

SEMI E54.1-0708 (Reapproved 0413) STANDARD FOR SENSOR/ACTUATOR NETWORK COMMON DEVICE MODEL

センサ／アクチュエータネットワーク共通デバイスモデルのスタンダード

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

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

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

本スタンダードは、global Information & Control Technical Committee で技術的に承認されている。現版は 2012 年 12 月 20 日、global Audits and Reviews Subcommittee にて発行が承認された。2013 年 4 月に www.semiviews.org および www.semi.org で入手可能となる。初版は 1999 年 2 月発行。前版は 2008 年 7 月発行。

注意: 以前本文書は SEMI E54 として存在していたものである。本文書は SEMI E54 の一連のスタンダードであり、その関連性を明確にするため SEMI E54.1 とされた。

注意: 本文書は、編集上の修正を伴い、再承認された。

1 目的

1.1 このスタンダードでは半導体装置センサ／アクチュエータネットワーク上にあるすべてのデバイスに共通したデバイスオブジェクトで構成されているモデル(Common Device Model)が定義されている。

2 範囲

2.1 本書は共通デバイスの構造および挙動(つまり、ネットワーク上で作動するためにすべてのデバイスがサポートしなければならない最少限のデータおよび挙動)について記述している。デバイスは簡単なセンサやアクチュエータからホスト、マスター、あるいはコントローラにまでわたっている。

2.2 本書で規定されているモデルは、個々のデバイスのデータ構造と挙動の特徴を説明してある、センサ・アクチュエータネットワーク個別デバイスモデル(Specific Device Model)に使用されている。ネットワークインタフェースから見たデバイスを完全に記述するにはこれら 2 つのモデルで十分である。

2.3 このスタンダードは、センサ／アクチュエータネットワーク相互運用性(インターオペラビリティ)ガイドラインとセンサ／アクチュエータネットワーク通信仕様書、および 1 つあるいはそれ以上の個別デバイスモデル仕様書により、完全な相互運用性仕様書となる。センサ／アクチュエータネットワーク文書の構造は図 1 に示されている。

2.4 このスタンダードに準拠するには、特に明示されていない場合を除き、デバイスは本書で決められているオブジェクトのインスタンス、オブジェクト属性、オブジェクトサービス、およびオブジェクトの挙動を実装し、サポートしなければならない。

注意: このスタンダードおよび安全ガイドラインはその使用に関連した全ての安全問題を取り扱うことを意図していない。この文書の使用者は、その責任において、適切な安全および健康上実施すべき事柄を確立し、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 このスタンダードは一連のセンサ／アクチュエータネットワーク通信ネットワーク仕様書への手引書である。このため、このスタンダードの中でネットワーク通信に関連している部分を使うには、必然的に関連ネットワーク仕様を理解することが必要となる。

3.2 本書は共通デバイスモデルのためのスタンダードであり、特定のデバイスあるいはデバイスタイプに関連した情報(例えばオブジェクト、属性、サービス、あるいは挙動の説明)は含まれていない。

3.3 図 1 に描かれているスタンダードはネットワークから見たデバイスを完全に記述するには十分だが、ネットワークから見えないデバイスの挙動を十分に記述してはいない。スタンダードオブジェクト内の詳細な挙動の中には

メーカーが定義できるよう、故意に残されているものもある。そのため製品を柔軟に差別化でき、技術が進歩する余地が生み出されている。メーカー固有のオブジェクトは適宜メーカーが定義できるが、当然このスタンダードの範囲外である。

3.4 このスタンダードは SEMI E39 との互換性を持っているがそれに準拠していない。つまり、このスタンダードは SEMI E39 への準拠を要求してはいないが、両方のスタンダードに完全に準拠するように開発できるインプリメンテーションに拡張することが可能である。このスタンダードのコンセプトと用語は SEMI E39 のものと互換性を持っている。しかし、SEMI E39 はより高いレベルのアプリケーションのための固有の必要条件を持っていて、そのため共通デバイスモデルには適用されない。

4 参照スタンダードおよび文書

4.1 SEMI Standards and Safety Guidelines

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

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

SEMI E148 — Specification for Time Synchronization and Definition of the TS-Clock Object

4.2 ISO Standard¹

ISO 7498 — Basic Reference Model for Open Systems Interconnection

4.3 Other Documents

David Harel, “Statecharts: A Visual Formalism for Complex Systems,” Science of Computer Programming 8, 1987

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

注意：別途指示のない場合、引用された全ての文書は最新の出版バージョンである。

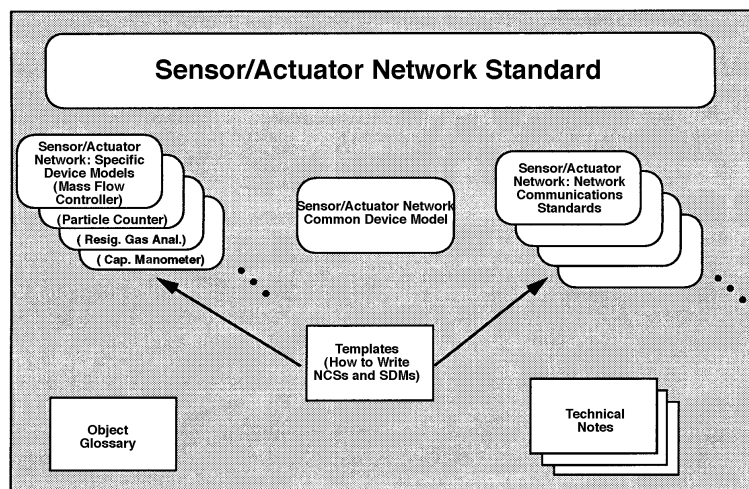


図 1

センサ/アクチュエータネットワーク参照文書

5 用語

5.1 他の SEMI 文書で定義されている用語をここに示す。

5.2 デバイスコンポーネント用語

5.2.1 属性 (attribute) — 外部から目にみえる、オブジェクトに関する情報。

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>

5.2.2 挙動(behavior) — オブジェクトがどのように行動するかについての規定。オブジェクトが検出する異なったイベント、例えばサービス要求の受信、内部の故障、あるいはタイマーの時間切れなどの結果としてアクションが発生する。

5.2.3 クラス(class) — オブジェクトの特定のタイプあるいは分類。

5.2.4 デバイス(device) — (1)少なくとも 1 個のセンサやアクチュエータ、またはコントローラ、(2)本書で規定されているような方法でネットワークに対して 1 ポイントでのアクセスをサポートしている通信コントローラ、(3)本書で詳細に述べてある仕様に準拠する性能を持ったシステムに(1)と(2)をまとめたものを提供する相互接続および管理ハードウェアおよびソフトウェア、で構成されている有形物。デバイスの例は図 2 に描かれている。

