

SEMI E133-0212

SPECIFICATION FOR AUTOMATED PROCESS CONTROL SYSTEMS INTERFACE

自動プロセス制御システムインタフェースの仕様

免責事項: この SEMI スタンドアードは、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 スタンドア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)。

本スタンダアードは、global Information & Control Technical Committee で技術的に承認されている。現版は 2011 年 12 月 24 日、global Audits and Reviews Subcommittee にて発行が承認された。2012 年 2 月に www.semiviews.org および www.semi.org で入手可能となる。初版は 2004 年 3 月発行。前版は 2010 年 11 月発行。

1 目的

1.1 R2R 制御(R2R:実処理間制御)、フォルトディテクション(FD:障害検出)、フォルトクラシフィケーション(FC:障害分類)、フォルトプレディクション(FP:障害予測)、統計プロセス管理手法(SPC)等のプロセス制御システム(PCS)の統合実装を現状、ならびに将来の工場側で効果的に設置するには、プロセス制御システムのインタフェアースを規定することが必要である。これは、(1) 上記の機能間で、(2) 依存関係のある上記以外の工場内のシステム(装置データ収集系を含む)が効果的に相互動作し、データを共有することを可能にするためである。本スタンダアードは、将来開発する製品を応用する予定のある装置サプライヤ、半導体製品製造会社、装置を構成するシステムのサプライヤ、制御システムサプライヤが使うことを意図するものである。

2 範囲

2.1 本スタンダアードは PCS 性能、機能化グループ、機能化グループのインタフェアースを定義することに主眼を置くものである。本スタンダアードは、将来的に投票を経て、新たな機能化グループが追加できるように構成されている。本書の構成要素には次のものを含む:

- 本書のドメインに適切な重要な用語と概念の定義、
- 本規定の表記法の説明、
- PCS 適用範囲内での主要な機能化グループの特定、
- 個々の機能化グループに求められる性能の説明(必須と任意)、
- 上記の機能化グループの定義、次のものを含む:
 - インタフェアース間をまたいで利用できるデータ。これにはインタフェアースデータの説明も含む(入出力共)。本規定の将来の改訂では、継承や集約モデルの形態による詳細データ構造も含まれる。PCS の機能化グループすべてに共通な構造、および個々に特化した構造も同様にモデル化する。
 - サポートするサービスおよびインタフェアースの想定される挙動、
 - 準拠程度を確認するテスト方法の説明。

2.2 特定のフォーマットやプロトコルの実装は、その適用する技術に依存して、本規定のサブドキュメントの「ドット」規定に記述する。想定し得るフォーマットの例として XML がある(データ型の定義やメタモデルのスキーマも含まれる)。

2.3 本スタンダアードは、本スタンダアードに盛り込まれた情報に加え、正当性検証手続きを明確にする。これは実装対象を決定し、現実の実装が可能な限り早期に出るように取り掛かるためである。

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 ここに提起する PCS スタンダードは以下の点には言及しない：

- PCS 機能化グループ間の低階層の通信機構 (例, ハードウェア接続機構, 伝達層のソフトウェアなど),
- 実装置への実装の方法論, あるいは工場側の制御方法,
- PCS 機能化グループの内部構造や実装方法の指定,
- 補完するアプリケーションの性能 (レシピ管理, データ収集, エンジニアリングに関する分析, データの保管, スケジューリング, 改定管理, など),
- 情報の管理と機能化グループ間の協調動作。これは実装に委ねられる,
- 装置内部の通信, あるいは工場側システム内の通信,
- 個々の制御アルゴリズムに決定, モデル, 実行時の選択的な挙動。これは「状況に対する整合」を含める PCS の機能には任意なものとしてふれる,
- プロセス制御システムジョブの生成や管理,
- スタンダードの正当性の検証に関するテスト計画。

4 参照スタンダードおよび文書

4.1 SEMI スタンダード

SEMI E81 — Provisional Specification for CIM Framework Domain Architecture

SEMI E121 — Guide for Style & Usage of XML for Semiconductor Manufacturing Applications

SEMI E125 — Specification for Equipment Self Description (ESD)

SEMI E134 — Specification for Data Collection Management

4.2 IEC スタンダード¹

IEC 61158-5 — Digital Data Communications for Measurement and Control – Fieldbus for Use in Industrial Control Systems – Part 5: Application Layer Service Definition

4.3 ISO スタンダード²

ISO 8601 — Data Elements and Interchange Formats – Information Interchange – Representation of Dates and Times

4.4 その他の文書

The Unified Modeling Language Reference Manual, James Rumbaugh, Ivar Jacobson, Grady Booch, 1999 / 0-201-30988-X / Addison Wesley Professional.

The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, 1999 / 0-201-57168-4 / Addison Wesley Professional.

Object-Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, NJ, 1991 by James Rumbaugh, *et. al.*

Equipment Engineering Capabilities (EEC) Guidelines (Phase 2.0), Version 2.5, SEMATECH, July 2002.

1 International Electrotechnical Commission, 3 rue de Varembe, Case Postale 131, CH-1211 Geneva 20, Switzerland. Telephone: 41.22.919.02.11; Fax: 41.22.919.03.00; <http://www.iec.ch>

2 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>

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 略語と頭字語

- 5.1.1 AE (analysis engine) — 分析エンジン
- 5.1.2 APC (advanced process control) — 先進プロセス制御
- 5.1.3 CM (context matching) — コンテキストマッチング
- 5.1.4 FC (fault classification) — 障害分類
- 5.1.5 FD (fault detection) — 障害検出
- 5.1.6 FDC (fault detection and classification) — 障害検出分類
- 5.1.7 FG (functional group) — 機能化グループ
- 5.1.8 FP (fault prediction (or prognosis)) — 障害予測
- 5.1.9 PCS (process control system) — プロセス制御システム
- 5.1.10 PCS job (process control system job) — プロセス制御システムジョブ
- 5.1.11 R2R (run-to-run) — 実処理間制御
- 5.1.12 SPC (statistical process control) — 統計手法プロセス制御
- 5.1.13 W2W (wafer-to-wafer) — ウェーハ間制御
- 5.1.14 XML (eXtensible Markup Language) — 拡張マークアップ言語

5.2 定義

- 5.2.1 先進プロセス制御(Advanced Process Control (APC))³ — 解析メカニズムおよび／または計算メカニズムを使用する制御戦略を使って最適なマシン設定を提示し、処理中の故障ならびに原因を検出するための製造方法。
- 5.2.2 分析エンジン(Analysis Engine (AE)) — 応答を行うために、データや場合によっては操作指示をもたらすプロセス。本スタンダードにおいては PCS のすべての機能化グループに共通な性質を総括するためにこの用語が使われる。
- 5.2.3 コンテキスト(context) — 製造にかかわる事物(例、ウェーハ、ロット、モジュール、装置、レチクル)および、その製造過程での動作の状態を一意に識別する一連の属性。
- 5.2.4 コンテキストマッチング(Context Matching (CM)) — システム(例、製造プロセス、製品、装置)の状態を表現する 1 組の属性値を 1 組の保存された、あるいは計算された値と比較し整合させるプロセス。通常は、コンテキストマッチングシステムによりユニークなアクションが指定可能なように行われる。
- 5.2.5 障害分類(Fault Classification (FC)) — 障害が検出された時に、その原因を特定する技術。
- 5.2.6 障害検出(Fault Detection (FD)) — プロセスの異常を検知するために装置内部の変動、および／あるいはプロセスデータを監視し分析する技術。障害検出は単変量、多変量のいずれの統計分析手法をも含む。
- 5.2.7 障害検出分類(Fault Detection and Classification (FDC)) — FD と FC の組み合わせ。
- 5.2.8 障害予測(Fault Prediction (or Prognosis) (FP)) — プロセスの異常を検知するためにプロセスデータの変動を監視し分析する技術。

³ 会社によっては APC と PCS は用語として等価なものとして用いられることがある。何れもプロセス制御の機能要素を包含するものとして用いられるものと見受けられる。

5.2.9 機能化グループ(Functional Group (FG)) — 緊密に関連するソフトウェアの性能で、統合実装された形で供給されるよう、関係者が望む単位。

5.2.10 プロセス制御システム(Process Control System (PCS))³ — プロセス制御を実行する性能を備えたシステムで、ここで言うプロセス制御とは 実処理 (R2R) 制御, FD, FC, FP, SPC, あるいは PCS 機能化グループとして本スタンダードが将来に定義するプロセス制御機能を 1 個以上含むものをいう。

