

SEMI E81-0600(Reapproved 0307)

CIM フレームワークドメインアーキテクチャに関する暫定仕様

免責事項:この SEMI スタンドは、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。

本暫定仕様は、global Information & Control Committee で技術的に承認されている。現版は 2006 年 11 月 21 日、global Audits and Reviews Subcommittee にて発行が承認された。2007 年 2 月に www.semi.org で、そして 2007 年 3 月に CD-ROM で入手可能となる。初版は 1999 年 6 月発行、前版は 2000 年 6 月に発行された。

1 目的

1.1 本文書は半導体工場で使用される際のコンピュータ統合生産(CIM)システムのためのアプリケーションフレームワークについて記した一連の文書について、その構造および内容をまとめたものである。フレームワークとはある分野において情報を共有化しアプリケーションを統合化するための共通環境を提供するソフトウェアインフラストラクチャである。このフレームワークの目的は複雑な製造システムのために業界スタンダードとなるアーキテクチャを確立し、オープン化されたマルチサプライヤによる CIM システム環境へと先導することである。本仕様に記されているフレームワークは CIM フレームワークと呼ばれる。

2 範囲

2.1 本文書の目的は CIM フレームワークのテーマである製造実行システム(Manufacturing Execution System (MES))ドメインを記述し、CIM フレームワークを規定している文書に共通したコンセプトのための参考文書になることである。これには CIM フレームワークを構成する幾つものコンポーネントの構造、関係および相互作用が定義されている。このアーキテクチャでは CIM フレームワークコンポーネントの分割、およびそのコンポーネント一つ一つの仕事が定義されている。ここには統一テーマとして CIM フレームワークで一貫して用いられている製造ジョブ、マテリアル、およびファクトリリソースに関して共通して使われる抽象化も規定されている。

2.2 CIM フレームワークドメインアーキテクチャでは上記のコンポーネントを実装する為に必要なコンピューティングテクノロジーへの依存は扱っていない。これは半導体製造コンセプトという観点から見たコンポーネントの機能性よりむしろソフトウェア製品としてのコンポーネントの実現の方に当てはまるものである。CIM フレームワークアーキテクチャの技術的な側面は別文書 SEMI E96, Guide for CIM Framework Technical Architecture に記されている。

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 CIM フレームワークは競争力を持つ元気な産業界のニーズを満たすために進化し続ける必要がある。このフレームワークの内容は多数の商業用ソフトウェアサプライヤおよびその顧客の現実の開発経験を大量に集めたものである。ここに記されている仕様はそういった会社のプロダクトアーキテクチャだけでなく顧客の要求事項も反映している。

3.2 SEMI 暫定スタンダードである CIM フレームワークドメインアーキテクチャに関する仕様には不十分な点があり、それは完全な SEMI スタンドとしてには対応しなければならないものである。不足している点は以下の通りである:

- このドメインアーキテクチャに基づいて今後作成される関連仕様の詳細との一貫性を確実にすること。
- 大まかなコンポーネント分割から、より小さなコンポーネントの代替性を提供するきめの細かなコンポーネントへの進化。

- 生成時(ビルドタイム)コンフィグレーション機能を持てるようにインタフェースを拡張すること。
- スタンダードな統合モデリング言語(UML)表記法を用いて書かれた完全に検証済みのモデルを提供すること。
- 装置, および装置オートメーションソフトウェアとの相互作用を行うためのインタフェースに関する CIM フレームワークでの表記を, オブジェクトベース装置モデル(OBEM)および自動マテリアルハンドリングシステム(AMHS)といった分野で現れつつあるスタンダードと連携させること。
- 「in」パラメータモードおよびオペレーションリターン値の CIM フレームワークでの使用法を修正して「out」および「inout」モードも持つようにして, オブジェクトの値による受け渡しを行う Microsoft DCOM および IDL 拡張に基づいた実装をより使いやすくすること。
- ドメインアーキテクチャの機能的分割を調整して, もう一つの CIM フレームワーク仕様に関して発行予定の改訂版でのサブコンポーネントの最終的な配置を反映させること。

4 参照スタンダードおよび文書

4.1 SEMI スタンダード

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E10 — Standard for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)

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

SEMI E32 — Material Movement Management (MMM)

SEMI E42 — Recipe Management Standard: Concepts, Behavior, and Message Services

SEMI E58 — Automated Reliability, Availability, and Maintainability Standard (ARAMS): Concepts, Behavior, and Services

SEMI E86 — Provisional Specification for CIM Framework Factory Labor Component

SEMI E96 — Guide for CIM Framework Technical Architecture

SEMI E97 — Provisional Specification for CIM Framework Global Declarations and Abstract Interfaces

SEMI E102 — Provisional Specification for CIM Framework Material Transport and Storage Component

SEMI E133 — Provisional Specification for Automated Process Control Systems Interface

4.2 ANSI スタンダード¹

ANSI/ISA-S88.01-1995, Batch Control Part 1: Models and Terminology

4.3 OMG 文書²

CORBA — Common Object Request Broker Architecture, Version 2.3.1 (OMG Document formal/99-10-07)

MfgDTF — Manufacturing Domain Task Force Roadmap, Version 3.1 (OMG Document mfg/98-06-11)

OMA — Object Management Architecture Guide, Version 3.0 (OMG Document ab/97-05-05)

UML — UML Notation Guide, Version 1.1 (OMG Document ad/97-08-05).

Workflow — Joint Workflow RFP Revised Submission (OMG Document bom/98-06-07)

¹ 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>

(国内連絡先) 日本規格協会 〒107-0052 東京都港区赤坂 4-1-24 03-3583-8003

² Object Management Group, 492 Old Connecticut Path, Framingham, MA 01701, USA; Website: <http://www.omg.org>

4.4 SEMATECH 文書³

CIMArch — Computer Integrated Manufacturing (CIM) Framework Architecture Concepts, Principles, and Guidelines, Version 1.0 (SEMATECH-Technology Transfer #97103379A-ENG).

CIMFW — Computer Integrated Manufacturing (CIM) Application Framework 2.0 (SEMATECH Technology Transfer #93061697J-ENG).