5.2.5 デバイスモデル(device model) — デバイスを抽象化したもので、デバイスを作ったり使ったりする前にそれを理解するために使用する。

5.2.6 エンベデッドオブジェクト(embedded object) — エンベデッドオブジェクトは、機能あるいは目的においてそれを内蔵しているオブジェクトと類似している、あるいはそれを内蔵しているオブジェクトの機能をサポートする。内蔵している構造は文書を構築して理解する目的のためにだけ利用される。そのため、これには直接的な関係、継承、構造における類似性、あるいは、エンベデッドオブジェクトとそれを内蔵しているオブジェクトとの間のアドレス対応法における連結性は含まれていない。

5.2.7 インスタンス(instance) — オブジェクトの基本的な実態。

5.2.8 メーカー(manufacturer) — 本文書ではデバイスのメーカーを指す。

5.2.9 オブジェクト(object) — 特定のデータおよび挙動のセットを持った実体。オブジェクトは物理的な存在でも、概念的な存在でもよい。オブジェクトは、その属性、それが提供するサービス、およびそれが示す挙動に関して記述できる。

5.2.10 サービス(service) — オブジェクトが提供する、あるいはサポートする機能。サービスは、パラメータのリストに一つずつ書かれているサービスの基本要素(プリミティブ)形のシーケンスで構成されている。サービスにはメッセージ構造およびプロトコルの定義は含まれない。

5.2.11 状態図(state diagram) — 状態遷移を表わす方法で、四角は状態を表わし、矢印は状態間の遷移を表わす。

5.3 データタイプ用語(Data Type Terminology) — 特記しない限り、SEMI E39 に定義されているデータタイプ用語が本書でも使用される。下記の用語も使用される:

5.3.1 ブーリアン(Boolean (BOOL)) — 0 と 1 を表すバイナリビットで、それぞれ FALSE (偽) および TRUE (真)、あるいは DISABLE (ディスエーブル) および ENABLE (イネーブル) に対応する。

5.3.2 バイト(byte) — 隣り合った 8 つのビットの列で、単位として解釈され、キャラクタを表わすことも多い。

5.3.3 キャラクタ(character) — テキスト記号、文字、桁、あるいはマークで、長さ 1 バイトの情報を表現し、コントロールし、まとめるのに使われる。

5.3.4 キャラクタ文字列(character string) — テキスト文字列。

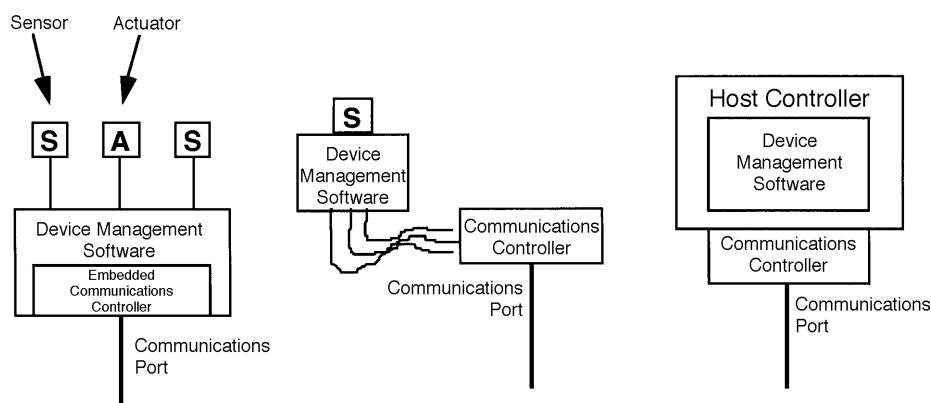


図 2
デバイスの例

5.3.5 データタイプ(*data type*) – 属性データフォーマットを規定するための、列挙バイトとしてフォーマットされている、符号なしの短い整数。この属性タイプは、属性あるいは属性のセットが定義されていて、一つ以上のレベルで(例えば INT あるいは REAL で)サポートできる場合に使われることを所期の目的としている。下記の値が定義されている:

- 0 = INT
- 1 = REAL
- 2 = USINT
- 3 = SINT
- 4 = DINT
- 5 = LINT
- 6 = UINT
- 7 = UDINT
- 8 = ULINT
- 9 = LREAL
- 10～99 = CDM のために保留
- 100～199 = SDM のために保留
- 200～255 = メーカーが指定する

5.3.6 データユニット(*data units*) – 属性データユニットを規定するための列挙された符号なし整数。この属性タイプは、属性あるいは属性のセットが定義されていて、一つ以上のユニットのコンテキストで使われることを所期の目的としている。値は本文書の付属書 1 に定義されている。

5.3.7 日付(*date*) – カレンダーでの日付を表すのに使われる 4 バイトのデータ構造。表 1 は日付のデータタイプのフォーマットを定義している。

表 1 日付のフォーマット

データ#	説明	範囲
0 ~ 1	年	符号なし整数 < 65,536
2	月	符号なし整数 (1 から 12 の範囲)
3	日	符号なし整数 (1 から 31 の範囲)

5.3.8 二重整数(*Double Integer (DINT)*) – 整数, 4 バイト長, 範囲は -2^{31} から $2^{31}-1$ 。

5.3.9 列挙式バイト(*enumerated byte*) – 0 から 255 間での値に意味が割り当てられているバイト。限られた数の取り得る値で構成されているセットの中の一つをとる場合がある。

5.3.10 フルスケールレンジ(*full scale range*) – 割り当てられた単位で属性に定義された値の 100% 値。この値は必ずしも属性の最大値となるわけではない。例として、フロー属性値で示された値はフルスケールレンジの 120% になる場合がある。