5.2.11 プロセス制御システムジョブ(Process Control System Job (PCS job)) — 追跡しうる作業の単位で、関連するデータを持ちうるもの。本スタンダードではプロセス制御ジョブは、PCS の分析エンジンが実行する作業の単位に対応する; ID 番号などのユニークな識別記号はプロセス制御ジョブの追跡によく利用される。

5.2.12 実処理間制御(Run-to-Run (R2R) Control) — レシピのパラメータあるいは制御パラメータの選択を実行処理単位間でプロセスの結果を改善するために変更する技術。実行単位はバッチ, ロット, 枚葉ウェーハが想定される。

5.2.13 統計手法プロセス制御(Statistical Process Control (SPC)) — 統計制御の安定点に到達させ、かつ定期的にプロセスの性能を改善するために、適切な操作を行うための、統計手法によるプロセスや製品の計測結果を分析する技術。

5.2.14 ウェーハ間制御(Wafer-to-Wafer (W2W) control) — 実処理間(R2R)制御の形式で、1つの「実行処理」は単一のウェーハの処理である。このことは、処理が始まる前に、レシピが複数の個々のウェーハに対して決定されるかもしれない「ウェーハレベルでの制御」から区別されるが、必ずしも、ウェーハ間から（例えばウェーハ間で）修正する必要はない。

6 背景

6.1 工場側、ならびに装置内でのプロセス制御システムアプリケーションの発展に伴い、そのアプリケーションのインタフェース用のスタンダードが無いことにより、工場内の広範囲な範囲でのプロセス制御が阻害されていることは、今や明らかである。更に、プロセス制御や装置組み込み計測の性能を装置内に置く場合、工場側のアプリケーションと装置側のアプリケーションとの協調動作が必要になる。

7 表記法

7.1 本章は本書で使用する表記法を規定するものである。

7.2 オブジェクトモデリング

7.2.1 統合モデリング言語(Unified Modeling Language (UML)) — 本スタンダードでは、クラス図と例示用オブジェクト図のすべてで UML の記法を用いるものとする。他の種類の図は本規定では使用しない。

7.2.2 UML クラス図はその意味を明確に定義するため、図自体が本規定の一部である。この図の詳細部分の説明は必ずしも本文で改めて説明するとは限らない。

7.2.3 クラス図 — UML のクラス図を言う。

7.2.4 クラス名 — クラス名は語頭に大文字を用いる。

7.2.5 抽象クラスと具象クラス — クラス毎に抽象か具象の指定がある。抽象クラスは直接実装されるものではない。本スタンダードでは「実装」の意味は通信インタフェースを介して工場側にその存在を見せることである。具象と指定されたクラスはどれも直接、実装を求められると考えなければならない。本スタンダードは装置制御システムの実装を意図するものではない。UML クラス図で抽象クラス名はイタリック体で示される。

7.2.6 クラスの属性定義 — クラスの属性は下記の表に示したような表形式で定義するものとする。UML クラス図で+記号を属性の頭に付けたものは公開する属性であることを示す。本規定で指定される属性はすべて公開のものである。

表 1 属性表のフォーマット

属性名	定義	アクセス	必要性	形式
-----	----	------	-----	----

属性名	定義	アクセス	必要性	形式
		RO, RW, または NA	Y または N	下記を参照のこと

7.2.6.1 アクセス — 属性は、装置へのインタフェース経由で設定可能 (ReadWrite すなわち RW) あるいは設定不能 (ReadOnly すなわち RO) かのどちらかである。属性がメッセージ要素を参照する場合 (例えば PCS 分析エンジンと通信する場合)、アクセスという概念は適用されず、アクセスフィールドには「NA」が用いられる。

7.2.6.2 必要性 — この属性が必須か否か。Y – 必須, または N – 任意

7.2.6.3 形式 — 属性のデータ型を定義する。共通装置モデル仕様で指定されるデータ型は、次の各項で説明する。本仕様のデータ型は、高レベルの定義であり、必要に応じて個々の技術に沿ってマッピングしなければならない。共通装置モデルに対する拡張は、本仕様で定義されるデータ型のリストには制限されない。ここで紹介されるもの以外、SEMI の文書でよく使用される形式の実態については、SEMI *Compilation of Terms* の Form の定義を参照のこと。

7.2.6.4 制約

7.2.6.4.1 UML の制約は点線で示され、制約を説明する表現が添えられるのが一般的である。この方法での制約の表記は、UML 図を乱雑に大きくする傾向がある。したがって、本仕様では、制約は個別のクラスに焦点を絞ったダイアグラムだけに示す。

7.2.7 UML の関連

7.2.7.1 UML のクラス間の関連を表現するために使われる機構は実装に依存する。本書は元々、抽象で考え、その機構を規定したり示唆したりはしない。本規定の実装を示す補助スタダードはいずれも、本書で使われる UML の関連を表現する機構の説明がなければならない。

7.2.8 クラスの関連定義表 — 本表は本規定で定義されるクラス間の関連を拾い上げ、説明するために使われる表の例を示す。

表 2 関連定義表のフォーマット

関連の役割名	役割の定義	補足

7.2.8.1 関連の役割名 — 定義される関連の役割の名称。

7.2.8.2 役割の定義 — 関連の機能か役割を説明する。

7.2.8.3 補足 — 関連を補足する特記事項や注意書き。

7.3 Data Types

7.3.1 本仕様では、次のデータ型を使用して属性やその他の類似の高レベルの定義の形式を記述できる。プロセス制御システムのインタフェース仕様の拡張は、本仕様で定義されるデータ型のリストに限定されない。

- **Any (任意)** — このフォーマットは、この項で記載されている他の任意のフォーマットでよい。フォーマットは実装によって決定される。
- **Binary (バイナリ)** — 任意の値を持ち得るバイトのシーケンス。バイナリ値は、構造が明確ではないので「フォーマットなし (unformatted)」と呼ばれることがある。
- **Boolean (ブーリアン)** — 「真」または「偽」の値をとる。
- **Checksum (チェックサム)** — 特定の計算方法を使用して、特定のデータストリームに対して計算されるチェックサムの値。

- **Enumeration (列挙)** — 可能性がある値のリストが可能なフォーマット。本文書では、これらの値はテキスト文字列として表現されるが、異なる方法で実装できる(例えば、規定された値に対応する整数)。
- **Status (ステータス)** — ステータス型は、送信中にエラーが発生したかどうかに関する情報およびエラーのタイプに関する情報を格納する構造体である。この構造体の形式は、§9に記載されている。
- **Integer (整数)** — 数値を表す。整数は常に整数である(つまり小数を含まない)。形式およびビット数は実装による定義に委ねられる。
- **xxx の list (リスト)** — 特定の型の複数のインスタンスを保持できる項目(ここで、xxx はデータ型)。
- **Real (実数)** — 任意の整数や小数を表現できる数値。形式およびビット数は実装による定義に委ねられる。
- **Text (テキスト)** — テキスト文字列。本文書で指定されていない限り、長さの制限は実装技術に委ねられる。文字列は、UTF-8として実装するように推奨される。
- **Time (時刻)** — 対象となる発生の日付および時刻の表現。この型の項目の構造と形式は実装に委ねられる。

7.4 サービスメッセージの表現

7.4.1 サービスリソースの定義

7.4.1.1 サービスリソースの定義表では下表のように、特定のサービスグループについての特定のメッセージのセットを定義している:

サービス	必要性	タイプ	説明
サービス名	Y/N	N または R	サービスの目的

7.4.1.2 必要性は、Yes (Y) または No (N) となる。タイプは通知 (N) か、あるいは要求 (R) のどちらかである。要求サービスは、データまたは実行すべきオペレーションを要求するサービスユーザーによって起動された要求メッセージとサービスプロバイダーが返信するそれに対する応答メッセージで構成される。

7.4.2 サービスパラメータの定義

7.4.2.1 サービスパラメータ定義表は、パラメータが明確に規定されている各サービスに対して、次の表に示すようにパラメータの詳細を定義する。

パラメータ	要求	応答	形式	説明
サービスパラメータ名	(下記参照)	(下記参照)	データタイプ	パラメータの目的

7.4.2.2 要求と応答は下記の値をとることができる:

M	必須パラメーター有効な値が与えられなければならない。
C	条件パラメーター定義される場合もあるし、定義されない場合もある。値が与えられるかどうかは完全にオプションである、あるいは別のパラメータの値によって異なる。
-	パラメータは使用されていない。
=	応答の値は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、応答の M と C の入力を(=)で修正することができる。

7.5 振る舞いの説明

7.5.1 シーケンスの振る舞い: 必要に応じて、本書では分析エンジンの動的振る舞いの記述にハレル状態図表記法を用いる。この表記法の概要は SEMI E30 の付属書に説明されている。正式な定義は Science of Computer Programming 8「Statecharts: A Visual Formalism for Complex Systems」(D ハレル, 1987)に説明されている。