4.5 その他の文書

ALBUS — J.S. Albus and A.M. Meystel, *A reference model architecture for design and implementation of intelligent control in large and complex systems*, International Journal of Intelligent Control and Systems vol. 1, no.1 p.15–30, World Scientific: Singapore, March 1996.⁴

COM+ — <http://www.microsoft.com/msj/1197/complus.htm>; <http://www.microsoft.com/com/>⁵

DCOM — http://www.microsoft.com/windows/down-loads/bin/nts/dcom_architecture.exe.⁵

JAVA — <http://www.javasoft.com>.⁶

WfMC — <http://www.wfmc.org>.⁷

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

AMHS — 自動マテリアルハンドリングシステム (Automated Material Handling System)

APC — アドバンスドプロセスコントロール (Advanced Process Control)

APCFI — アドバンスドプロセスコントロールフレームワーク構想 (Advanced Process Control Framework Initiative)

API — アプリケーションプログラミングインタフェース (Application Programming Interface)

CIM — コンピュータ統合生産 (Computer Integrated Manufacturing)

MES — 製造実行システム (Manufacturing Execution System)

MMMS — 材料搬送管理サービス (Material Movement Management Service (SEMI))

OBEM — オブジェクトベース装置モデル (Object Based Equipment Model)

OMA — オブジェクトマネージメントアーキテクチャ (Object Management Architecture)

PFC — プロセスフローコンテキスト (Process Flow Context)

PFI — プロセスフローイテレータ (Process Flow Iterator)

RFP — 提案要求 (Request for Proposal)

RMS — レシピ管理システム (Recipe Management System)

UI — ユーザインタフェース (User Interface)

WIP — 仕掛品 (Work In Process)

3 SEMATECH, 2706 Montopolis Dr., Austin, TX 78741, USA; Website: <http://www.semitech.org>

4 World Scientific Publishing Co., 1060 Main St., River Edge, NJ 07661, USA; Website: <http://www.worldscientific.com>

5 Microsoft, 10500 NE 8th St., Ste. 1300, Bellevue, WA 98004, USA; Website: <http://www.microsoft.com>

6 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto, CA 94303, USA; Website: <http://www.sun.com>

7 Workflow Management Coalition Office, 2 Crown Walk, Winchester, Hampshire, S022 5XE, United Kingdom

5.2 定義

5.2.1 抽象インタフェース(*abstract interface*) – CIM フレームワークの共通機能を一般化する、コンポーネント外で定義されたインタフェース。抽象インタフェースはインタフェースインヘリタンスメカニズムを介して複数のコンポーネントで使用されることを目的としている。

5.2.2 アプリケーション(*application*) – 1. エンドユーザあるいは別のアプリケーションにドメイン固有の機能を提供する相互運用可能なオブジェクトの集合でできている一つあるいはそれ以上のプログラム。2. 相互作用を行うオブジェクトの集合でできている一つあるいはそれ以上のプログラムが提供する機能。

5.2.3 アプリケーションフレームワーク(*application framework*) – あるドメインでのアプリケーションあるいはアプリケーションのセットを構成するフレームワーク。

5.2.4 アプリケーションインタフェース(*application interface*) – アプリケーションあるいはアプリケーションプログラムが提供するインタフェース。

5.2.5 アプリケーションオブジェクト(*application object*) – アプリケーションインタフェースを実装するオブジェクト。

5.2.6 アーキテクチャ(*architecture*) – あるプログラム/システムのコンポーネント、その相互関係、そのデザインおよび進化の過程において守られる原則およびガイドラインの構造。

5.2.7 属性(*attribute*) – オブジェクトと値の間にある識別可能な関係。属性は値をセットし検索するための関数を持つ場合がある。

5.2.8 挙動(*behavior*) – 要求されたサービスを実行することによって発生する影響およびその結果。

5.2.9 結合(*binding*) – プラットフォームテクノロジーおよびその他の実装固有の基準を独自に選択したもの。

5.2.10 クラス(*class*) – いくつかのオブジェクトが共有している共通構造および共通挙動。クラスは共通した構造および挙動の実装を意味することが多いが、インタフェースはそういった共通機能の仕様を表す。

5.2.11 クライアント(*client*) – 別のオブジェクトに働きかけてあるいは状態を参照して、そのオブジェクトのサービスを利用するオブジェクト。

5.2.12 集合(*collection*) – 自分以外の(複数の)オブジェクトについての記述およびそのオブジェクトを管理するためのサービスを持ち、関連した一つのオブジェクトグループとしてその(複数の)オブジェクトへのアクセスを提供するオブジェクト。

5.2.13 コンポーネント(*component*) – カプセル化されたオブジェクトおよび/あるいは別のコンポーネントを集めた再利用可能なパッケージで、明確に規定されたインタフェースを持つ。コンポーネントは CIM フレームワークでの標準化および代替性の要因の一つである。

5.2.14 コンピュータ統合生産(*Computer Integrated Manufacturing*) – コンピュータの情報処理能力を高めて製造情報を管理し製造オペレーションの実行をサポートあるいは自動化しようというアプローチ。

5.2.15 準拠(*conformance*) – プロダクト、プロセス、あるいはサービスの実装でスタンダードあるいは仕様に適合していること。

5.2.16 準拠要求(*conformance requirement*) – ある実装がその仕様に準拠するために要求する挙動および/あるいは性能の規定を特定すること。

5.2.17 準拠実装(*conforming implementation*) – 規定されている関連準拠要求をすべて満たす実装。

5.2.18 分散システム(*distributed system*) – ローカルなシステムに見えるように配置がユーザに判るようになっていく複数のプロセッシングコンポーネントおよびメモリコンポーネントを統合したもの。