5.3.11 最終有効値(*Last Valid Value (LVV)*) – 最も最近、属性に割り当てするのに成功した値。

5.3.12 長整数(*Long Integer (LINT)*) – 整数, 8 バイト長, 範囲は -2^{63} から $2^{63}-1$ 。

5.3.13 長実数(*Long Real (LREAL)*) – IEEE 754 に定義されているとおりの、二重浮動小数点の数, 8 バイト長。

5.3.14 ニブル(*nibble*) – 隣り合った 4 つのバイナリビットの列。

5.3.15 ヌルキャラクタ(*null character*) – 値がゼロのバイト。

5.3.16 実数(*Real (REAL)*) – IEEE 754 に定義されているとおりの、浮動小数点の数, 4 バイト長。

5.3.17 短整数(*Short Integer (SINT)*) – 整数, 1 バイト長, 範囲は -128 から 127。

5.3.18 符号付き整数(*Signed Integer (INT)*) – 整数, 2 バイト長, 範囲は -32768 から 32767。

- 5.3.19 テキスト文字列(text string) — 1 バイトキャラクタの文字列。
- 5.3.20 符号なし二重整数(Unsigned Double Integer (UDINT)) — 符号なし整数, 4 バイト長, 範囲は 0 から $2^{32}-1$ 。
- 5.3.21 符号なし整数(Unsigned Integer (UINT)) — 整数, 2 バイト長, 範囲は 0 から 65535。
- 5.3.22 符号なし長整数(Unsigned Long Integer (ULINT)) — 符号なし整数, 8 バイト長, 範囲は 0 から $2^{64}-1$ 。
- 5.3.23 符号なし短整数(Unsigned Short Integer (USINT)) — 整数, 1 バイト長, 範囲は 0 から 255。

6 規則

6.1 ハレル状態モデル — 本書では定義されているオブジェクトの動的な挙動を記述するのにハレル状態図を使用している。この表記法の概要は SEMI E30 の付属書に書かれている。この表記法の正式な定義は D.ハレル著 Science of Computer Programming 8, "Statecharts: A Visual Formalism for Complex Systems," 1987 に書かれている。

6.1.1 遷移表(本書では「状態遷移マトリクス」と呼ばれている)が状態図といっしょに掲載されており, 各状態遷移の性質を明確に記述している。各遷移表には遷移#, 現在の状態, トリガー, 新しい状態, アクションのカラムがある。遷移の「トリガー」(カラム 3)は「現在の」状態にある時に発生する。「アクション」(カラム 5)には次の 3 つが組み合わされたものが入っている:(1)現在の状態を出た時に取るアクション, (2)新しい状態に入った時にとるアクション, (3)遷移に最も密接に関連しているアクション。区別はされていない。

6.2 OMT オブジェクト情報モデル — オブジェクトモデルはランボー, ジェイムズ, その他が"Object-Oriented Modeling and Design," Prentice Hall, Englewood Cliffs, NJ, © 1991 で開発したオブジェクトモデル作成テクニックを使って描かれている。この表記法の概要は SEMI E39 の付属書に書かれている。

6.3 オブジェクト属性の表記 — スタンダード化されたオブジェクトのためのオブジェクト情報モデルは, 下記のカラムの見出しがついた属性定義表でサポートされている:

属性	定義	アクセス	必要条件	形式
属性の正式なテキスト名	含まれている情報の説明	RO あるいは RW	Y あるいは N	(下記参照)

6.3.1 アクセスのカラムでは RO(読み込みのみ)あるいは RW(読み込みおよび書き込み)が使われていて, オブジェクトサービスのユーザが属性に対して持っているアクセスを表わしている。本書ではアクセスのカラムはネットワークを介してのユーザアクセスを表わすためだけに使用されている;このレベルの仕様はインプリメンテーションごとに異なるとみなされているので, ローカルアクセスについては何も書かれていない。このように, 本書のコンテキストでは, RW 属性はネットワークでセットできる属性(ネットワークから値を変更できる属性)であり, 一方, RO 属性はネットワークでセットできない属性である。必要条件のカラムの「Y」あるいは「N」は, サービスのための基本的な準拠を満たすためにはこの属性がサポートされていないかならないかどうかを表わしている。

形式のカラムは属性のフォーマットを表わすのに使われている。

6.4 サービスメッセージの表記

6.4.1 サービスリソースの定義 — サービスリソースの定義表では, 下記の表のように, あるサービスグループのための特定のメッセージを定義している:

サービス	タイプ	説明
メッセージ名	N あるいは R	サービスの目的

6.4.1.1 タイプは通知(N)か, あるいは要求(R)のどちらかである。通知メッセージはサービスの提供者(プロバイダ)が送出する。応答は期待されていない。要求メッセージはサービスユーザが送出する。要求メッセージはデータあるいはあるオペレーションの実行を依頼する。要求メッセージは特定の応答を期待する(メッセージの内容は定められている)。

6.4.2 サービスパラメータディクショナリ — 各パラメータはオブジェクトモデルからの属性か、あるいは動的モデル(ハレル状態図)のイベントのどちらかに関係していなければならない。サービスのパラメータはすべて 1 つの表あるいはディクショナリにリストアップされている。このパラメータディクショナリのカラムの見出しは下記の通りである:

パラメータ	形式	説明
パラメータ X	データタイプ	パラメータ

6.4.2.1 サービスのパラメータ一つ一つに表の 1 列が与えられている。最初のカラムにはパラメータの名前が書かれている。このあとに形式とパラメータの内容を記述したカラムが続いている。形式のカラムはパラメータに含まれているデータのタイプと、それが取り得る値を表わしている。説明のカラムはパラメータの意味と、それ以外のパラメータとの相互関係を記述している。

6.4.2.2 サービスメッセージの定義 — パラメータがはっきり指定されているサービス一つ一つについてパラメータの詳細を表わした表がある。サービスの詳細のカラムの見出しは下記の通りである:

パラメータ	要求/指示	応答/確認	説明
パラメータ X	(下記参照)	(下記参照)	サービスの説明

6.4.2.2.1 要求/指示および応答/確認と書いてあるカラムは、パラメータをメッセージの方向と結び付けている。イニシエータが送信するメッセージは「要求」と呼ばれる。レシーバはそのメッセージを「指示」と呼ぶ。レシーバは今度は「応答」を送信し、それを最初のセnderは「確認」と呼ぶ。

6.4.2.2.2 要求(要求/指示)および応答(応答/確認)には下記の値が入る:

"M"	必須パラメータ: 必ず有効な値が与えられなければならない
"C"	条件付きパラメータ: ある状況では定義され、それ以外の時には定義されない。値が与えられるかどうかは完全にオプションであり、もう一つのパラメータの値次第である。
"U"	ユーザ定義パラメータ
"—"	パラメータは使用されていない
"="	応答に書かれている M および C は、応答が要求の値と一致しなければならないということを表わすために、(=)で修正できる

7 デバイスのハイレベルな構造

7.1 デバイスを集めたハイレベルなオブジェクト図は図 3 に描かれている。デバイスは sensor/actuator/controller (SAC) オブジェクト, device manager (DM) オブジェクト, そして少なくとも 1 つのアクティブな要素オブジェクト(つまりセンサあるいはアクチュエータあるいはコントローラオブジェクト)で構成されているものとして描かれている。これらのオブジェクト一つ一つは必然的に属性、サービスおよび挙動を持っている。「デバイス」オブジェクトが図 3 に描かれているのは、デバイスおよびそのコンポーネントオブジェクトをハイレベルな観点から図示するためだけである; 本書のコンテキストから言って、「デバイス」オブジェクトはアドレスラブルでない、アドレスラブルな属性もアクセス可能なサービスも持っておらず、定義された挙動もない。

