラメータ名は、装置のノード全体にわたり一意である必要はない。
node	記述する装置ノード。	SEMI E120 の Nameable クラスを継承する型のエレメント。
simpleEvents	この装置ノードによって報告できる(状態マシンなし) simpleEvents の順不同のリスト。	¶10.10 で説明する simpleEvent を要素とするリスト。

10.4 パラメータ

10.4.1 パラメータは、¶10.5 に記述される任意の基本型または複合型で記述できる、何らかの概念を表現できるものである。パラメータは一般に、名称があり、特定の意味をもつ型の決まった値の保持をし、種々の用途で、あるいは他に情報量を持つものと見なすことができる。例としては、プロセス制御変数、装置構成設定、測定結果、ウェーハマップ、データ変数等がある。パラメータは、装置ノード(¶10.3)、SEMI ObjType (¶10.9)、例外事象(¶10.7)によってのみ定義できる。

10.4.2 名称があり、単位を持ち、パラメータの型の区別があり、制約事項があり、他のパラメータとの関連があるパラメータがどう記述されるのかを図 9 に示す。型と単位の情報 は DefinedType(ParameterType が正しい)クラスを参照する ParameterTypeDefinition で示される。ParameterTypeDefinition については、¶10.5.1 で説明する。

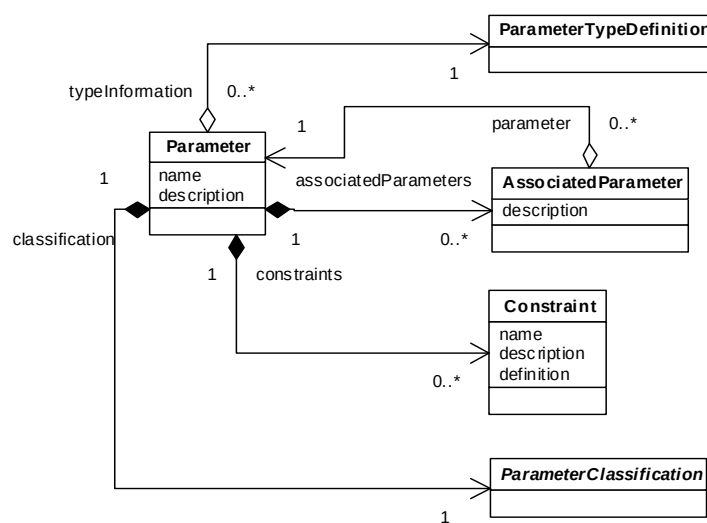


図 9
型の区別のあるパラメータの記述

10.4.3 サプライヤは、(少なくとも)稼働率トラッキング、プロセス制御、技術サポートの目的に備えるために必要かつ現実的な範囲のパラメータのモデルを用意しなければならない。

10.4.4 *Parameter*の属性定義表

表 23 *Parameter* の属性定義

属性名	定義	必須	形式
name	パラメータの名称。	Y	テキスト。
description	人間に分かるパラメータの目的や使用方法の記述。	Y	テキスト。

10.4.5 *Parameter*の関連定義表

表 24 *Parameter* の関連定義

関連の役割名	定義	コメント
constraints	このパラメータに関連する制約事項の記述。	表 26 に記述される <i>Constraint</i> 型の構造体データからなるリスト。複数の制約事項が記述される場合は、制限事項間に論理積が適用される。
associatedParameters	このパラメータに関連するシステム内の他のパラメータのリスト。	¶10.4.7 に記述される <i>AssociatedParameter</i> 型の構造体データからなるリスト。
typeInformation	このパラメータの型と単位を説明する型の定義を指す。	<i>ParameterTypeDefinition</i> 型の唯一の要素。
classification	パラメータの性質を説明する。	¶10.4.11.2 に記述される <i>ParameterClassification</i> から収集されるあらゆる型の唯一の要素。

10.4.6 各パラメータは、0 個以上の関連するパラメータを示すことができ、人が読むことができるその関係の説明を与える。

10.4.7 *AssociatedParameter* — このクラスを使用して、装置の階層構造内のどこかにある別のパラメータとの一般的な関連付けを記述できる。そのような関連付けが作成されると、これは、定義中のパラメータがこのクラスによって参照されるパラメータに何らかの関係があることを示す。関係の意味は、*AssociatedParameter* クラスの属性に書き入れられる。更に深い意味は、¶10.4.11 で説明する型の分類で結びつくパラメータに譲る。