7.5.2 状態の振る舞い: 必要に応じて、各状態遷移の本質を明確に記述するために状態図と併せて遷移表が規定されている。各遷移表には遷移番号に対して、「前回状態」、「トリガー」、「新状態」、「動作」、「コメント」の欄がある。動作に対するトリガーは「前回状態」の間に起こる。「動作」は、(1) 「前回状態」から出るときに起こる動作、(2) 「新状態」に入るときに起こる動作、(3) 遷移と密接に関連して起こる動作の組み合わせを含む。それらの区別はない。

8 概要

8.1 用語の定義の章で行った機能化グループの平易な定義を使って、初版となる PCS 機能化グループが定義されている。こうしたグループの適切な範囲(およびその境界)を決めるために、想定した性能が有効な PCS の機能化グループとして相応しいか否かを調べる場合、以下の基準が適用される。機能化グループは特に以下の要求事項を最低 1 個は満足していなければならない:

- 全体でタスクが首尾一貫していること。
- 共通のデータの集合を共用し、かつ、それを同様の制約(特性や保存性など)の中で利用しなければならないこと。
- グループ内部のオブジェクトは強く結合していること。
- 過去の実装や統合には制約があること(例、外部のシステムは内部と、全体が統合されているように相互作用しなければならないので性能は分離できないこと)。
- 市販のものと強い関連性が存在すること(すなわち、あるものは依存関係を持つ製品を使って性能を実現するようできている)。

8.2 上記の基準と、当該タスクフォースに加わっている半導体メーカーとサプライヤが表明した実装の優先順位に従って、以下に初版の PCS 機能化グループが特定されている:

- 実処理間制御(R2R(Run-to-Run)Control)
- フォルトディテクション: 障害検出(FD(Fault Detection))
- フォルトクラシフィケーション: 障害分類(FC(Fault Classification))
- フォルトプレディクション: 障害予測(FP(Fault Prediction))
- 統計プロセス管理手法(SPC(Statistical Process Control))

8.3 実処理間(R2R)制御はウェーハ間(W2W)制御を含み、この場合「実行処理」は単一のウェーハの処理である。

8.4 PCS 機能化グループは必要に応じて追加され得るものと承知されていることを注記する。本書の将来的な改訂版では新たなグループを追加する手続きについて説明を施す予定である。

8.5 前述のグループは装置、あるいは工場レベルのいずれにも実装し得る。要求事項として、機能化グループのインタフェースは一貫した方法で定義されるべきものである。図 1 はここに提案された機能化グループと、工場内および装置内の環境下でのインタフェースとの例を表したものである。

8.5.1 プロセス制御システムは多岐にわたる性能、すなわち上記の図 1 に示した工場ならびに装置レベル双方における機能群で構成される。更に、このことは異なるサプライヤが提供する同一の機能の種類の複数の実態が存在することも含まれるものである。図 1 の例では、内部の性能としてわかる違いのみの、幾つかの R2R 制御部が工場レベルで動作している; R2R 制御性能はウェーハ間(W2W)制御を含み、図 1 に示すように装置の内外で提供することができるが、これらの工場レベルでのコントローラはロット間の粒度において R2R 制御を提供することがあり、また一方、装置レベルでの R2R 制御はウェーハ間(W2W)制御を提供することがある。本インタフェース・スタンダードは、装置レベルの機能と工場レベルのシステムの協調動作が、内部の機能の性能を開発したサプライヤに独立して、容易に構築できるようになるものである。

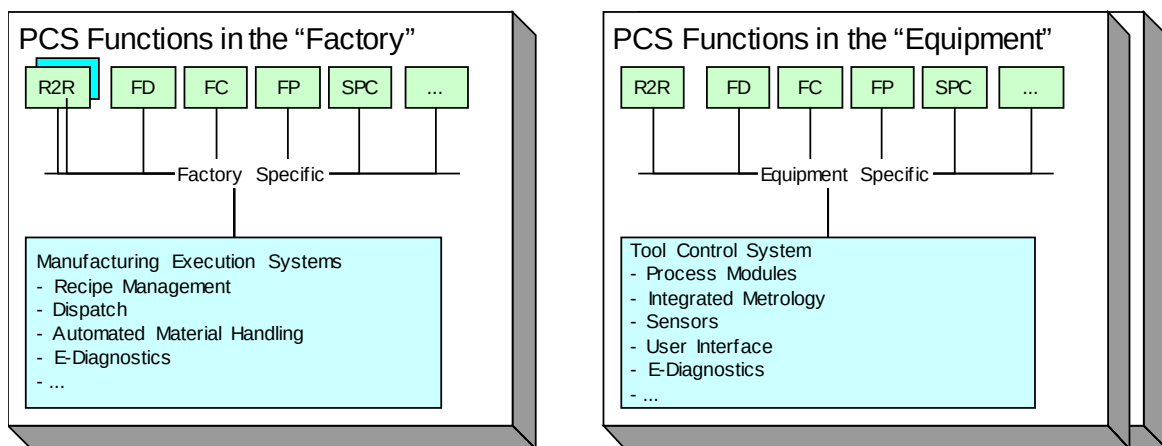


図 1
PCS の機能

9 プロセス制御システム機能化グループの性能とインタフェースのデータ構造

9.1 機能化グループの分解

9.1.1 プロセス制御システムは § 8 に定義した基準を使って機能化グループに分解される。本スタンダードはインタフェースのデータ構造, サービス, 挙動を機能化グループ毎に定義するものである。本章で PCS 機能化グループの性能がより詳細に定義される。まず, 何れの機能化グループにも共通な基本性能が示される。その後, 個々の機能化グループにユニークな基本性能を示す。上位のデータ構造, サービス, 挙動の規定は本章で提供されている。

9.1.2 様々な機能化グループの個々の実装は(同じグループでも)実現される性能, 性能, サービスの質, 内部構造などのレベルにより大きく異なることがあるという点に注意を払うことが重要である。更にそれぞれの実際の工場側で, 同一の機能化グループの, 場合によっては複数のサプライヤから供給された複数の実態が同時並行に動作するかも知れない。機能化グループによっては他の PCS 機能化グループとは全く独立していることも考えられる(言い換えれば, 相関のない独立で, PCS 外部のシステムからの情報にのみに依存する);他には, 他の機能化グループに強く依存し, 自らが実現できるよりも高度の性能を実現するために親密に協調動作を行うものもある。

9.1.3 PCS インタフェーススタンダードの最終目標は, このような実装シナリオ(ここに述べたものより更に多くのものも含む)が, 製品内部の構造を特に明記することなくすべて可能となることである。

9.2 PCS 分析エンジン

9.2.1 § 8 に述べたように, 機能化グループはすべてある PCS 分析エンジンの特殊系である考えることができ, それ故に PCS 分析エンジンの構造や挙動を継承する。本章は PCS 分析エンジンの共通の性能を定義するものである。

9.2.2 一般的な性能表現

9.2.2.1 どの PCS 分析エンジンも図 2 に示されるような入力と出力がなければならない。これは, 本スタンダードで規定した分析エンジンに必要とされる性能を実現するために必要に応じた, ある決まった入力を取り込み, 決まった出力を作り出すように PCS 分析エンジンがなっていないなければならないことを意味する。この入出力は集約のデータ構造をとるものとして規定される。構造は, データ構造に応じて「収集タイプ」に集約される。これら収集タイプの定義は表 4 に示す。PCS 分析エンジンは決まった入力値を利用し, 入力値の内容程度に基づいて操作を行い, 出力を作り出すと言う, 一般的な挙動をするものである。PCS 分析エンジンのこの他の構造や挙動については, 本章のリマインダで規定されている。

9.2.2.2 § 4 に示した本規定の制約に加え, 以下は PCS 分析エンジンの定義範囲を超えるものであることを注記する:

- 操作の性能や出力の作成に伴う、入力値の同期の程度、
- PCS 分析エンジンの内部状態に関する規定。

9.2.2.3 機能化グループが備える PCS 分析エンジンはいずれも表 5 で取り挙げた性能をもっていなければならない。

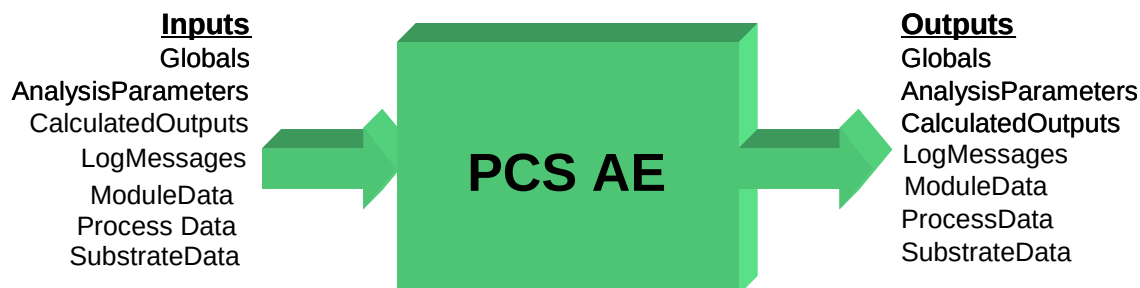


図 2
PCS 分析エンジンに共通な入出力の構造

9.2.2.4 図 2 に示した単純なデータモデルに基づくと、入力には分析エンジンの出力を充てることができることを注記する。分析エンジンの入力として与えられるデータは、前回かそれ以前の回の、分析エンジンからの出力も利用できることが望まれる。

表 3 PCS 分析エンジンの入出力収集タイプの定義

収集型	定義	コメント
Globals	操作全般に渡って広範囲に適用される情報。	例えばロット ID は、他のデータで指定しなくとも良いように、広域変数(global)とみなせる。
AnalysisParameters	分析エンジンが幾多の計算を行うために使う数値データを区別する情報。	方程式の係数、フィードフォワード・データなど。
CalculatedOutputs	分析エンジンからの分析結果を表すデータ。	分析エンジンに与えた様々な入力から導き出された計算値で、ウェーハの状態、プロセスの状態、モデルの状態の情報、推奨される装置の設定値なども含まれる。
LogMessages	分析エンジンがタスクを実行するときの活動を記述するデータ。	分析エンジンの活動を記述するデータ。これは AE との通信プロトコルに関するログ情報を含まないことに注意すること。許容可能な例は、「モデルパラメータ A, B, C によるアルゴリズム #3 (Algorithm #3 with model parameters 'A', 'B', and 'C')」である。
ModuleData	プロセスモジュールの今現在か直近の知り得た物理状態を表す情報で、材料を処理する際に分析エンジンが使うことになるもの。	装置やモジュールの E10 ステート、RF 印加時間、処理したウェーハ枚数など。
ProcessData	プロセスの名称や種類、処理環境にあるセンサの計測値に基づく現在か直近の既知のプロセス状態などの、プロセス自体を示す情報。	プロセス名 (例えば CMP)、プロセスその場計測、トレーサデータ、統計上の集計など。
SubstrateData	分析エンジンが係わる基板についての情報。材料の特質の他、ウェーハ 1 枚、ウェーハ複数枚、1 ロットや複数ロットのデータであっても良い。	表面形状や構成データ、および物理的、電氣的に計測されたその他の特性。

表 4 PCS 分析エンジンの機能化グループの性能

#	説明	必要性	補足
3.1	PCS 分析エンジンの性能の提供	Y	

#	説明	必要性	補足
	PCS 分析エンジンは 1 個以上の R2R 制御, FD, FC, FP, SPC, あるいは本スタンダードで将来 PCS 機能化グループとして定義する機能をもっていなければならない。		
3.2	必要な PCS 分析エンジンのメソッドの提供 PCS 分析エンジンは本スタンダードで示す必要なインタフェースのサービスを提供しなければならない。	Y	これらは PCSAnalyze および PCS UpdateModel を含む。
3.3	指定されるプロセス制御システム環境のモデルの保守 PCS 分析エンジンは指定されたプロセス制御環境の明示または暗示モデルを保守しなければならない。このモデルは分析をするときや分析エンジンの制御性能を使うときに利用されなければならない。	Y	形態や挙動を決めることは本スタンダードの範囲を超えるものである。
3.4	分析エンジンに動作をユニークに指示する機構の提供 PCS 分析エンジンは既定のユニークな動作を分析エンジンに行う指示が出来る入力の仕掛けを用意しなければならない。	Y	仕掛けは単独のキー変数, 例えばモデル名や, 内部でコンテキストマッチングに使うためのコンテキストデータの 1 組などでも良い。動作には特定のモデルのアプリケーションや PCS 結果の生成, モデルの更新, モデルパラメータのやり取りが含まれても良い。
3.5	外部の複数のシステムとのインタフェース PCS 分析エンジンは複数の外部システムから入力を受け入れ, 逆にシステムに向けて出力の生成ができなければならない。	Y	システムに関する定義はアプリケーション環境側の仕事である;これにはプロセス装置, 計測システム, ユーザインタフェースを入れても構わない。
3.6	モデル構成の編集 PCS 分析エンジンはモデル構成の生成/追加, 編集, モデルの削除ができなければならない。	Y	この性能は分析エンジンのインタフェースを介して行われる。
3.7	他の分析エンジンとのインタフェース PCS 分析エンジンは他の分析エンジンの出力を入力として受け取るか, 別の分析エンジンで利用される出力を生成するか何れかできなければならない。	N	特定の機能化グループに属す分析エンジンに要求される場合がある。
3.8	上級モデル管理 PCS 分析エンジンは上級のモデル管理性能を提供する場合がある。それにはモデルの安全なアクセス, プロセス制御環境変化に応じたモデルの自動更新が含まれるが, これだけに限定される訳ではない。	N	

9.2.3 共通インタフェースのデータ構造

9.2.3.1 共通データモデルの概要 — PCS 機能化グループが何れも受け継ぐ共通インタフェースのデータ構造を図 3 に示す。本図はインタフェースによってサポートされる情報の基本的なタイプ(オブジェクト用語の「クラス」)を示す。このモデルは図 2 に示す PCS 分析エンジンの出力および入力構造に適用される。このモデルは, PCS ジョブに関連付けられた完全なメッセージに適用される。つまり, PCS ジョブについて PCS AE に伝達される, あるいはそれから伝達される情報の完全な集合に適用される。PCS AE に入るあるいはそこから出る完全なメッセージは, 9.2.3.5 で定義されるように, アプリケーション対アプリケーションの複数の通信に分けることができることに注意すること。

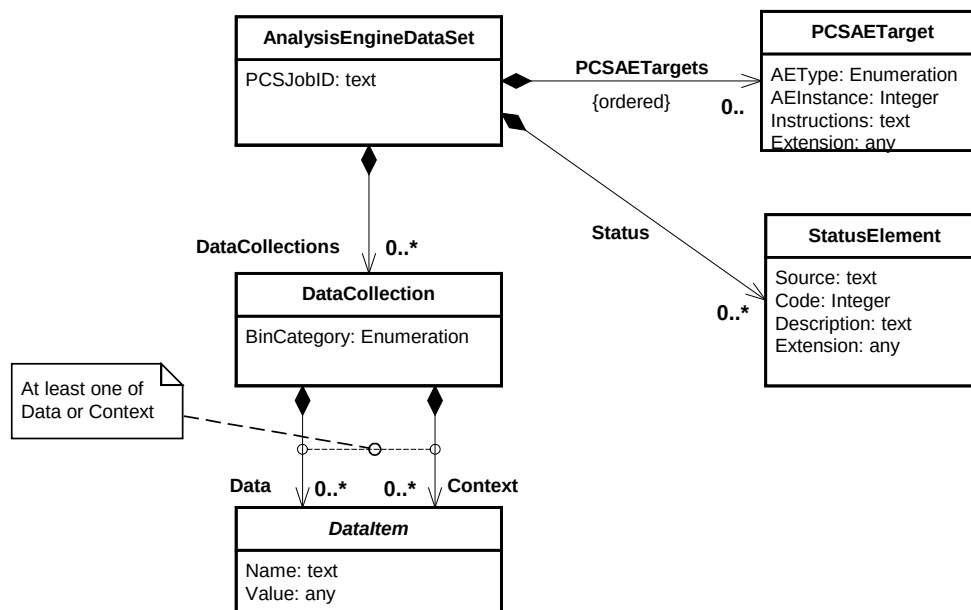


図 3

PCS ジョブに関連付けられた完全なメッセージに対する分析エンジンデータ構造モデル

9.2.3.2 クラス定義 — クラス定義は各クラスの簡単な記述、そのクラスに属する属性を定義する表、必要に応じてそのクラスを十分に定義するテキストから構成される。