7.1.1 本書では DM オブジェクトだけ詳細に定義してある。SAC オブジェクト, センサオブジェクト, アクチュエータオブジェクト, およびコントローラオブジェクトがここで定義されているのはすべてのデバイスに共通の特徴という観点からだけである。SAC オブジェクトの完全な定義には該当するセンサ/アクチュエータネットワーク個別デバイスモデルが含まれている。この、対になっているモデルは、デバイスモデルを指定するのにふさわしいセンサ, アクチュエータ, およびコントローラオブジェクトの定義を完成させることもできる。

7.2 一般的な必要条件 — オブジェクトはそのオブジェクト名とインスタンス識別子で定義されている。本書に書かれているオブジェクトすべての識別子は表 1 に要約されている。この識別子は、センサ/アクチュエータネットワークを介してオブジェクトインスタンスをリモートで尋ねるのに使うことができることに注意する(各デバイスのセンサ/

アクチュエータ通信仕様書を参照のこと)。また、表 2 では本書で指定されているオブジェクトの多くは、デバイスごとのインスタンス化がちょうど 1 つであることを示している。

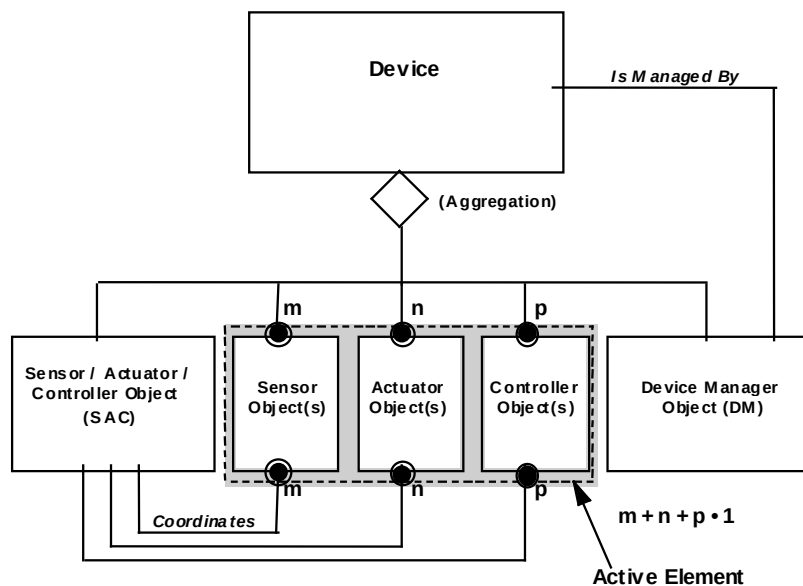


図 3
ハイレベルな観点から見たデバイスのオブジェクト

表 2 デバイスオブジェクトと識別子

オブジェクト名	オブジェクト識別子 (タグ)	デバイスで サポートが必須か	コメント
sensor/actuator/controller (センサ/アクチュエータ/コントローラ)	SacI0 ^{注1}	Yes	デバイス一つにつきインスタンスは一つ だけ許可
sensor (センサ)	SenIn ^{注2}	No	デバイス 1 つにつきゼロあるいはそれ以 上のインスタンスが許される ^{注3}
actuator (アクチュエータ)	ActIn ^{注2}	No	デバイス 1 つにつきゼロあるいはそれ以 上のインスタンスが許される ^{注3}
controller (コントローラ)	CntIn ^{注2}	No	デバイス 1 つにつきゼロあるいはそれ以 上のインスタンスが許される ^{注3}
device manager (デバイスマネージャ)	DmI0 ^{注1}	Yes	デバイス 1 つにつきインスタンスが 1 つだ け許される

注 1 デバイス 1 つにつきオブジェクトインスタンスは 1 つだけ; 識別子は「インスタンスゼロ」を指定するのに「I0」を使用する

注 2 オブジェクトのインスタンス数を表すのに「In」が使われている; 「n」は負でない整数である。

注 3 デバイス 1 つについて許可されているセンサ、アクチュエータやコントローラオブジェクトインスタンスの数についての仕様は、それに該
当するセンサ・アクチュエータネットワーク仕様書でさらに制約を受けている場合がある。

7.2.1 本書に集合的に記述されているオブジェクトは、デバイスの機能について、ネットワークの相互運用性の
ためにそれがどのように構成されたかということを含め定義している。こういったオブジェクトインスタンスの属性
に含まれている情報はネットワークを使ってアクセスできなければならない、デバイスで保存されなければならない。

7.2.2 本書で記述されているキャラクタ文字列には規定の最大長がある。文字列の内容が規定の長さより短い
場合には、その文字列はヌルキャラクタ:1 バイトの値が 00h で終わらなければならない。このようにヌルで終了しな
ければならないという仕様は、ネットワークを使ってキャラクタ文字列を伝えているセンサ/アクチュエータネットワ
ークプロトコルインプリメンテーションがヌル終了を送信しなければならない、ということを表わしているのではない;

ネットワークを使つてのキャラクタ文字列データの提示は、センサ・アクチュエータネットワーク通信プロトコル固有のことである。

7.3 センサ／アクチュエータコントローラ(SAC)オブジェクト – SAC オブジェクトは、センサ／アクチュエータ／コントロール環境とのデバイスの相互作用を調整する責任を持ったデバイスコンポーネントである。SAC オブジェクトは図 4 に描かれている。SAC オブジェクトは、デバイスのセンサ／アクチュエータコントロール部分を集合的に構成している 1 つあるいはそれ以上のセンサ、アクチュエータやコントロールオブジェクトインスタンスのオペレーションを調整して、希望するデバイスの挙動をイネーブルにできるようにする。例えば、デバイスセンサ、アクチュエータやコントロール要素のオペレーションを調整して、デバイスレベルでのデータ報告あるいは始動、アラームの検出およびサービス実行、ステータス報告、デバイスの自己テスト実行、デバイスの運転停止などをイネーブルにすることができる。デバイス 1 つごとに許可されているセンサ、アクチュエータやコントローラオブジェクトインスタンスの数は、それに該当するセンサ／アクチュエータネットワーク個別デバイスモデルで規定される。作動しているデバイスには SAC オブジェクトのインスタンスが正確に 1 つだけはいっている。