5.2.19 ドメインインタフェース(*domain interface*) – ある分野のアプリケーションに特有なインタフェース。

5.2.20 ドメインオブジェクト(*domain object*) – ドメインインタフェースを実装するオブジェクト。

- 5.2.21 イベント(event) — 重要な出来事の発生を意味する非同期的メッセージ。例えば状態変化あるいは新規オブジェクト生成。
- 5.2.22 イベントチャネル(event channel) — 発行されたイベントをそのイベントに関心を持つオブジェクトへ転送する中間オブジェクト。
- 5.2.23 例外(exception) — オペレーションをコールしたクライアントに、そのオペレーションを実行する際に異常な状況が発生したことを通知するのに使われるインフラストラクチャメカニズム。
- 5.2.24 拡張性(extensibility) — アーキテクチャの完全性およびコンポーネントのスタンダードへの準拠は保持したまま、既存のコンポーネントを拡張あるいは特化し、新しいオブジェクトクラスあるいはコンポーネントを追加できること。
- 5.2.25 フレームワーク(framework) — 特定のドメインでサービスおよび機能を提供するクラスあるいはコンポーネントの集合。
- 5.2.26 実装(implementation) — クラス、オブジェクトあるいはモジュールを内側から見たもので非公開挙動を含む。インタフェースを実現する特定のコードおよび機能。
- 5.2.27 インフラストラクチャ(infrastructure) — 分散オブジェクト相互の協調および分散オブジェクトのライフサイクルをサポートするサービス、設備、および通信メカニズム。
- 5.2.28 インヘリタンス(継承)(inheritance) — 一つ以上のクラス(スーパークラス)で定義された構造あるいは挙動を別のクラス(サブクラス)が共有するというクラス間の関係。サブクラスは普通、既存の構造および挙動を大きくしてあるいは定義し直して自分のスーパークラスを特化する。
- 5.2.29 インスタンス(instance) — 状態、挙動、およびアイデンティティを持つソフトウェアエンティティ。インスタンスおよびオブジェクトという用語は交換可能である。あるインタフェースが規定したオペレーション、シグニチャ、意味をオブジェクトが提供するなら、そのオブジェクトはそのインタフェースのインスタンスである。ある実装がオブジェクトの挙動を提供するなら、そのオブジェクトはその実装のインスタンスである。
- 5.2.30 インタフェース(interface) — クラス、オブジェクト、あるいはモジュールを外部から見たもの。自分の抽象性は強調するが構造および内部挙動は隠す。インタフェース定義に意味が含まれるのが理想である。
- 5.2.31 インタフェースインヘリタンス(interface inheritance) — 別のインタフェースへ修正をどんどん加えて作るインタフェース構造(実装インヘリタンス参照)。CIM フレームワークではインタフェースインヘリタンスは規定しているが実装インヘリタンスは規定していない。
- 5.2.32 相互運用性(interoperability) — 2つのアプリケーションあるいは一つのアプリケーションのいくつかの部分が協力できること。CIM フレームワークでは、アプリケーションコンポーネントはデータを共有できる、互いの挙動(オペレーション)を呼び起こせる、イベントを交換できる、およびサービスの例外を発行できることが要求されている。
- 5.2.33 ジョブ(job) — あるシステムレベルのオペレーションで、ジョブの管理が仕事であるエンティティがその実行を要求する。ジョブのコンセプトは実際のファクトリの「現場」で行われる作業に似ている。現場ではオペレータは上司あるいは管理職が要求した作業(ジョブ)を実行することを求められる。ジョブは長時間に渡ってシステムの複数のリソースを使うことがよくある。CIM フレームワークにおいてジョブの概念は特定のジョブスーパーバイザとジョブがシステムの解決策を実施できるようになっている。
- 5.2.34 ライフサイクル(lifecycle) — 生成、削除、コピーなど、オブジェクトの一生。
- 5.2.35 メソッド(method) — クラスの宣言の一部として定義されたオブジェクトに関して実行されるオペレーション。一般的にはメッセージ、メソッド、およびオペレーションという用語は取り替えても使える。技術的にはメソッドはクラスの中で定義されていて、オペレーションは IDL の中で定義されている。オペレーションはメソッドが実行する。
- 5.2.36 オブジェクト(object) — クライアントが要求できる一つ以上のサービスを実現する、カプセル化された識別可能なエンティティ。クラスのインスタンス。

5.2.37 オブジェクトサービス(*object services*) – 分散オブジェクトに基づいたプログラムで使われることが多い一般的なサービスのためのインタフェース。

5.2.38 オブジェクトマネジメントグループ(*Object Management Group – OMG*) – 異機種分散環境のオブジェクト指向システムに関するオープン仕様の開発を専門に行う国際的な団体。

5.2.39 オペレーション(*operation*) – オペレーションとは一つのエンティティで、オペレーション識別子で識別され、要求可能なサービスを表している。オペレーションには要求パラメータおよび返された結果、および例外の有効値を記したシングニチャが付いている。

5.2.40 永続オブジェクト(*persistent object*) – 自分を生成したプロセスあるいはスレッドが消滅した後も存続できるオブジェクト。永続オブジェクトははっきり消去されるまで存在する。

5.2.41 プロセス定義(*process definition*) – 製造プロセスの特徴を記した情報。プロセスリソースがプロセスにかける時間、プロセスリソース設定、およびプロセスに必要なプロセス性能を含む。

5.2.42 プロセスフロー(*process flow*) – 特定の製品の製造に関するプロセスステップシーケンスを定義する製品仕様の一部。プロセスフローを表すデータ構造は方向性のあるグラフ、具体的にはツリー構造である。ツリーのノードはプロセスフローノードと呼ばれる(下記参照)。プロセスフローのナビゲーションにはサービスが必要である。

5.2.43 プロセスフローコンテキスト(*process flow context*) – コンテキストプロセスフローでの製品の進行に関連したナビゲーション情報。

5.2.44 プロセスステップ(*process step*) – プロセスフロー内で定義可能な最小のプロセッシング活動。一つ以上の連続したプロセスステップはオペレーションセットとして定義できる。

5.2.45 レシピ(*recipe*) – ジョブをどのように実行するかを決定する指示、設定、およびパラメータのセットのうち前もって計画され再利用可能な部分。例えばレシピはプロセスステップを記述するのに使われ、普通は製品仕様の中に入っている。レシピは製造される製品(ウェーハなど)から見たプロセッシング環境を決定する。プロセッシングレシピは製品ランあるいはプロセッシングサイクルの合間に変更されることがある。