9.2.3.2.1 *AnalysisEngineDataSet* — AE の各入出力メッセージに関連するデータを表わす複合的なデータ構造。

表 5 *AnalysisEngineDataSet* の属性

属性名	説明	アクセス	必要性	形式
<i>PCSJobID</i>	プロセス制御システムジョブの一意の識別子。	NA	C	ObjID — SEMI E39 参照。
<i>PCSAETargets</i>	<i>AnalysisEngineDataSet</i> のターゲットとなる AE タイプのリストを示す属性。	NA	Y	<i>PCSAETarget</i> 型の構造化されたデータの順序付きリスト。下記を参照。
<i>Status</i>	PCS ジョブを実行するために、 <i>PCSAETargets</i> で識別される PCS AE の成功または失敗のプロパティに関連付ける拡張可能なデータ構造としてフォーマットされる属性。	NA	C	<i>StatusElement</i> 型の構造化されたデータの順序なしリスト。下記を参照。

9.2.3.2.1.1 *PCSJobID* — *PCSJobID* が *AnalysisEngineDataSet* の必要な属性であることを指定すると共に、それが各 PCS ジョブに一意の *PCSJobID* を関連付けなければならないことを意味する PCS ジョブの一意の識別子であることを指定すること。*PCSJobID* の一意性の管理は、AE あるいは AE の性能を利用したいアプリケーションによって実行できる。AE と通信するアプリケーションは、すべての PCS ジョブの *PCSJobID* を生成し、管理するものと仮定されるが、アプリケーションがこの機能を提供したくない場合は、AE がこの機能を提供する必要がある。*PCSJobID* 属性の存在は、次の事項が条件となる。

- AE と通信するアプリケーションが PCS ジョブの *PCSJobID* を管理している場合、その PCS ジョブに関連付けられる *AnalysisEngineDataSet* 型のすべてのインタフェースデータ構造体に対して、その *PCSJobID* を *PCSJobID* 属性の値として提供しなければならない。
- AE が PCS ジョブに対して *PCSJobID* を生成する機能を提供する場合、生成されたジョブに対する AE から（消費するアプリケーションへ）の最初の通信とそのジョブに関連するその後のすべての通信に *PCSJobID* が提供されなければならない。

- ステータスチェックや処理前の照会など、プロセス制御ジョブを実行する AE が関わらないトランザクションについては、PCSJobID は不要である。

9.2.3.2.1.2 **PCSAETargets** — この属性は AnalysisEngineDataSet がターゲットとする AE タイプのリストを示し、PCSAETarget 型(下記参照)の順序付きリストの構造をとる。多くても 1 つの PCSAETargets の値を各 PCSJobID に関連付けなければならない。したがって、PCS AE によって受信される同じ PCSJobID を有する 2 つの任意のメッセージは、どちらか一方あるいは両方のメッセージによって提供される PCSAETargets について多くても 1 つの値を持つことになる。入力される AnalysisEngineDataSet 内でこのパラメータを受信する各 AE は、その PCSJobID を有するすべての AnalysisEngineDataSet 出力メッセージに対してそのパラメータおよびその値をエコーする。

9.2.3.2.1.2.1 **PCSAETarget** — この属性クラスは、現在の PCS ジョブがターゲットとする AE を表す。複数の AE 間で 1 つのメッセージが通信される場合(しばしば、「デ이지チェーン」と呼ぶ)、PCSAETarget 属性を使用して PCS AE のターゲットを区別する。この場合、PCSAETarget(下記参照)の AEType 属性および AEInstance 属性を利用して AnalysisEngineDataSet データの特定の消費者を識別できる。AE クライアントと単一の AE 間のポイントツーポイントの通信など、多くの場合、PCSAETarget 属性は不要である。

9.2.3.2.1.2.1.1 この属性は次の形式をとる。

PCSAETarget
AEType
AEInstance
Instructions
Extension

図 4
PCSAETarget のデータ構造体

表 6 PCSAETarget の属性の定義

属性名	説明	アクセス	必要性	形式
AEType	命令のターゲットとなる分析エンジンのタイプを表す列挙。	NA	N	可能性のある次の値を有する列挙。(“FDD” – 障害検出, “FCC” – 障害分類, “FPP” – 障害予測, R2R” – 処理間制御, “SPC” – 統計プロセス管理, “PCS” – プロセス制御システム)
AEInstance	ターゲットとなるタイプ内の分析エンジンの特定のインスタンスを示す番号。	NA	N	整数。
Instructions	命令の順序付きリスト。各命令はテキスト型。	NA	Y	テキスト型の要素の順序付きリスト。
Extension	この属性により、サプライヤはターゲットの AE インスタンスに関する追加情報を含めることができる。	NA	N	任意。

9.2.3.2.1.2.1.2 AE が情報を消費する方法の仕様は、本スタンダードの範囲を超える。

9.2.3.2.1.2.1.3 「PCS」の AEType の列挙を使用して、一般の PCS タイプを示す。そうでなければ未指定のタイプである。

9.2.3.2.1.2.1.4 AEInstance の値が指定されていない場合、AEType のインスタンスが未指定であることを示す。

9.2.3.2.1.2.1.5 複数の PCSAETarget インスタンス値が指定されている場合、これらの値が参照するターゲット PCS アプリケーションは、PCSJobID に関連付けられている PCS ジョブを順次、実行しなければならない。この動作を確保するメカニズムは、本スタンダードの範囲を超える。

9.2.3.2.1.2.1.6 *Instructions* は、テキスト型の命令の順序付きリストである。順序付きリスト内の 2 つの要素が同じ値になることはできないことに注意すること。1 つの *Instructions* (命令)は、ターゲットの AEInstance によって実行されるように要求されている特定のタスクに関係づけられる。

9.2.3.2.1.3 *Status* — この条件付き属性クラスは、PCSAETargets 内で特定された PCS ジョブを実行する PCS AE の成功または失敗のプロパティに関連付けられる。

StatusElement
Source
Code
Description
Extension

図 5
StatusElement のデータ構造体

表 7 StatusElement の属性の定義

属性名	説明	アクセス	必要性	形式
Source	属性コードの値が定義される文書を表す。例えば、「Source」は、1 つの SEMI スタンダードを特定することもできる。SEMI スタンダードによって定義されるステータスコードについては、Source の内容はスタンダードの番号を urn で表現したものである。サプライヤによって定義されるステータスコードについては、サプライヤ名を urn で表現したものである。(例えば、SEMI E132 のエラーについては、「Source」の内容は urn:semi-org:E132 であり、サプライヤが定義したエラーについては、企業名が「Acme」とすれば内容は urn:Acme-com である。)コロン記号「:」の後に追加情報を加え、フィールドをピリオド記号「.」で区切ることによって、サプライヤは追加情報を含めることができる。詳細については、SEMI E121 を参照すること。	NA	N	テキスト。
Code	特定の仕様ステータスコードのリストに直接マッピングされる値。このリストのソースについては、本文書の仕様を超える。ステータスコードがゼロの場合、エラーが発生しなかったことを示し、ゼロ以外のステータスはエラーが発生したことを示す。本スタンダード用に定義される特定のエラーコードは、表 8 で提供されている。	NA	Y	整数。
Description	レポートされるコード属性に対する人間が読める説明。Code 属性の定義に直接マッピングされる。	NA	N	構造化データの順序なしリスト。下記参照。
Extension	この属性により、サプライヤはステータスコードに関する追加情報を含めることができる。この属性内に追加情報を記載して任意のエラーコードに関する詳細を提供できる(例、共通コンポーネント、コード 5001、「引数不足、提供される引数“SessionID”がない」)。	NA	N	任意。

表 8 SEMI E133 コード属性値の定義

StatusElement パラメータ			
ソース	コード	説明	拡張
SEMI E133	0	エラーが発生しなかった	<製造者の指定>
SEMI E133	≠ 0	1 つのエラーが発生した	<製造者の指定>
SEMI E133	-1	メッセージシンタックスエラー(例、XML の妥当性エラー、SEMI E133.1 参照)	<製造者の指定>
SEMI E133	-2	不明または予期しない PCSJobID	<製造者の指定>

SEMI E133	-3	不明または予期しないサービス	<製造者の指定>
SEMI E133	-4	不明または予期しない AETarget	<製造者の指定>
SEMI E133	-5	サポートされないサービス	<製造者の指定>
SEMI E133	-6	この時点では利用できないサービス	<製造者の指定>
SEMI E133	-7	不明または予期しないデータアイテム	<製造者の指定>
SEMI E133	-8 ~ -31	<指定されない; 将来の一般的な AE 使用のために予約される>	<製造者の指定>
SEMI E133	-32 ~ -99	<特定の AE タイプ使用のために予約される; 特段の指示がない限り AE 機能化グループ仕様に予約される>	<製造者の指定>
注 1	< -99	製造者の指定	<製造者の指定>
注 1	>0	製造者の指定	<製造者の指定>