10.4.7.1 *AssociatedParameter*の属性定義表

表 25 *AssociatedParameter* の属性定義

属性名	定義	必須	形式
description	定義されたパラメータとこの関連で指定されるパラメータとの関係の記述。	Y	テキスト。

10.4.7.2 *AssociatedParameter* 関連定義表

表 26 *AssociatedParameter* 関連定義

関連の役割名	定義	コメント
associatedParameters	このパラメータと関連するパラメータのリスト。	パラメータ型の構造体をしたデータを適当な順序でリストにまとめたもの。

10.4.8 *Constraint* — このパラメータの関連の制約事項を表現する。制約事項を使用してパラメータの有効な値の制限、装置外部への報告用にパラメータを収集できる速度の制限やパラメータの値や使用方法に関するその他の類似の制限を記述できる。

10.4.8.1 *Constraint*の属性定義表

表 27 Constraint の属性定義

属性名	定義	必須	形式
name	制約事項の名称。	Y	テキスト。
description	制約事項の目的や意味の記述。	Y	テキスト。
definition	制約の定義。	Y	テキスト。¶10.4.9 に従ってフォーマット。

10.4.9 制約の定義

10.4.9.1 パラメータの制約は、パラメータの有効な値の制限を記述する。制約は、有効なパラメータ値でTRUEと評価する命題式として文書化できる場合がある。関係代数では、命題式のための多数の構文が定義されているが、この構文は商用の用途にはあまり使用されない。標準照会言語 (SQL) は、関係代数の商用の用途にうまく文書化されており、この用途によく適する。特に、(SQL SELECT文の) **WHERE**節の構文は、パラメータ値の制約の記述に適している。SQL SELECT文の**WHERE**節は、1 つまたは複数の値と制約と比較する命題式である。次のような形式となる。

WHERE 命題式[ネストした命題式];

本文書の目的では、SQL WHERE 構文(構文全体は、関連情報のR2-1で述べる)のサブセットは制約を定義するために使用されなければならない。特にIN, BETWEEN, IS演算は制約の定義ではサポートされない。

10.4.9.2 例えば、100 を下回ったり、200 を超えてはならない温度設定値を表現する “TemperatureSetpoint” という名前のパラメータは、次のSQL **WHERE**節を使用して記述される。

WHERE TemperatureSetpoint > 100 **AND** TemperatureSetpoint < 200;

10.4.9.3 報告頻度の制約の記述

10.4.9.3.1 任意のパラメータを報告し、データ取得できる速度を記述するには、¶10.4.9.1 に説明されるSQL **WHERE**節のサブセットを使用したある種の表記法を使用しなければならない。検討事項であるパラメータについて、“ReportingPeriod”という名前の制約を使うだろう。ReportingPeriod制約の定義は、次の形式を有する。

WHERE parameterName.ReportingPeriod *comparison*;

10.4.9.3.2 ここで、*parameterName*は、制約が適用されるパラメータの名前であり、*comparison*は、パラメータの有効な報告間隔を記述するために必要な項、演算、制約の正しい構文による組み合わせである。報告間隔を参照する制約の時間刻みは秒である。制約の定義を使用して基本報告間隔の整数倍を表現する場合は、変数 ‘n’ を使用して乗数を表現しなければならない。

10.4.9.3.3 例えば次の制約は、“Temperature”にサポートされる報告間隔が 0.01 秒から 60 秒までの任意の連続な値であることを指定する。その他の例については、関連情報の ¶ R4-1 に記載する。

WHERE Temperature.ReportingPeriod >.01 **AND** Temperature.ReportingPeriod < 60;

10.4.10 SEMIスタンダードの変数や属性の記述

10.4.10.1 サプライヤがParameterを使用してSEMIスタンダードで定義されている変数や属性を表現する場合は、Parameterクラスのname属性は、対応するスタンダードに記載されている同じ場合の表記法を使用し、正確に同じスペルの変数名や属性名と等しくなければならない。¶10.5.4 に記述する型のマッピングによってParameterに割り当てる型が決まらなければならない。