5.2.46 サブコンポーネント(*sub-component*) – より大きなコンポーネントの中に完全に含まれるコンポーネント。サブコンポーネントのインタフェースはそれを包んでいるコンポーネントによって外部に曝されている場合と隠されている場合がある。

5.2.47 代替性(*substitutability*) – あるサプライヤからのコンポーネントと別のサプライヤからの機能的に同等なコンポーネントを、システムの別のコンポーネントあるいは入れ替えるコンポーネントのクライアントに影響を与えずに、入れ替えられること。

5.2.48 タイプ(*type*) – オブジェクトの集合に共通したプロパティおよび挙動を記述したもの。タイプは共通インタフェースでオブジェクトを分類したもの。クラスは共通実装でオブジェクトを分類したもの。

6 概要

6.1 本項には読者が本仕様の内容を最大限に理解できるように背景となる情報が記されている。

6.2 対象者

6.2.1 フレームワーク仕様は以下の CIM 技術者のニーズを処理することを目的としている:

- CIM テクニカルマネージャ
- システム制作者およびエンジニア
- アプリケーション開発者およびインテグレータ
- スタンダード開発者

6.2.2 上記のグループは半導体メーカー、ソフトウェア製品サプライヤ、システムインテグレータ、装置サプライヤ、スタンダード機関、大学、国立研究所、およびその他の研究組織など様々な組織に存在すると思われる。

6.2.3 CIM テクニカルマネージャ

6.2.3.1 CIM テクニカルマネージャの仕事は複雑な製造ソフトウェアアプリケーションの開発、配布、および統合を管理することである。彼らは CIM フレームワーク仕様を使って開発活動を計画・組織化し、コンポーネントのテストおよび確認を指導できる。さらにそのソフトウェアを外部のソースから購入する人はシステムアーキテクチャおよびインテグレーションの要求事項をサプライヤと話し合う際に CIM フレームワーク仕様を購入用ガイドとして使用できる。

6.2.4 システム制作者およびエンジニア

6.2.4.1 システム制作者およびエンジニアの仕事はコンピューティングおよび通信インフラストラクチャ、ソフトウェア開発プロセス、プロダクトロードマップ、および関連トピックを含めた全体的なシステムデザインである。また、CIM フレームワークの使い方を拡張して、以下を含めた多くの活動のスタートポイントにすることができる：

- アプリケーションの機能を特定のモジュールに分割および割り当てを行う。
- 分散システムインフラストラクチャとシステムのそれ以外の部分の境界を定義する。
- システムで自分たちがデザインしている部分と外部環境とのオープンインタフェースを規定する。

6.2.4.2 CIM フレームワーク仕様を使ってオープン化された分散システム環境への移行に使う戦略的なシステムロードマップを定義できる。

6.2.5 アプリケーション開発者およびインテグレータ

6.2.5.1 アプリケーション開発者およびインテグレータは半導体製造のためのソフトウェアアプリケーションを製作、インストール、およびサポートしなければならない。CIM フレームワーク仕様は特定のフレームワーク「結合」(つまりターゲットとなるコンピュータシステムハードウェアとソフトウェアテクノロジー)と提携してアプリケーション開発者のための詳細なデザイン要求事項となる。少なくとも CIM フレームワークは製造実行システムの基本的なスタンダードコンポーネントの範囲および境界を定義し、主にインタフェース仕様として使用できる。オブジェクトモデルは新規アプリケーションおよび／あるいは伝統的な統合「ラッパー」の内部デザインに利用して開発プロセスをさらに促進することもできる。最後に仕様は準拠を検証するのに必要な独自のテストを作る際の土台になる。

6.2.6 スタンダード開発者

6.2.6.1 CIM ソフトウェアスタンダードの開発者の仕事は現代の半導体工場で見られる多くのソフトウェアプロダクトが協調するように公開インタフェースおよび共有情報を規定することである。スタンダード開発者は、複数のサプライヤのプロダクトで構成されているはずの工場内の多くの品目を正確に定義するためのオープンな情報ソースとして CIM フレームワークを使うことができる。それには以下が含まれる：

- 複雑な装置を分割し、装置どうしの通信を行うためのスタンダード
- プロダクトおよび原材料の特性および関係
- プロセス定義およびルーティング
- 装置データサンプリング法および保管法
- 従業員の資格
- その他

6.3 CIM フレームワークの基本コンセプト

6.3.1 本項では CIM フレームワークドメインアーキテクチャで使用されているフレームワーク、コンポーネントおよびサブコンポーネントの基本コンセプトについて説明する。

6.3.2 フレームワーク

6.3.2.1 フレームワークとはある分野において情報を共有化しアプリケーションを統合化し開発するための共通環境を提供するソフトウェアインフラストラクチャである。CIM フレームワークとは半導体ウェーハ製造のオブジェクト指向性モデルに基づいた特定のタイプのフレームワークである。CIM フレームワークは、互換性があり代替可能な

アプリケーションコンポーネントで構成されたフレームワークから半導体 CIM アプリケーションを構築可能にする製造オブジェクトおよびオブジェクトインタラクションプロトコルを規定する。

6.3.2.2 フレームワークの心臓部は半導体製造の抽象化概念(ウェーハ、仕様、マシンなど)およびサービス(ウェーハ位置、指定パラメータ設定、マシン利用率取得など)のセットで、それは通常アプリケーション(マテリアル管理、仕様管理、マシンコントロールなど)に具体的に表される。このような抽象化概念の実装はスタンダードなソフトウェアシステムテクノロジー(通信、データベース、およびユーザインタフェースなど)を使う分散コンピュータプラットフォーム(ワークステーション、サーバなど)に実装される。現在の CIM フレームワークは製造情報管理および半導体ウェーハ製造に関する計画とオペレーションの両面のコントロールを具体的なターゲットとしている。