7.3.1 SAC オブジェクトは、デバイスとセンサ／始動／コントロール環境との相互作用の協調に関連した特定のタスクを処理するオブジェクトを内蔵している。内蔵されているオブジェクトは表 3 にリストアップされている。デバイス一つにつき SAC オブジェクトのインスタンスは一つしか許容されていないが、デバイスは内蔵オブジェクトのインスタンスを複数持つことができる。そういったオブジェクトは § 7.3.5 に記されている。

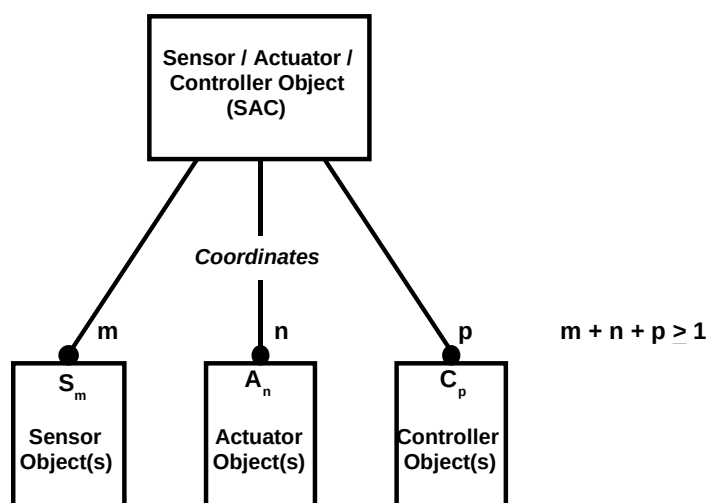


図 4
センサ／アクチュエータコントローラオブジェクトの詳細図

表 3 SAC オブジェクトに内蔵されているオブジェクトおよび識別子

オブジェクト名	オブジェクト識別子 (タグ)	デバイスでのサポートが必須か	コメント
Assembly	AsmIn ^{注1}	No	デバイス一つにつきゼロあるいはそれ以上のインスタンスが許容されている ^{注2}
Local Link	LnkIn ^{注1}	No	デバイス一つにつきゼロあるいはそれ以上のインスタンスが許容されている ^{注2}

注 1 "In"はオブジェクトのインスタンスナンバーを表すのに使われている。"n"は負でない整数。

注 2 デバイス一つにつき許容されている Assembly あるいは Local Link オブジェクトのインスタンスの数に関する仕様は該当するセンサ／アクチュエータネットワーク仕様でさらに制約される場合がある。

7.3.2 SAC オブジェクト属性 – 要求されている SAC オブジェクトおよび「共通オプション」SAC オブジェクトは全部表 4 にリストアップされている。属性の多くは「人間が読める」テキスト形式であることに注意する。この表で提供されている情報は属性ごとの、(1)属性識別子タグ、(2)(ネットワークを使つての)属性へのユーザアクセス(つまり属

性がネットワークで設定可能かどうか), および(3)属性タイプである。要求されている属性は SAC オブジェクトのすべてのインスタンス化でサポートされなければならない。オプションの属性はサポートしてもしなくてもよい。こういったオプションの構成のサポートに関する要求事項の仕様は § 7.3.4 項および該当するセンサ/アクチュエータネットワーク独自のデバイスモデル仕様に書かれている。

表 4 SAC オブジェクト属性

属性名	属性識別子	アクセスネットワーク	必須か	形式
Last Calibration Date	SacA1	RW	No	日付
Next Calibration Date	SacA2	RW	No	日付
Expiration Timer	SacA3	RW	No	符号付き整数
Expiration Warning Enable	SacA4	RW	No ^{注 1}	ブーリアン
Run Hours	SacA5	R	No	符号なし整数
Last Calibration Timestamp	SacA6	RW	No	日付と時間
Next Calibration Timestamp	SacA7	RW	No	日付と時間
Data Timestamp	SacA8	RW	No	日付と時間
Reserved	SacA9 ~ SacA64	–	–	将来の拡張のために保留

注 1 Expiration Timer 属性がサポートされているなら, Expiration Warning Enable 属性も要求される。

7.3.2.1 Last Calibration Date (オプション) — デバイスが前回校正された日付を特定する属性。属性は 5.2 項で定義されている日付としてフォーマットされる。

7.3.2.2 Next Calibration Date (オプション) — デバイスが次に校正されることになっている日付を特定する属性。属性は 5.2 項で定義されている日付としてフォーマットされる。

7.3.2.3 Expiration Timer (オプション) — 次の勧告校正日時までに残っているランの時間数を特定する属性。属性は解像度 1 時間の符号なし整数。

7.3.2.4 Expiration Warning Enable (オプション) — 校正時間切れタイマーが, DM オブジェクトインスタンスの Exception Status 属性に特定のワーニングステータスビット(特に「校正時間切れ」例外ビット, ¶7.4.1.15 参照)をセットするかどうかを指定する属性。

7.3.2.4.1 Expiration Warning Enable 属性はブーリアンで, 下記の値のどちらかを取り得る:

- 0=ディスエーブル
- 1=イネーブル

7.3.2.5 Run Hours (オプション) — デバイスの電源がオンになっていた時間数を特定する属性。属性は解像度 1 時間の符号なし整数。

7.3.2.6 Last Calibration Timestamp (オプション) — デバイスが最後に校正された日付と時刻を特定する属性。この属性は § 7.4.1.23 に定義される Data and Time Type タイプの通りフォーマットされる。

7.3.2.7 Next Calibration Timestamp (オプション) — デバイスの次回の校正が計画されている日付と時刻を特定する属性。この属性は § 7.4.1.23 に定義される Data and Time Type タイプの通りフォーマットされる。

7.3.2.8 Data Timestamp (オプション) — データ値が生成された時刻を特定する属性。この属性は § 7.4.1.23 に定義される Data and Time Type タイプの通りフォーマットされる。

7.3.2.9 初期値およびデフォルト値 — 関連する SAC 属性の初期値およびデフォルト値は表 5 に書かれている:

表 5 SAC 属性の初期値およびデフォルト値

属性	初期値	デフォルト値	コメント
Last Calibration Date	LVV	0,0,0 ^{注1}	
Next Calibration Date	LVV	0,0,0	
Expiration Timer	LVV	0	
Expiration Warning Enable	LVV	0	
Run Hours	LVV	0	

注1 年, 月, 日に対応する。§ 5.2 参照。

7.3.3 SAC オブジェクトサービス — SAC オブジェクト事例はすべて、表 6 にリストアップされているサービスを提供しなければならない(SAC オブジェクトはこういったサービスの「サービスプロバイダ」である)。SAC サービスは "SacSn" という形式の属性識別子で識別される。ここで "n" は正の十進数の数値である。SacS1 から SacS3 までの SAC サービス識別子がこの共通デバイスモデルスタンダードで利用されている。SacS4 から SacS32 までの SAC サービス識別子は保留されている。