10.4.11 パラメータの分類

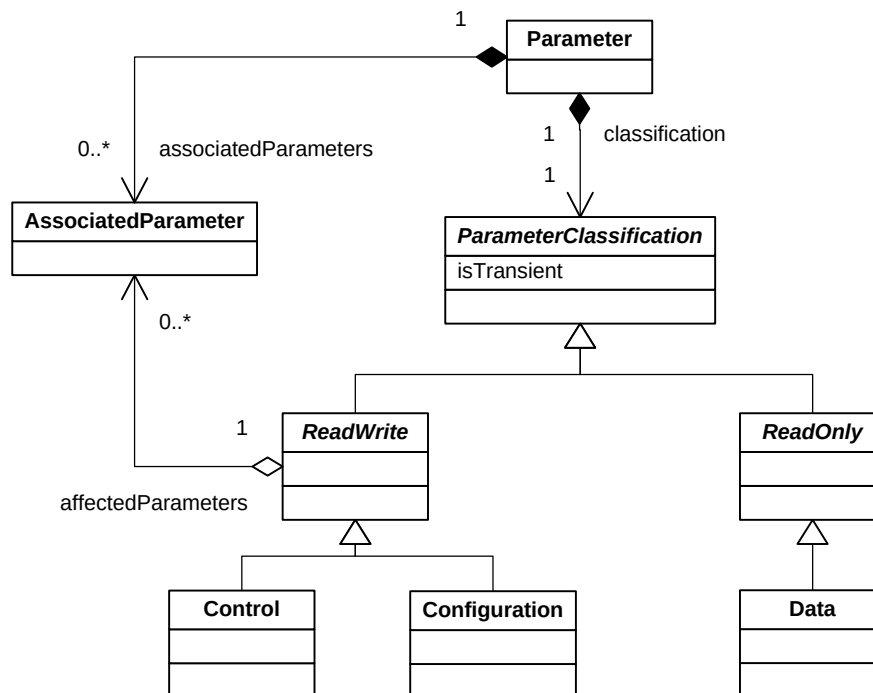


図 10
パラメータの分類

10.4.11.1 パラメータの性格を示すために使用される分類体系を図 10 に示す。2 つの主な分類、ReadOnly と ReadWriteがあり、パラメータ値が外部のエンティティによって設定できるかできないかを指定する。ReadWriteの場合、パラメータの機能を更に明細に決めるサブクラスが複数ある。

10.4.11.2 *ParameterClassification* — このクラスはParameter分類すべてに渡っての基本抽象クラスである。

10.4.11.3 *ParameterClassification* 属性定義表

表 28 *ParameterClassification* 属性定義

属性名	定義	形式
isTransient	パラメータ値がアクセス可能なだけ、もしくは、ある状態下で意味があるか否か。	論理値。¶10.4.11.3.1 参照。

10.4.11.3.1 isTransient

10.4.11.3.1.1 偽である場合、装置が使用可能な状態にあり外部のエンティティと通信可能な状態にある時は、パラメータの値はアクセス可能で意味のある値であることが保証される。例えば、SEMI E30 の装置定数および状態変数に対応するパラメータは一過性のものではない。

10.4.11.3.1.2 真である場合、パラメータは「一過性の値をとるもの」と言われる。一過性の値をとるパラメータの値のアクセス可能性は保証されず、特定の状況下以外は意味のある値を取らない。例えば、SEMI E30 に対応する「データ値」は一過性なものである。さらに、SEMI E39 の一時的なObjTypeの属性を表現でき、その場合のアクセス可能性はそのObjTypeのライフサイクルによって決まる。パラメータは、装置（またはSEMI E39 のObjType）が特定の状態にある時に限り使用可能なデータが提供されるものもある。一過性の値をとるパラメータが特定のイベントによる報告に使用可能であれば、その値は、装置がそのイベントが報告された遷移後の状態にある時にだけ有効値を取る。あるいは、パラメータ値はそのイベントが発生した時にだけ有効値となる。メタデータを使って一過性の値をとるパラメータのライフサイクルを説明するメカニズムを述べることは、本仕様の範囲外である。