6.3.3 コンポーネントアーキテクチャ

6.3.3.1 CIM フレームワークは MES アプリケーションを一貫したシステムに統合するのに役立つ、MES アプリケーション共通のソフトウェア機能を規定している。CIM フレームワークソフトウェアアーキテクチャはコンポーネントを基にしている。コンポーネントとはソフトウェアを構築するブロック、すなわちソフトウェアアプリケーションを作り上げている機能の「かたまり」である。共通 MES コンポーネントのスタンダードインタフェースおよび挙動を規定することによって、メーカーは複数のサプライヤによるコンポーネントを集めてシステムを作り上げることができる。また、共通コンポーネントを拡張し、改良・拡張された方法で古いコンポーネントと同じインタフェースおよび挙動を実現する改良コンポーネントを古いコンポーネントに替えて使うことによってメーカーが作り上げたシステムを進化させることができる。CIM フレームワークは製造実行システムアーキテクチャを定義しているが、そのコンポーネントは様々な方法で組み合わせられ様々なビジネスプロセスおよびオペレーションポリシーで稼働させられる。

6.3.3.2 図 1 は CIM フレームワークアーキテクチャを階層システムとして描いており、CIM フレームワークはシステムの真ん中の階層にあたる。図 2 には階層が詳細に描かれており、コンポーネント、コンポーネントの相互作用、および拡張が描かれている。後続項には CIM フレームワークアーキテクチャが紹介されている。

Error! Objects cannot be created from editing field codes.

図 1

CIM フレームワーク階層構造

6.3.3.3 インフラストラクチャ

6.3.3.3.1 インフラストラクチャはアプリケーションのための分散コンピューティング環境を提供する。そのサービスにはオペレーティングシステム、通信、データ保存、ユーザインタフェース、イベント配布、例外管理などが含まれる。CIM フレームワークでは、Object Management Group®(OMG)の Object Management Architecture(OMA)あるいは Microsoft® DCOM および COM+アーキテクチャが定義したインフラストラクチャサービスおよび機器を使うことになっており、CIM フレームワークは Java®用のインフラストラクチャのような別のインフラストラクチャにマッピングできている。インフラストラクチャ階層については SEMI E96 に書かれている。

6.3.3.4 共通コンポーネント

6.3.3.4.1 共通コンポーネントとは MES アプリケーションに共通した機能エンティティのことである。例えばマテリアルトラッキング、マシン管理、およびスケジューリングアプリケーションはどれもウェーハグループ(ロット)、マシン、およびプロセスレシピといった共通の共有可能なコンセプトを必要とする。共通コンポーネントはこのようなエンティティに共有モデルを提供し、マテリアルトラッキング、マシン管理、スケジューリングおよびその他のアプリケーションの開発および統合が迅速にできるようにしている。共通コンポーネントはアプリケーションどうしの相互運用性を得るのに必要なコンポーネントのデータおよび挙動を規定する。

6.3.3.5 アプリケーションオブジェクト

6.3.3.5.1 アプリケーションオブジェクトは共通コンポーネントを越えたアプリケーション機能を提供する。こういったアプリケーションオブジェクトはアプリケーション固有のデータおよび挙動(例えば独自のスケジューリングアルゴリズムあるいは独自のレシピ管理関数など)を提供し、アプリケーションが他のアプリケーションと相互に運用できるようにする共通コンポーネントデータおよび挙動を拡大する。アプリケーションオブジェクトはビジネスプロセスワークフロー、ビジネスロジック、およびアプリケーション用のユーザインタフェースの定義も行う。アプリケーションオブジェクトはプロダクト固有あるいはサイト固有の場合が多い機能を提供する。この機能は共通コンポーネントに関する

業界規模のスタンダードの中に入れるべきではなく、共通コンポーネントの拡張性および再利用メカニズムを使って適応させるべきである。

6.3.3.6 コンポーネントの細分化および増加するスタンダード準拠

6.3.3.6.1 CIM フレームワークコンポーネントは機能的なインタフェースおよび挙動に関するスタンダード化で最小の要素である。CIM フレームワークでは SEMATECH CIM Framework Specification Version 2.0 [CIMFW]で規定されているのと同じように、(機能上の範囲から見て)比較的細分化されたコンポーネントを規定している。そのコンポーネントはオブジェクトより大きい(オブジェクトの仕様は通常 3 つから 5 つのオブジェクトを含む一つのオブジェクトモデルについて規定されている)が、従来の MES アプリケーションより細分化されている。しかし、最初の SEMI CIM フレームワークスタンダードはもっと大きなコンポーネントの特定も行っており現在の MES プロダクトの限界にも合わせてある。こういう大きなコンポーネントの場合、図 3 に示されているようにその仕様には細分化されたサブコンポーネントも含まれている(普通は大きなコンポーネント一つにつき 2 つから 4 つのサブコンポーネント)。

6.3.3.6.2 大きなコンポーネントはオブジェクトのメソッドおよび関係を選択してあるものは外から見えるようにする、またあるものは隠す、あるいは抽象化することによって、内部オブジェクト、関係およびコンポーネントの詳細を封じ込めている。大きなコンポーネントはサブコンポーネント、オブジェクトインタフェースおよびオブジェクト挙動の詳細とともに規定されているが、スタンダード準拠には 2 つのレベルがある。最初のレベルの準拠は大きなコンポーネントのインタフェースへの準拠で(カプセル化された詳細を明らかにする必要はない)、2 番目のレベルの準拠はサブコンポーネントの詳細への準拠である。

6.4 CIM フレームワークの機能上の範囲

6.4.1 製造実行システム(MES)という用語はソフトウェア実装の集合を抽象的に表したものである。MES 機能はかなり部分カバーした実装の例はあるが、業界の傾向としてはコア性能の分野にサプライヤが集中する方向に向かっている。多くの場合サプライヤによる MES ドメインのサブセットや単一のソースからの分割可能である部分的プロダクトという提供形態になると思われる。大きくて、よりモノリシックな実装はコンポーネントパッケージングというモデルへ徐々に進化し、より小さな実装になる。理想的には MES の範囲は「MES 内/MES 外」の分割ラインに接する代表的なプロダクトに見られる自然な境界線に対応すべきである。CIM フレームワークが提供するのは MES 範囲内のその性能である。

6.4.2 以下のリストにはコンピュータ統合生産と呼ばれることが多い製造企業システムというより大きなコンテキストで MES の範囲を限定する際に役に立つ基準が記されている。