表 6 SAC サービス

サービス	サービス識別子	必須	タイプ	説明
Reset (リセット)	SacS1	Yes	R	オブジェクトを INITIALIZING 状態にするのに使われる
Abort (アボート)	SacS2	Yes	R	オブジェクトを ABORT 状態にするのに使われる
Recover (リカバー)	SacS3	Yes	R	オブジェクトを ABORT 状態から RECOVERING 状態へ移動させるのに使われる
Get Attribute (属性取得)	SacS4	Yes	R	オブジェクト属性を読み込むのに使われる
Set Attribute (属性設定)	SacS5	Yes	R	オブジェクト属性を修正するのに使われる
Operate (作動)	SacS6	Yes	R	オブジェクトを INITIALIZING あるいは RECOVERING 状態から OPERATING 状態へ移動させるのに使われる
Restore Default (デフォルト回復)	SacS7	No	R	オブジェクト属性をデフォルト値に戻すのに使われる
Publish Attribute (属性発行)	SacS8	No	N	オブジェクト属性値を前向きに報告するのに使われる
	< SacS32	—	—	CDM の将来の拡張に備えて保留

7.3.3.1 Reset (サービス識別子:SacS1) — SAC オブジェクトを INITIALIZED 状態とするのに使われる。この状態の説明はデバイスによって異なる。このサービスに指定されているパラメータはない。

7.3.3.2 Abort (サービス識別子:SacS2) — SAC オブジェクトを ABORT 状態とするのに使われる。この状態の説明はデバイスによって異なる。このサービスに指定されているパラメータはない。

7.3.3.3 Recover (サービス識別子:SacS3) — SAC オブジェクトを ABORT 状態から RECOVERED 状態に移すのに使われる。この状態の説明はデバイスによって異なる。このサービスに指定されているパラメータはない。

7.3.3.3.1 デバイス固有あるいはインプリメンテーション固有の SAC オブジェクトが追加サービスやサービスパラメータを提供する場合がある。そういうサービスは通知タイプあるいは要求タイプのサービスである (§ 6.4.1 参照)。デバイス固有の追加サービスおよびサービスパラメータの完全な定義は各デバイスのセンサ/アクチュエータネットワーク個別デバイスモデル仕様書に書かれている。

7.3.3.4 *Get Attribute* — このサービスはネットワークでオブジェクトインスタンス属性の値を読み込むのに使われる。表 7 にはこのサービスに規定されているパラメータが書かれている。

表 7 SAC オブジェクトの Get Attribute パラメータの定義

パラメータ	要求／指示	応答／確認	形式	説明
属性 ID	M	C	ネットワーク独自	値が要求されている属性の属性識別子
属性値	—	M	コンテキスト独自	要求された属性の値

7.3.3.5 *Set Attribute* — このサービスはネットワークで修正可能であると特定されているオブジェクトインスタンス属性の値を修正するのに使われる。表 8 にはこのサービスに規定されているパラメータが書かれている。

表 8 SAC オブジェクトの Set Attribute パラメータの定義

パラメータ	要求／指示	応答／確認	形式	説明
属性 ID	M	C	ネットワーク独自	値が修正されている属性の属性識別子
属性値	M	C	コンテキスト独自	修正する属性の値

7.3.3.6 *Operate* — このサービスはオブジェクトインスタンスを **INITIALIZING** あるいは **RECOVERING** 状態から **OPERATING** 状態へ移動させるのに使われる。このサービスに規定されているパラメータはない。作動要求は S, A, および C オブジェクト全部に対して発行される。

7.3.3.7 *Restore Default* (オプション) — このサービスはこのオブジェクトインスタンスの属性をデフォルト値に戻すのに使われる。表 9 にはこのサービスに規定されているパラメータが書かれている。Restore Default 状況=2 の場合, Restore Default 状況=1 要求が S, A, および C オブジェクト全部に対して発行される。

表 9 SAC オブジェクトの Restore Default パラメータの定義

パラメータ	要求／指示	応答／確認	形式	説明
回復状況	M	C	バイト, 列挙	0=このオブジェクトの特定の属性を回復させる 1=このオブジェクトを回復させる 2=このデバイスのオブジェクト全部を回復させる
属性 ID	C	C	ネットワーク独自	値が戻されている属性の属性 ID。このパラメータは Restore Conditions=0 の場合にのみ使われる

7.3.3.8 *Publish Attribute* (オプション) — SAC オブジェクト属性を前向きに伝えるのに使われる。表 10 にはこのサービスに規定されているパラメータが書かれている。

表 10 SAC 属性発行サービスメッセージの定義表*

パラメータ	要求／指示	応答／確認 ^{注1}	説明
属性 ID	M	—	値が報告されている属性の属性識別子
属性値	M	—	報告されている属性の値

注 1 サービス応答メッセージおよびそのフォーマットの利用は本文書の範囲外である。

7.3.4 SAC インスタンスの挙動 — すべてのデバイスに共通の SAC オブジェクトの挙動は図 5 に描かれている。関連した状態および状態遷移は表 11 および 12 に記述されている。下記も SAC オブジェクトの挙動に適用される:

7.3.4.1 SAC オブジェクトサービスはその SAC オブジェクト内部から要求される場合がある。

7.3.4.2 SAC オブジェクトがインスタンス化された時, あるいは Reset 要求を受信した時, その SAC が **INITIALIZED** 状態に入るか, あるいはそこにとどまることにより応答する。

7.3.4.3 ABORT 要求を受信すると SAC オブジェクトは ABORT 状態に入るか、あるいはそこにとどまることにより応答する。

7.3.4.4 Recover 要求を受信すると、現在 ABORT 状態にいる時だけ、SAC オブジェクトは RECOVERED 状態に遷移して応答する。その他はエラーを示す。

7.3.4.5 デバイス固有の挙動を SAC オブジェクトが追加で示す場合がある。この、SAC オブジェクトの追加の挙動の完全な定義は各デバイスのセンサ／アクチュエータネットワーク個別デバイスモデル仕様書に書かれている。

7.3.4.6 すべての SAC サービスは特記されていない限り、無効サービス要求に関連した同一の挙動を持っている。共通サービスエラー応答は付属書 2 にリストアップされている。

7.3.4.7 ネットワークでセット可能でない属性について Set Attribute 要求がネットワークで受信されたら、あるいは、Set Attribute 要求が指定された属性の値に許容された指定範囲外だったら、属性値は変更されず、エラーサービス応答が作成される。