注 1 製造者の指定。

9.2.3.2.1.3.1 StatusElement 属性の有無は、AnalysisEngineDataSet 型のデータ構造体を格納するメッセージのプロパティの条件による。次の条件が適用される場合に限り、StatusElement 属性が存在しなければならない。

- メッセージが AE からの出力である。
- 返信メッセージなど、メッセージがその AE のサービスの要求に対する応答である。

9.2.3.2.2 DataCollection — データの集合および 9.2.2.1 で定義される PCS AE 共通入力／出力構造に合致するそのデータの関連コンテキスト。

表 9 DataCollection の属性

属性名	説明	アクセス	必要性	形式
BinCategory	表 4 に記載されている収集タイプの列挙内で定義される可能なタイプのうち、そのタイプのデータ収集の列挙された識別子。	NA	Y	CollectionType — 定義の通り。

9.2.3.2.2.1 CollectionType — 表 4 で定義される BinCategory の列挙。

9.2.3.2.3 DataItem — PCS 環境で意味があるデータ収集の個々のコンポーネント。各データ項目には、PCS AE のクライアントによって決定される「コンテキスト」または「データ」のどちらかの役割がある。

表 10 DataItem の属性

属性名	説明	アクセス	必要性	形式
Name	DataItem の人間が読める名前。	NA	Y	StringValue — 下記の名前付けの規則を参照。
Value	割り当てられたあるいは計算された量。	NA	Y	AbstractValue — SEMI E134 での定義通り。

9.2.3.2.3.1 Name — この属性を使用して DataItem の意味と目的の理解のレベルを提供する。名前は、英数字、スペース、ハイフン、下線で構成しなければならない。名前の先頭文字は英数字でなければならない、先頭あるいは末尾にスペースがあってはならない。

9.2.3.2.3.2 Value — SEMI E134 で定義される AbstractValue は、整数、実数、ブーリアン、文字列、バイナリ、構造体、配列、列挙を含む標準タイプを表現するために使用される抽象ベースクラスである。この方法により、SEMI E134 によって収集されるパラメータ値を、DataItem の値属性により PCS 準拠の AE に直接渡すことができる。AbstractValue タイプの詳細な定義については、SEMI E134 の AbstractValue を参照すること。AbstractValue ベースクラスにより、任意の複雑なデータ項目を AE に渡すことができる。

9.2.3.2.3.3 **CollectionType** が「Globals」である **DataCollection** に属する **DataItem** は、データの役割を果たすことができない(つまり、コンテキストの役割しか果たすことができない)。

9.2.3.2.3.4 役割は PCS AE クライアントによって指定されるので、PCS AE は同じ役割(データまたはコンテキスト)のデータ項目を使用する必要はないが、PCS AE インタフェースを越えて生成される、あるいは消費される I/O データ構造体での役割表現を維持する必要がある。

9.2.3.3 **AnalysisEngineDataSet** 内の要素の順序 — データ構造モデルへの準拠に加え、本スタンダードでは、AE I/O メッセージのデータ内容に対応して、**AnalysisEngineDataSet** の各インスタンス内のクラスと属性の順序を支配する規則も定義する。図 4 に、この順序の全般図を示す。順序を支配する規則は次の通りである。

9.2.3.3.1 PCSJobID の値が 9.2.3.2.1.1 の定義に従って決定されている場合、すべての **AnalysisEngineDataSet** の最初の属性は、プロセス制御システムジョブ ID (PCSJobID) でなければならない。

9.2.3.3.2 すべての **DataCollection** の最初の属性は、**BinCategory** でなければならない。

9.2.3.3.3 **AnalysisEngineDataSet** に **BinCategory** が「Globals」の種類の **DataCollection** が含まれる場合、最初の **DataCollection** は、**BinCategory** が「Globals」の **DataCollection** でなければならない。すべての **AnalysisEngineDataSet** において、多くとも 1 つの **DataCollection** は、**BinCategory** が「Globals」の **DataCollection** でなければならない。

9.2.3.3.4 各 **DataCollection** 内では、役割が「コンテキスト」であるすべての **DataItem** は、役割が「データ」である最初の **DataItem** より前になければならない。

9.2.3.4 **AnalysisEngineDataSet** の追加の要求条件

9.2.3.4.1 9.2.3.2.1 で定義されるように **AnalysisEngineDataSet** の複数のインスタンスが「関連」していて、両方に同じ値の **BinCategory** の同じ **DataCollection** が含まれていて、さらにその **DataCollection** 内に(同じ「名前」の)同じ **DataItem** が含まれている場合、最後の **DataItem** が正しいものと見なさなければならない。基礎をなすプロトコルによって通信される情報の順序の解釈については、本文書の範囲を超える。

9.2.3.4.2 9.2.2.1 で定義されるように **AnalysisEngineDataSet** のインスタンス内あるいは「関連する」インスタンスのグループ内では、1 つの **DataItem** は役割がコンテキストの複数の **DataCollection** 内に存在してよい。その場合には、次の規則が適用される。

9.2.3.4.2.1 **BinCategory** が「Globals」である **DataCollection** 内で定義される任意のコンテキストの **DataItem** は、次の 9.2.3.4.2.2 で定義される場合を除き、関連するすべてのメッセージ内の他のすべての **BinCategory** についてコンテキストの **DataItem** として使用しなければならない。

9.2.3.4.2.2 **BinCategory** が「Globals」以外の **DataCollection** が、現在のメッセージまたは関連メッセージの **Global** の **BinCategory** 内でも定義されているコンテキストの **DataItem** を定義する場合、ローカルの **BinCategory** の **DataItem** の値をコンテキストとして使用しなければならない、つまりローカルのコンテキストの **DataItem** は、この **DataCollection** 内の **Global** の値より優先するということである。

9.2.3.5 複数のアプリケーション対アプリケーションを利用した PCS メッセージとの通信対通信 — PCS メッセージは、多数のアプリケーション対アプリケーション通信あるいはパーティションに分けることができるが、PCS AE がこの切り分け性能をサポートすることは要求されない。これらの通信パーティションのそれぞれは、次の変更を行った図 3 のデータモデルに適合しなければならない。

9.2.3.5.1 PCSAETarget の最小濃度(cardinality)はゼロである。

```

PCSJobID == "[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"
PCSAETargets
APCSAETarget: PCSAETarget
    AEType == "R2R"
    AEInstance == 1
    Instructions
        Instruction == "PCSanalyze"
    Extension == "execute serially"
Status
AStatusElement: StatusElement
    Source == "urn:semi-org:E132"
    Code == 0
• ADataCollection: DataCollection
    BinCategory == "Globals"
    Role == "Context"
        ADataItem: DataItem
            AName: (type == text)
            AValue: (type == any)
        ADataItem
            AName: (type == text)
            AValue: (type == any)
    BinCategory == "ModuleData"
    Role == "Context"
        ADataItem: DataItem
            AName: (type == text)
            AValue: (type == any)
        ...
    Role == "Data"
        ADataItem: DataItem
            AName: (type == text)
            AValue: (type == any)
        ...
    BinCategory == "ProcessData"
    Role == "Context"
        ADataItem: DataItem
            AName: (type == text)
            AValue: (type == any)
        ...
    Role == "Data"
        ADataItem: DataItem
            AName: (type == text)
            AValue: (type == any)
        ...
    ...

```

図 6

AnalysisEngineDataSet のインスタンス内の情報の順序を示す図
 (部分的な例 ; “...”を使用してリストされたフォーマット内であり得る継続項を示す。
 この例の PCSJobID は, SEMI E120.1 の仕様に沿っている。)

9.2.4 共通インタフェースサービス

9.2.4.1 ¶9.2.4.1.1 の規定を除き, PCS AE は, ¶9.2.3 に規定される共通インタフェースデータ構造に準拠するメッセージを受け入れおよび発行することによりサービスを提供しなければならない。PCS AE はここで明記される通りに要求および通知サービスの両方をサポートしなければならない。要求サービスにおいては, サービスに関連する要求および応答の両メッセージは, § 9.2.3 に規定される共通インタフェースデータ構造に適合しなければならない。要求およびオプションの要素の仕様を含む。このサービス実行の成功, 失敗は Code 属性(¶9.2.3.2.1.3 参照)を利用して応答メッセージで伝達される。Code 属性は要求サービスに関連したすべての応答メッセージに要求される。通知サービスにおいては, AE が発行しようと消費しようと, 通知に関連するメッセージインタフェースデータ構造に適合しなければならない。サービスメッセージに提供されるとき, 各サービスパラメータは, 特に規定