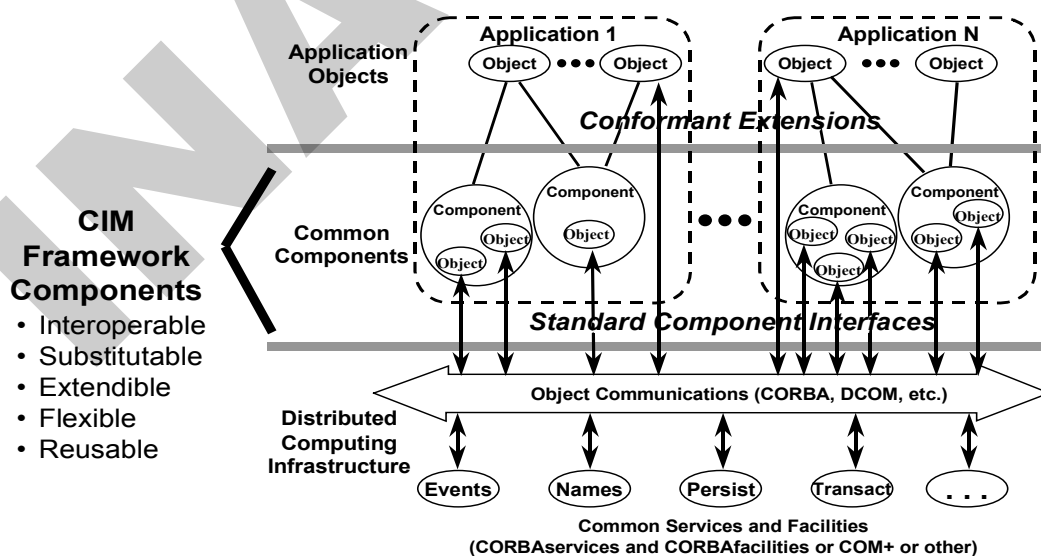


図 2
CIM フレームワークコンポーネントアーキテクチャ

Error! Objects cannot be created from editing field codes.

図 3

コンポーネントおよびサブコンポーネント

6.4.3 コンポーネントは以下にあてはまる場合、「MES 内」でありかつ CIM フレームワークの範囲内である：

- 製造施設でその時進行中の作業を管理するジョブ概念を提供する
- 製造ジョブの実行を通してプロダクトおよびプロセス仕様、マテリアル、および製造リソースの結合を表す
- 製造活動に関して施設レベルでの計画およびスケジューリングを提供する
- 履歴データ、およびプロダクト、製造施設あるいはそのリソースの状態を変えた出来事についての報告へのアクセスを提供する
- ファクトリリソースをコントロールするために調整されたアクションが実行されるようにする
- 製造施設およびそのリソースが抽象的に表現されるようにする
- データ収集および製造プロセスの分析を通して製造パラメータ(設定)が自動的にアップデートされるようにする
- 主要な計測数値(イールド、スループット、サイクルタイム、および利用率など)を捕捉することによって品質管理をサポートする

6.4.4 コンポーネントは以下にあてはまる場合、「MES 外」であり CIM フレームワークの範囲外である：

- 製造装置の内部状態あるいはオペレーションをコントロールする、あるいは操作する
- 製造企業と、顧客あるいはサプライヤといった外部企業との間のビジネス上の取引を扱う
- 製造プロセスの実行ではなくプロダクトデザインを中心にしてプロダクトあるいはプロセス定義を操作する
- ファクトリあるいは製作状態の仮説モデルの生成および操作を中心に行う
- マテリアルを初期状態(原料あるいは半完成品状態)からより価値のあるプロダクトへ変えることに直接関わっていない
- 主に製造プロセスに直接統合されていない研究分析(オフライン計測など)のサポートに使用されている

6.4.5 MES 定義および範囲の基準の適用例は以下のハイレベルな作用から得る事もできる。

6.4.5.1 「MES 内」の範囲

- 製品のデリバリを請け負う一つの施設に対して製品(半完成品あるいは完成品)の制作要求が出される。
- ターゲットとなるプロダクトに関するグルーピング構造と一緒に製作ジョブが生成される。
- 製作ジョブがサブジョブに分割され、再度統合される。
- 製作ジョブは特定の状況のために別個のリソースヘルレーティングし直される。
- 製作ジョブはリソースに振り分けられ、必要なマテリアルが割り当てられ、発送されて実行される。
- 結果的に配送のコミットが変更されるようなプロダクト、リソースあるいはマテリアルの状態変化が報告される。
- 製造プロセスの変更が制作ジョブの実行の変更という結果になる。
- 制作ジョブはマシンリソース性能のために特定の製造マシンに振り分けられる。
- プロセス仕様の値が、それに影響を与える品質コントロールプロセスに関して集められた値のために変更される。
- マテリアルが利用できるようになる、あるいは施設内の新たな物理ロケーションに移動される。

- コンタミネーションコントロール(コンテナのクリーニング)などのために別々の場所にあるコンテナの間でマテリアルが交換される。
- 製造マシンがメンテナンスのために取り外され、ジョブの実行に使用できなくなる。
- 複数のリソースを使うマシンのリソースの一つが外されて使用できなくなるが、マシン全体が使用できないわけではない。

6.4.5.2 「MES 外」の範囲

- プロダクトが定義され製作用に設計される – プロダクトおよびプロセスエンジニアリング。
- 施設があるプロダクトを制作できる状態になる – プロダクション向けリリース。
- 製作用にオーダーがリリースされる – オーダー管理システムからの要求として表される。
- 消費財のストックがなくなる – 設備への配送オーダーがリリースされる – 企業リソーストラッキング。
- 必要なプロダクトを制作できる施設が分析され、要求は供給可能な施設に割り当てられる(将来の制作能力も含む) – 企業プランニングシステム。
- 製造装置はレシピを取得し、マテリアルに製造プロセスを施す – 装置オートメーション。
- 材料搬送装置は指定された搬送先へのマテリアルの搬送をコントロールする – 装置オートメーション。
- ファクトリ状態の想定変更を表すモデリングデータが操作、分析され、変更の影響が決定される – モデリングおよびシミュレーション。
- フロントエンド施設からバックエンド施設へ搬送するためにマテリアルがパッキングされ、キャリアは出荷オーダーを通知される – 搬送物流。
- 要求された配送期日に間に合うように顧客のオーダーは 2 つの施設を使う 2 つのプロダクト要求に分割される – 製作用リリース。
- ファクトリ内の装置ロケーションの配置が新しいツールに合わせて修正される – ファクトリデザイン。