7.3.4.8 SAC オブジェクトインスタンスが INITIALIZED の時には、すべての属性は(関連オブジェクトインスタンス属性の説明に書かれているとおり)該当する初期値にセットされる。初期値はデバイスの不揮発性メモリから検索された値に対応していて、多くの部分でその前の NORMAL OPERATING 状態でセットされた値を反映しているかもしれない。

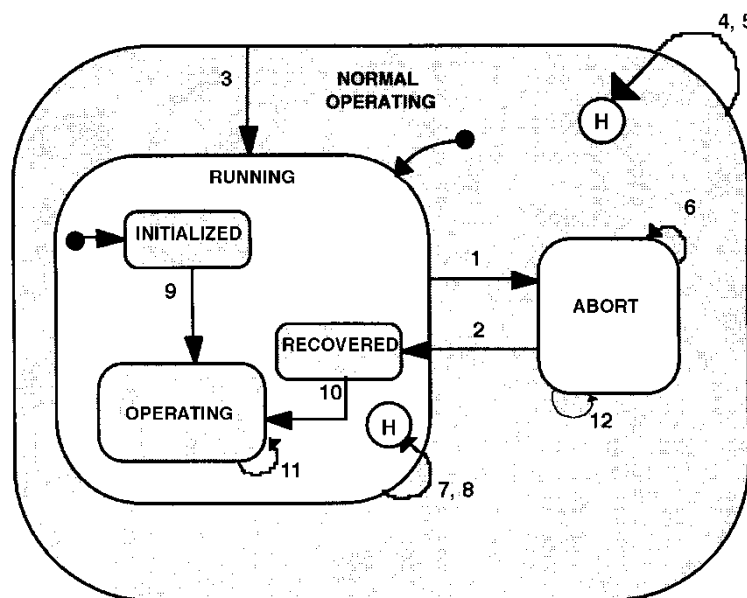
7.3.4.9 さらに、SAC オブジェクトインスタンスは INITIALIZED 状態に入る度にあらゆるセンサ、アクチュエータ、およびコントローラオブジェクトインスタンスに Reset サービス要求を発行しなければならない。そうすると各オブジェクトインスタンスはそれぞれの INITIALIZING アプリケーションプロセスの完了時に合格あるいは失敗の応答で応えなければならない。SAC オブジェクトインスタンスは、デバイスが OPERATING 状態にはいるための制約をクリアしているかどうかを決めるために、その応答を調整しなければならない。制約をクリアしているなら、Execute サービス要求が DM オブジェクトインスタンスへ発行される。制約をクリアしていないなら、DM オブジェクトインスタンスには例外状況が報告される (§ 7.4 参照)。

7.3.4.10 SAC オブジェクトインスタンスが ABORT 状態から recover しているときには、属性はすべて(メーカーが定義した関連オブジェクトインスタンスリカバリングアプリケーションプロセスで決められているとおり)該当する値にセットされる。さらに、SAC オブジェクトインスタンスはあらゆるセンサ、アクチュエータ、およびコントローラオブジェクトインスタンスに Recover サービス要求を発行しなければならない。各オブジェクトインスタンスはそれぞれのリカバリングアプリケーションプロセスの完了時に合格あるいは失敗の応答で応えなければならない。SAC オブジェクトインスタンスは、デバイスが OPERATING 状態にはいるための制約をクリアしているかどうかを決めるために、その応答を調整しなければならない。制約をクリアしているなら、Execute サービス要求が DM オブジェクトインスタンスへ発行される。制約をクリアしていないなら、DM オブジェクトインスタンスには例外状況が報告される。

7.3.4.11 SAC オブジェクトはオプションで時間切れタイマー属性を持つ。この属性の値は"ExpirationTimer"属性フォーマット定義に記されている割合で減っていく。"ExpirationWarningEnable"属性が Enable にセットされているなら、"ExpirationTimer"属性がゼロ以下になった時、警告例外状況になり、(PublishAttribute サービスを使って)DM オブジェクトインスタンスに報告される。"ExpirationWarningEnable"属性が Disable にセットされているときには、"ExpirationTimer"属性は減り続ける。しかし、警告例外状況は DM オブジェクトインスタンスには報告されない。

7.3.4.12 "ExpirationTimer"属性はゼロ以下まで減り続ける。そのため、負の数字は時間切れからさらに経過した時間を表す。この属性が負の最大値に達すると、この値を維持してそれ以上は減らない。

7.3.4.13 "ExpirationTimer"は ABORT 状態にある間はオブジェクトもその値を維持している。RECOVERING 状態の時、デバイスのカレントステータスが評価され、存在するあらゆる例外状況が DM オブジェクトインスタンスに報告される。



注 1: この図には包括的挙動だけが書かれている。完全な状態図には必然的にデバイス独自の挙動も含めなければならない。

注 2: 遷移 4 から 8, 11 および 12 は再帰的である、つまり、新しい状態は現在の状態(ネスト化された下位状態すべてを含む)と同じである。

図 5
SAC オブジェクトインスタンスの挙動*

表 11 SAC オブジェクトインスタンスの挙動の状態説明

状態	説明
NORMAL OPERATING	オブジェクトインスタンスが存在する。オブジェクトサービスの処理ができる
RUNNING	NORMAL OPERATING への入り口となる下位状態である。オブジェクトインスタンスは ABORT 状態にはない
INITIALIZED	NORMAL OPERATING および RUNNING への入り口となる下位状態である。オブジェクトインスタンスが存在し初期化されている。属性はすべて(各デバイスごとのセンサ/アクチュエータネットワーク個別デバイスモデル仕様書に示される)適切な初期値にセットされている
RECOVERED	オブジェクトインスタンスは abort からリカバリした状態にある
OPERATING	RUNNING の中で、INITIALIZED および RECOVERED を除く下位状態全体を表わしている。この下位状態について、およびこの下位状態への遷移についての詳細は本書の範囲外である
ABORT	オブジェクトインスタンスは abort された状態にある;この状態の詳細な説明は本書の範囲外である

表 12 SAC オブジェクトインスタンスの挙動の状態遷移マトリクス^{注1, 注2*}

#	現在の状態	トリガー	新しい状態	アクション	コメント
1	RUNNING	Abort 要求	ABORT	Abort Abort 応答	ABORT 状態はデバイスによって異なる
2	ABORT	Recover 要求	RECOVERED	Recover Recover 応答	RECOVERED 状態はデバイスによって異なる
3	NORMAL OPERATING	Reset 要求	RUNNING	Reset Reset 応答	NORMAL OPERATING の下位状態すべてに有効。 RUNNING/INITIALIZED に移行