がない限り、適切な BinCategory (図 3 および関連テキストを参照) のデータグループの DataItem として提供されなければならない。

9.2.4.1.1 例外: 明確に述べたように、次のサービスは共通インタフェースデータ構造への適合に要求されない。

- PCSPing サービス: Ping サービス応答は、PCSJobID パラメータを持つことを要求されない。

9.2.4.2 PCS 分析エンジンは何れも以下の共通インタフェースサービスを用意しなければならない:

9.2.4.2.1 エンジンからのデータにアクセスを提供する一連のサービスがなければならないこと、

9.2.4.2.2 ログ記録や例外事象の処理を行う一連のサービスがなければならないこと、

9.2.4.2.3 AE の特定のサービスは 9.2.3 に定義されるように、PCSAETarget 属性の命令エレメントを利用して呼び出される。具体的には、サービス名は命令リストのエレメントの値である。要求 (同期) サービスにおいては、サービス名は要求メッセージとそれに対応する応答メッセージに対する命令リストのエレメントの値である。通知サービスにおいては、サービス名は通知メッセージに対する命令リストのエレメントの値である。表 11 に定義するように、このメカニズムを利用した一連のサービスがなければならない。

表 11 一般 AE オブジェクトサービス資源定義^{注1}

サービス	必要性	タイプ	説明
PCSanalyze	Y	R	論理データサンプルを計算、処理することを AE に命令するために使用される。分析の結果は応答メッセージで提供される。このサービスは同期、非同期の両方の PCS ジョブ ^{注2} に対して使用することができる。
PCStartJob	N	R	非同期 PCS ジョブ ^{注2} の開始を AE に命令するために使用される。
PCAddMoreData	N	N	非同期 PCS ジョブ ^{注2} を実行している AE に対してさらに未加工データを提供するために使用される。
PCEndJob	N	R	非同期 PCS ジョブ ^{注2} を完了することを AE に命令するために使用される。
PCJobCompleted	N	N	非同期 PCS ジョブ ^{注2} が完了したことを示すために AE が使用する。AE がこの通知メッセージを発行する。
PCAbortJob	N	N	非同期 PCS ジョブ ^{注2} をアボート (つまり、中断) するために使用される。AE がこの通知メッセージを消費する。
PCJobAborted	N	N	非同期 PCS ジョブ ^{注2} がアボートされたことを通知するために AE が使用する。AE がこの通知メッセージを発行する。
PCUpdateModel	Y	R	分析モデルを新たな情報で更新することを AE に命令するために使用される。このサービスは同期、非同期の両方の PCS ジョブ ^{注2} に対して使用することができる。
PCSEvent	N	N	イベントの検出の表示を提供するために使用する。イベントには特定のデータの受信を含むことができる。このイベントの構成要素の仕様は本標準の範囲外である。
PCSProduceLogData	N	R	AE にログメッセージを生成するよう指示するために使用する。ログデータ要求を使って送られた情報は、その要求に関連するデータの範囲とコンテキストに関係づけることができる。そして、戻された情報は、要求されたログデータを含むことができる。
PCSProduceExceptionData	N	R	AE がもたらした例外の発生に関する情報を提供するよう AE に指示するために使用する。例外情報は、AE によって要求者に戻され、ログデータと同様にデータの範囲を含むことができる。
PCSExceptionProduced	N	N	例外が発生したことを示すため、AE によって使用される。通知によって提供される例外情報は、ログデータと同様にデータの範囲を含むことができる。
PCSPing	Y	R	AE が存在していることおよび通信が健全であることを確認するために使用される。

サービス	必要性	タイプ	説明
PCSCapabilitiesDiscovery	N	R	本スタンダードに定義されているものを含む PCS AE のもつ性能のサポート状況と、性能がそれぞれ現在有効(enable)か、無効(disable)かについて PCS AE に対して問い合わせるために使用される。サービス応答で Capabilities DataItem (¶9.2.4.2.4 で定義されているように) の値はどの性能がサポートされて、そのうちどれが有効(enable)でどれが無効(disable)かを示す。
PCSEnable	N	R	提供可能に指定された性能のサブセットを、PCS AE で提供状態にするために使用される。Capabilities DataItem (¶9.2.4.2.4 の中で定義されているように) の値は、どの能力が有効(enable)あるいは無効(disable)になるように要求されているかを示す。対応する応答で Capabilities DataItem の値は、どの性能が実際に有効(enable)にされたかを示す。

注 1 ¶9.2.4.1.1 で規定される場合を除き、PCS AE は、§9.2.3 で定義される共通インタフェースのデータ構造に準拠するメッセージを受諾し、生成することによってサービスを提供しなければならない。

注 2 §9.2.5 参照。

9.2.4.2.4 サービスパラメータ—これらのサービス用パラメータについて表 12 に説明する。

表 12 PCS サービスパラメータ定義

BinCategory	DataItem 名	説明	役割#1	形式
Globals グローバル	DataSource	本 PCS ジョブに関係するデータのソースを示す。 例えば、EDA ツールおよびデータ収集プランを示す場合もある。	D	DataSource パラメータの形式は本スタンダードの範囲外である。 例えば、それは EDA データソースを識別するテキスト文字列または、特定のツール上の特別のインタフェースを指すためのコンテキストデータかもしれない。
Analysis Parameters 分析パラメータ	Capabilities 性能	次のものを含む複合データ項目: (1) ブーリアン形式のデータエレメント; 要求サービス中の真値はサポートされている性能全てが有効にすることを指示である; 応答サービスメッセージ中の真値は性能がすべて現在有効であるという表示である; 要求サービス中の偽値はサポートされている性能全てが無効にするという指示である; 応答サービスメッセージ中の偽値はすべての能力が現在有効であるとは限らないという表示である。 (2) サポートされている各 AE 機能化グループ性能の情報を提供する(表 13 および表 14 にリストされているような)エレメント個々のリスト。特に、リスト中のエレメントはそれぞれ次の情報を提供する; (2a) 本スタンダードで定義され、サポートされている性能の数、(例えば R2R.A.1) (2b) 性能がサポートされる別の方法。 (2c) 要求メッセージで、性能が有効にするか無効にするかの指示。応答メッセージで性能が現在有効か無効かの表示。	D	説明を参照のこと; 形式の付加仕様はこのスタンダードの範囲外である。 必要なことは単にどの性能がサポートされているかを示すためにこのパラメータが使用されることである。 しかしながら、望ましいことは性能がサポートされていないことを表示するために、サポートの欠如の表示を(例えば、偽値を経由して) 2b で提供できることである。

注 1 ここで、役割カラムでは、「D」が「データ(Data)」を示し、「C」はコンテキスト(Context: 関連情報)を示す。

9.2.4.2.4.1 PCSEnable — 表 12 は、このサービスに対して規定されたパラメータを記述する。

表 13 PCSStartJob サービスパラメータ定義

パラメータ	要求	応答	形式	説明
DataSource	C 注 1	—	§ 9.2.4.2.4 参照。	§ 9.2.4.2.4 参照。

注 1 サービスの要求者がこの(非同期)PCS ジョブ内で利用すべきデータソースまたはソースを示そうとすると、要求メッセージにパラメータが提供される。

9.2.4.2.5 *PCSCapabilitiesDiscovery* — 本サービスのために規定されたパラメータを表 14 に示す。

表 14 PCSCapabilitiesDiscovery サービスパラメータ定義

パラメータ	要求	応答	形式	説明
Capabilities	N	R	§ 9.2.4.2.4 参照。	§ 9.2.4.2.4 参照。

9.2.4.2.6 *PCSEnable* — *PCSEnable* — 本サービスのために規定されたパラメータを表 15 に示す。

表 15 PCSEnable サービスパラメータ定義

パラメータ	要求	応答	形式	説明
Capabilities	R	R	§ 9.2.4.2.4 参照。	§ 9.2.4.2.4 参照。

9.2.5 共通の挙動

9.2.5.1 分析エンジンは何れも入力を受け付け、サービスの組み合わせによって出力を提供できなければならない。そのサービスは表 11: 一般 AE オブジェクトサービス資源定義に定義されているサービスを含む。

9.2.5.2 *PCSJob* 管理の振る舞い

9.2.5.2.1 PCS AE は PCS ジョブを実行することにより、PCS 性能を提供する。本項は、そのような AE における PCS ジョブの管理に関連した AE の振る舞いを定義する。PCS AE は複数の PCS ジョブを同時に管理するかもしれない。AE が維持しなければならない同時実行可能なジョブの最大数は本スタンダードの範囲外であるが、本項は PCS AE が管理している各ジョブの振る舞いを定義する。