6.4.6 フレームワーク機能の複数レベルでのパッケージングが仕様でサポートされているため、複数のサプライヤからの、交差する性能を持つ様々なアプリケーションを適応させることが可能である。つまり、コンピュータとソフトウェアシステムテクノロジーのセットが特定のフレームワークで結びつけられた場合、アプリケーションはその環境でインスタンス化および実行が可能であり、フレームワークが指定するサービスのうちの中心部分を提供する周知のオブジェクトとしてアプリケーション自身を登録する、ということである。

6.4.7 CIM フレームワーク仕様は統合された製造システムを協働して形成するようデザインされた機能的なコンポーネントを定義している。CIM フレームワークコンポーネントは製造実行システム(MES)の機能をカバーしている。MES とは、装置でマテリアルを処理させてプロダクトを制作するファクトリ規模の機能で、その目的を達成するためにリソース全部を管理する。

6.4.8 図 4 は CIM フレームワークの機能範囲を示している。左側はコンフィグレーションといった製造システムのエンジニアリング面を表し、右側は MES が製造システムのどこに当てはまるかを示している。CIM フレームワークは製造実行オペレーションを大きく、コンフィグレーションを小さく取り上げている。

Error! Objects cannot be created from editing field codes.

図 4

CIM フレームワークの境界：製造実行

6.4.9 CIM フレームワークは半導体ウェーハ製造の機能に関する仕事を規定している。半導体パッケージング、アセンブリおよびテストに関する仕事は含まれていないが、半導体製造のこのフェーズに拡張を適応させるようにデザインされている。

6.4.10 6.3.3 項に記されているコンポーネントの定義を使って、CIM フレームワークのコンポーネントは MES のコンフィグレーションおよび実行に見られる以下の機能を提供する:

- プロダクト管理: プロダクトマテリアルおよびマテリアルグループを管理する。
- 耐久財管理: レチクルおよびマテリアルキャリアといった耐久財を管理する。
- 消耗品管理: ガスおよび化学薬品といった消耗品を管理する。
- 仕様: プロダクト仕様およびプロセスフローを管理する。
- ファクトリオペレーション: ファクトリリソースを効率的に使用してプロダクトを製造し、同時にファクトリの全体的な目標を達成する。
- スケジューリング: ファクトリオペレーションおよびその他の機能をサポートして、プロダクト製造およびファクトリの目標の達成のためにリソースを最大限に利用する。
- 装置トラッキングおよびメンテナンス: 装置状態および利用状況をトラッキングし、装置メンテナンスを管理する。
- プロダクションマシン: プロセス装置および計測装置での製造プロセスステップの実行を管理する。
- レシピ管理: プロセッシングレシピをファクトリ規模で管理、利用する。
- アドバンスドプロセスコントロール: ラン・ツー・ランのプロセスコントロールおよび欠陥検出手法を実行する。
- 材料搬送・保管: マテリアルを移動させ保管する。
- ファクトリ労働力: 労働力および彼らの資格を管理する。
- ファクトリサービス: アクセスセキュリティ、文書管理／変更コントロール、履歴保存および検索といった、コンポーネント共通の機能をサポートする。

6.4.11 § 7 には MES コンフィグレーション(生成時と呼ばれる)および実行(ランタイムと呼ばれる)によるコンポーネント一つ一つの機能関連の仕事がリストアップされている。CIM フレームワークをコンポーネントに分離することによって MES プロダクトの予想される限界を反映させ、複数のサプライヤからのプロダクトを統合され一貫した MES に統合できるようにする。

Error! Objects cannot be created from editing field codes.

図 5

MES コンポーネント統合

6.4.12 図 5 に描かれているように CIM フレームワークは MES インタフェースを規定して MES コンポーネントどうしの統合、つまり MES 内での統合、を可能にしている。CIM フレームワークは MES と外部機能との統合を直接扱ってはいない。企業の機能およびエンジニアリング機能といった外部機能は CIM フレームワークが規定した MES 内インタフェースおよび通信メカニズムと作用を及ぼし合うことができる。MES 機能と製造装置との相互作用では SECS II (SEMI E5), GEM (SEMI E30), ARAMS (SEMI E58), RMS (SEMI E42), および MMMS (SEMI E32) といった既存の装置関連の SEMI スタンドर्डを利用できる。装置はプロダクションマシンおよび装置内の装置トラッキングおよびメンテナンスなどの CIM フレームワークコンポーネントを実装して、CIM フレームワークコンポインタフェースを介して MES (装置ホスト) と統合することもできる。

6.4.13 Object Management Group の製造ドメインタスクフォースは企業、MES、マシンコントロールおよびエンジニアリングアプリケーションに関するスタンダードを採用している (MfgDtf)。OMG 内のその他の関連活動にはワークフローに関する作業 (Workflow) およびコンポーネントアーキテクチャがある。SEMI では OMG と協力してこういった活動の経過を観察している。

6.4.14 CIM フレームワークでは MES の統合に役立つ共通ソフトウェア機能に焦点を合わせている。CIM フレームワークはファクトリリソース、マテリアル、仕様およびファクトリコントロールに関するコンポーネントを規定し、機能的なニーズを満たすようにコンポーネントを組み合わせる会社独自のビジネスプロセス、ワークフロー、およびオペ

レーションルールで稼働させられるように、機能的技術的なアーキテクチャを作り上げる。CIM フレームワークはワークフローおよびビジネスロジックによってコンポーネントの挙動が実行されるようにするがコンポーネントの挙動を定義してはおらず、またユーザインタフェース(意志決定サポートおよび報告を含む)も定義していない。ワークフロー、ビジネスロジックおよびユーザインタフェースはユーザによって異なるので業界の合意であるスタンダードの範囲には入らないと見なされている。CIM フレームワークは MES 機能へのアプリケーションインタフェースを規定するが、その機能の実装は規定しない。