#	現在の状態	トリガー	新しい状態	アクション	コメント
4	NORMAL OPERATING	無効サービス要求	NORMAL OPERATING	エラー応答	NORMAL OPERATING の下位状態すべてに有効
5	NORMAL OPERATING	GetAttribute, SetAttribute あるいは RestoreDefault 要求	NORMAL OPERATING	適切な属性を取得あるいはセットする, あるいはデフォルト値を回復させる	NORMAL OPERATING の下位状態すべてに有効
6	ABORT	Abort 要求, 作動要求	ABORT	エラー応答	この状態では無効な要求
7	RUNNING	Recover 要求	RUNNING	エラー応答	Recover サービスは ABORT 状態にある時にしか引き起こされない
8	RUNNING	PublishAttribute	RUNNING	PublishAttribute サービスで属性 ID で特定された属性が発行される	通知サービス
9	INITIALIZED	作動要求, あるいはデバイスによって異なる	OPERATING	デバイスによって異なる	この遷移に関連した挙動は該当するセンサ/アクチュエータネットワーク独自のデバイスモデル仕様でさらに規定される場合がある。
10	RECOVERED	作動要求, あるいはデバイスによって異なる	OPERATING	デバイスによって異なる	この遷移に関連した挙動は該当するセンサ/アクチュエータネットワーク独自のデバイスモデル仕様でさらに規定される場合がある。
11	OPERATING	作動要求	OPERATING	エラー応答	OPERATING 状態では無効な要求
12	ABORT	PublishAttribute (属性発行)	ABORT	なし	ABORT 状態にある時には属性は発行できない

注 1 遷移 4~8, 11 および 12 は再帰的である, つまり, 新しい状態は現在の状態 (ネスト化された下位状態すべてを含む) と同じである。

注 2 この表には包括的な挙動しか書かれていない。

7.3.5 SAC オブジェクトに内蔵されているオブジェクト — SAC オブジェクトに内蔵されているオブジェクトタイプは表 3 にリストアップされている。そのオブジェクトタイプのインスタンスの構造および挙動は本項で詳細に説明されている。

7.3.5.1 アセンブリオブジェクト — アセンブリオブジェクトインスタンスは, デバイスにある一つ以上のオブジェクトインスタンスから一つ以上の属性をグループ分けしてネットワークでの通信のためのシングルデータ構造にするためのメカニズムを提供するのに使用できる。

7.3.5.1.1 Assembly オブジェクト属性 — 要求されている Assembly オブジェクト属性および「共通オプションの」Assembly オブジェクト属性は全部表 13 にリストアップされている。

表 13 アセンブリオブジェクト属性

属性名	属性識別子	アクセスネットワーク	必須	形式
Data	AsmA1	コンテキストによって異なる	Yes	コンテキストによって異なる

7.3.5.1.1.1 Data — 集められたデータを含む属性。このデータの構造はコンテキストによって異なる。

7.3.5.1.2 Assembly オブジェクトサービス — Assembly オブジェクトインスタンスは 2 つのサービスを提供する: Get Attribute および Set Attribute である。

7.3.5.1.2.1 Get Attribute — このサービスはネットワークでオブジェクトインスタンス属性の値を読み取るのに使われる。表 14 にはこのサービスに規定されているパラメータが書かれている。

表 14 Assembly オブジェクト Get Attribute パラメータの定義

パラメータ	要求／指示	応答／確認	形式	説明
属性 ID	M	C	ネットワークによって異なる	値が要求されている属性の属性識別子
属性値	–	M	コンテキストによって異なる	要求されている属性の値

7.3.5.1.2.2 *Set Attribute* – このサービスはネットワークで修正可能であると特定されているオブジェクトインスタンス属性の値を修正するのに使われる。表 15 にはこのサービスに規定されているパラメータが書かれている。

表 15 Assembly オブジェクト Set Attribute パラメータの定義

パラメータ	要求／指示	応答／確認	形式	説明
属性 ID	M	C, =	ネットワークによって異なる	値を修正することになる属性の属性識別子。応答／確認に属性を入れるかどうかはオプション
属性値	M	M	コンテキストによって異なる	属性を修正してこの値にする。応答／確認に属性を入れるかどうかはオプション。値が関連した要求／指示メッセージで与えられたものと違う(つまり否定応答)状況は本文書では定義されていない

7.3.5.1.2.3 デバイス独自および／あるいはインプリメンテーション独自の **Assembly** オブジェクトインスタンスは追加サービスおよび／あるいはサービスパラメータを提供するかもしれない。それは通知あるいは要求タイプのサービスかもしれない(§ 6.4.1 参照)。デバイス独自の追加サービスおよびサービスパラメータの完全な定義は該当するセンサ／アクチュエータネットワーク独自のデバイスモデル仕様に書かれているだろう。

7.3.5.1.3 *Assembly* オブジェクトの挙動 – **Assembly** オブジェクトインスタンスが示す挙動は *GetAttribute* および *SetAttribute* サービスを支持する。つまり、属性"Data"に関する *GetAttribute* サービス要求が受信されると、応答には"Data"属性値が含まれる。同様に、属性"Data"に関して *SetAttribute* サービス要求が受信されると、"Data"の値が本当に修正可能なら修正され、成功サービス応答が返される。そうでなければ不成功サービスが返される。

7.3.5.2 *Local Link* オブジェクト – **Local Link** オブジェクトインスタンスはあるオブジェクトインスタンスの属性を別のオブジェクトインスタンスの属性へ「リンク」するのに使用できる。

7.3.5.2.1 *Local Link* オブジェクトの属性 – 要求されている **Local Link** および「共通オプションの」**Local Link** はすべて表 16 にリストアップされている。

表 16 Local Link オブジェクトの属性

属性名	属性識別子	アクセスネットワーク	必須	形式
Source Object Class (ソースオブジェクトクラス)	LnkA1	コンテキストによって異なる	N	コンテキストによって異なる
Source Object Instance (ソースオブジェクトインスタンス)	LnkA2	コンテキストによって異なる	N	コンテキストによって異なる
Source Object Attribute (ソースオブジェクト属性)	LnkA3	コンテキストによって異なる	Y	コンテキストによって異なる
Destination Object Class (リンク先オブジェクトクラス)	LnkA4	コンテキストによって異なる	N	コンテキストによって異なる
Destination Object Instance (リンク先オブジェクトインスタンス)	LnkA5	コンテキストによって異なる	N	コンテキストによって異なる
Destination Object Attribute (対象オブジェクト属性)	LnkA6	コンテキストによって異なる	Y	コンテキストによって異なる
Commit (コミット)	LnkA7	コンテキストによって異なる	N	ブーリアン