9.2.5.2.2 次の PCS ジョブ振る舞いクラスが規定される。

- 同期 PCS ジョブ: AE は、ジョブ出力の要求に対する応答を含む発行された単一の応答メッセージで、新たなジョブに対する PCS ジョブ要求メッセージに応答する。その応答が AE により発行された後、ジョブは終了する。
- 非同期 PCS ジョブ: AE は、要求者とのセッションをオープンすることで新たなジョブに対する PCS ジョブ要求メッセージに応答する。セッション中、特定の PCS ジョブに関連するメッセージにおいて AE は、要求メッセージに同期して応答するかもしれないし(適切な応答メッセージを発行して)、応答メッセージを提供せずに通知メッセージを消費するかもしれないし、非同期に通知メッセージを発行するかもしれない。次の条件のいずれかを満たすなら、非同期 PCS ジョブは AE で終了する。
 - ジョブを終了させる要求は *PCSEndJob* サービスによって AE によって受信され、AE はジョブが終了可能でそうすべきかを決定し、AE が終了要求に対して肯定応答を送信する。
 - AE はジョブの完了を確認しジョブを終了させるべきで、ジョブが終了したという *PCSJobCompleted* サービスによって通知を提供する。
 - AE は *PCSAabortJob* サービスによってジョブをアボートする通知を受け取り、その要求を処理できる。
 - AE は、ジョブをアボートするべきと判断して、*PCSJobAborted* サービスによってジョブをアボートする通知を送信する。

9.2.5.2.3 AE はこれら振る舞いクラスのひとつあるいは両方をサポートしているかもしれない。このサポートに関する追加の仕様は、個々の AE タイプ仕様で提供されているかもしれない。PCS ジョブの実行における AE が適合する振る舞いクラスは、受け取ったサービス要求の特定のタイプ (¶9.2.4 参照) および AE の性能に依存する。

9.2.5.2.4 AE の振る舞いについての追加仕様は、特定の分析エンジンタイプに対する振る舞いに関する仕様に記述される。本スタンダード内のそのクラスに対して定義される振る舞いに矛盾が生じないかぎり、AE は振る舞いについての追加仕様を提供あるいは規定しているかもしれない。

9.2.6 分析エンジンの一般的性能の説明

9.2.6.1 AE モジュールは、図 2 に記述されている共通インタフェース構造を使用して次の入力および出力を許容しなければならない。AE が必要とする機能によっては、すべての入力や出力が必要なわけではない。

9.2.7 入力(Inputs) — AE の入力メッセージには次のデータ型およびデータ値が使用される。

9.2.7.1 一般の AE にはデータ型やデータ値は指定されない。

9.2.8 出力(Outputs) — AE の出力メッセージには次のデータ型およびデータ値が使用される。

9.2.8.1 一般の AE にはデータ型やデータ値は指定されない。

9.3 実処理間(R2R)制御機能化グループ

9.3.1 本項は、R2R 制御機能化グループの共通な性能とインタフェースデータ構造を定義する。これにはウェーハ間(W2W)制御性能が含まれることに留意する。

9.3.2 上記の表 5 の一般性能の各々を受け継ぎながら、R2R 制御機能化グループ用へと特化した部分を下記の表 16 に示す。

9.3.3 表 17 に R2R 制御機能化グループにユニークな性能を別途示す。

9.3.4 R2R 一般性能の説明

9.3.4.1 R2R モジュールは図 5 に説明される共通インタフェース構造を使った以下の入出力を受け入れるのが望ましい。R2R のモデルの必要とする機能によっては、入出力がすべて必要と言う訳ではない。例えば、R2R の機能は基板のデータの形態でフィードバックデータのみ必要であるかも知れない。

9.3.5 入力 — 以下のデータ型およびデータ値が R2R 制御の入力メッセージに使われる：

9.3.5.1 R2R 制御部の PCS 分析エンジンは、PCS 分析エンジンから入力データ構造およびデータ構造要求を受け継ぐ。

9.3.5.2 次のデータ型およびデータ型の値が入力データ構造体に含まれている場合は、それらのデータ型およびデータ型の値は R2R 制御分析エンジンによってサポートされていなければならない。ここで「サポートされる」とは、AE がそのデータ型によって提供される情報を利用でき、そのデータに関連付けられ指示されたアクションを実行できるということを意味する。

9.3.5.2.1 PCSAETarget クラス

9.3.5.2.1.1 AEType 属性の値が”R2R”

9.3.5.2.1.2 Instructions 属性のリスト要素の値が“PCSAalyze”

9.3.5.2.1.3 Instructions 属性のリスト要素の値が“PCSUpdateModel”

9.3.6 出力(Outputs) — R2R 制御出力メッセージでは、次のデータ型およびデータ型の値が使用される。

9.3.6.1 R2R コントローラ分析エンジンは、PCS 分析エンジンから出力データ構造およびデータ構造の要求を継承する。

9.3.6.2 次のデータ型およびデータ型の値が出力データ構造体に含まれている場合は、それらのデータ型およびデータ型の値は R2R 制御分析エンジンによってサポートされていなければならない。ここで「サポートされる」とは、次に示す挙動条件が正当であれば AE がそのデータ型および値を提供できるということを意味する。

9.3.6.2.1 *DataCollection* クラス — 指定された BinCategory 内の DataItem について、次の DataItem 名または名前／値のペアがサポートされていなければならない。

9.3.6.2.2 *ControlAdvice* — この属性は、名前と値のペアの順序なしリスト構造の制御通知を格納する。制御通知が利用されるメソッドは、本スタンダードの範囲外である。

表 16 実処理間（R2R）制御機能化グループの継承を伴う性能

#	説明	必要性	補足
R2R.I.1.	§ 5 に定義された R2R 制御性能の提供 R2R 制御の分析エンジンはプロセス品質データを受け取り、後のプロセスをどう改善するかの手言を作成する際に利用しなければならない；特に R2R の実装では、プロセス性能の改善のため処理単位間のレシピのパラメータの変更や制御パラメータの選択を提供しなければならない。処理単位はバッチ、ロット、1 枚 1 枚のウェーハがありうる（コンテキストで決まる）。	Y	R2R 制御の定義に従う。
R2R.I.2.	必要な PCS 分析エンジンのメソッドの提供 PCS 分析エンジンは本スタンダードで示す必要なインタフェースのサービスを提供しなければならない。	Y	これらは、PCSanalyze および PCSUpdateModel を含む。
R2R.I.3.	指定されるプロセス制御システム環境の R2R モデルの保守 R2R 制御の分析エンジンは指定されたプロセス制御環境の明示、あるいは暗示 R2R モデルを保守しなければならない。このモデルは R2R 制御の性能を提供するために使わなければならない。このモデルはシステムの状態を動的に表現しなければならない。これは構成情報だけでなく指定されたプロセス制御環境に関連して、分析エンジンが受け取った情報に基づいているものである。	Y	形態や挙動を決めることは本スタンダードの範囲を超えるものである；しかしながら、一般的にプロセスの特徴や制御される装置をモデルする。
R2R.I.4.	分析エンジンに動作をユニークに指示する機構の提供 R2R 制御の分析エンジンは既定のユニークな動作を分析エンジンに行う指示が出来る入力の仕掛けを用意しなければならない。	Y	仕掛けは単独のキー変数、例えばモデル名や、内部でコンテキストマッチングに使うためのコンテキストデータの 1 組などでも良い。動作には特定のモデルのアプリケーションや PCS 結果の生成、モデルの更新、モデルパラメータのやり取りが含まれても良い。
R2R.I.5.	外部の複数のシステムとのインタフェース R2R 制御の分析エンジンは複数の外部システムから入力を受け入れ、逆にシステムに向けて出力の生成ができなければならない。	Y	システムに関する定義はアプリケーション環境側の仕事である；これにはプロセス装置、計測システム、ユーザインタフェースを入れても構わない。
R2R.I.6.	基本モデル管理 R2R 制御の分析エンジンはモデル構成の生成／追加、編集、モデルの削除ができなければならない。	Y	この性能は分析エンジンのインタフェースを介して行われる。
R2R.I.7.	他の分析エンジンとのインタフェース R2R 制御の分析エンジンは他の分析エンジンの出力を入力として受け取るか、別の分析エンジンで利用される出力を生成するか何れかできなければならない。	N	R2R 制御の分析エンジンは他の分析エンジンとのインタフェースを必要としないが、モデルのパラメータの更新や制御部の操作の決定に SPC や FD の情報を利用しようとしたインタフェース性能の使例がある。
R2R.I.8.	上級モデル管理 R2R 制御の分析エンジンは上級のモデル管理性能を提供する場合がある。それにはモデルの安全なアクセス、プロセス制御環境変化に応じたモデルの自動更新が含まれるが、これだけに限定される訳ではない。	N	