6.4.15 CIM フレームワークの仕様にあるもの

6.4.15.1 本項には一連の CIM フレームワーク文書についての追加情報が記されている。各文書についての要約は以下の通りである。

6.4.15.2 *Specification for CIM Framework Domain Architecture* (本文書)

- CIM フレームワーク仕様についての紹介および概要を記している
- CIM フレームワークコンポーネントの機能上の仕分けおよび機能上の仕事を規定している
- 機能的に CIM フレームワークコンポーネントを統合する共通アーキテクチャパターンを規定している
- CIM フレームワークコンポーネントすべてに関する図表および文項での仕様表記法を記している。
- 関連仕様あるいは CIM フレームワークで触れている文書を列挙している

6.4.15.3 *SEMI E96, Guide for CIM Framework Technical Architecture*

- CIM フレームワークコンポーネント相互運用性アーキテクチャを、OMG CORBA[®]および Microsoft DCOM を含む分散コンピューティングスタンダードに基づいて記している。

6.4.15.4 さらに CIM フレームワーク仕様を詳細に説明している文書がある。文書の構成はドメインアーキテクチャの機能上の仕分けを反映している。構成し直していない限り、詳細は実質的には CIM フレームワークブルーバロットに要求拡張をいくつか追加したものと同じである。こういった文書の構成は SEMATECH の CIM Framework Specification [CIMFW]の影響を受けている。このような姉妹文書は実際の仕様を表しており、本文書はまずその紹介文書としての役割を果たしている。

7 要求条件

7.1 本項は CIM フレームワークコンポーネントの機能上の要求事項を規定している。本項は CIM フレームワークコンポーネントを機能で仕分けし、リソース追跡および保守用の機能アーキテクチャおよびファクトリコントロール用の機能アーキテクチャを定義する。CIM フレームワークコンポーネントを規定するのに必要な図表および文項での表記法も記述する。以下のリストは本項の概要である。

- 全体・抽象定義グループ
- ファクトリサービスグループ
- 製品管理コンポーネント
- 耐久資材管理コンポーネント
- 消耗品管理コンポーネント
- 仕様コンポーネント
- ファクトリオペレーションコンポーネント
- スケジューリングコンポーネント
- 装置追跡・保守コンポーネント
- プロダクションマシンコンポーネント
- レシピ管理コンポーネント

- アドバンスドプロセスコントロールコンポーネント
- 材料搬送・保管コンポーネント
- ファクトリ労働力コンポーネント

7.2 機能区分

7.2.1 本項は CIM フレームワークドメインコンポーネントの機能上の責務を要約する。責務の詳細は個々の CIM フレームワークドメインコンポーネント仕様に規定される。

7.2.2 ドメインコンポーネントの役割は 2 つの種類の機能を担う：

- コンポーネント生成時、即ち、図 4 における製造システムエンジニアリングのコンフィグレーションに係わる機能。具体的なマシンとその状態モデル、あるいはウェーハキャリアとその最初の位置など、コンポーネント内の具体的な実体を規定し、(システム上に)配置する機能。
- 実動時、即ち、図 4 における生産作業の実行に係わる機能。コンポーネントの動作の実行、コンポーネント相互の関係の更新、コンポーネント状態と履歴の更新、およびコンポーネントの状態と挙動に対する動作時のアクセス能力を提供する機能。

7.2.3 初期の CIM フレームワークドメインスタンダードはコンポーネントの生成時の機能を扱っていない。生成時の機能はスタンダードが暫定スタンダードからフルスタンダードの移行に伴い追加されることになっている。しかし本項、機能区分ではコンポーネント生成時の責務を取って特定し、CIM フレームワークの発達に合わせて、生成時の機能を正しいコンポーネントに適切に割り振れる様にする。

7.2.4 コンポーネントは CIM フレームワーク機能を分割するものである。コンポーネントは重なり合いのない様に設計されている。ある機能が存在するのは一つのコンポーネントにだけである。唯一の例外は、他の大部分が全部のコンポーネントでも使用し、特化し、あるいは実装する様な、システム全体にまたがる機能がある程度、存在することである。こういったコンポーネントを越えるドメイン機能は全体・抽象定義物グループ、およびファクトリサービスグループという別々のグループに属している。

7.2.5 完全な MES 実現に必要な責務を羅列すると CIM フレームワークコンポーネントの責務よりずっと多くなってしまう。CIM フレームワークはアプリケーションを統合するのに役立つ機能に焦点を合わせる。サプライヤおよびユーザは機能面の責務をスタンダードでの責務に加えて提供することになる。CIM フレームワークコンポーネントの責務は、CIM フレームワークの範囲をウェーハ加工の MES の実動時の必要事項に絞ることによっても限定されるが、その責務は半導体製造プロセス全部に関しての CIM の実現範囲全部はカバーしない。

7.2.6 § 7.2.7 ~ § 7.2.20 までは CIM フレームワークのコンポーネントとグループの各々の役割を記している。意味を伝えるため半導体製造で特有の例が書かれているが、他の業界でも使えるようなアプリケーションへの特化を妨げるものではない。

7.2.7 全体・抽象定義物グループの責務

7.2.7.1 全体・抽象定義物グループ (SEMI E97 に準拠する) は CIM フレームワークで使用されている抽象データ型および抽象インタフェースを規定している。この仕様は個々のグループに分けられ、ここで規定され、それ以後、更に必要に応じて当然の様に包含または継承される。

7.2.7.2 全体的定義の役割

- 共通データ構造の型定義を提供して表記が一貫するようにする。この項目には座標、優先度、タイムスタンプ、および基本データ型の並びなどの共通概念用のデータ型が含まれる。
- 履歴を考慮した事象の発生の表現ができる型の定義を提供する。
- CIM フレームワークで一貫して使われる共通例外事象の定義を提供する。

7.2.7.3 抽象インタフェースの役割

- マテリアルアーキテクチャ:ファクトリ内にあるマテリアルの識別、グループ化、移動、配置および追跡に共通した機能。このアーキテクチャは製品材料、耐久資材、および消耗品に特化される。§ 7.3.2 参照。