

SEMI T12-1105

治具と工具のトラッキングに関する仕様

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)。

本仕様は、Global Traceability Committee で技術的に承認されたもので、Japanese Traceability Committee が直接責任を負うものである。現版は 2004 年 11 月 24 日 Japanese Regional Standards Committee にて承認されている。2005 年 1 月にまず www.semi.org で入手可能になり、2005 年 3 月発行に至る。前版は 2004 年 7 月発行。

注意: SEMI T12 は、SEMI T12.2 の改訂を反映して、改訂年月が更新された。

1 目的

1.1 半導体製造装置は半導体デバイスを生産する際治具や工具を使うことがある。ある場合には、製品の品質や装置の性能などの重要な状態が治具や工具に依存することがある。この差異はプロセスの回数やプロセスの対象とする製品、保管する環境などの履歴が多く関係する。これら状態と事象の相関関係を明確にするため、治具や工具のトレースを行うことが必要となる。

1.2 本文書の目的は材料を装置上で加工する際に使用される治具や工具のトレースの概念を確立すること、ならびに、この概念を実現するために必要な情報、通信、サービスを明確にすることである。

2 範囲

2.1 本仕様の範囲には必要とされる情報を装置で管理することも含まれる。通常は同一サプライヤの同一型式の、治具の個体が別の装置で使われる可能性もあるので、装置とホストコンピュータの間で交わされる情報も本仕様の範囲内とする。

2.2 マーク位置やマークリーダなどの物理的な案件は範囲外とする、というのは、本仕様はトレースする情報、その情報交換、および理想とする管理事項であるからである。物理的な仕様は別個のスタンダードとなり得るものである。

2.3 ある装置で本仕様する管理や上記の要求事項が求められる限り、プロセス種類や装置の種類の如何にかかわらずその装置は範囲内とする。

2.4 しかしながら、ある種の装置の治具が既存の仕様に準拠し、あるいは慣例的な仕様に沿ってトレースされるか管理されている場合は範囲に含めない。ただし、ユーザ若しくはサプライヤが望む場合は、その様な治具に本仕様を適用できるものとする。

2.5 本仕様はトレーサビリティを実現する基本的な機能を説明するだけのものであるため、模造品の認識や管理運用を含む本機能の応用方法は範囲外とする。

注意: 本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利用者は、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 本仕様はコンセプトと情報通信の基本事項を定義し確認するものである。リーダー／ライターを含め、治工具への物理的な ID マーキング、ならびにセキュリティや運用方法など本トラッキングのアプリケーションはスタンダードの利用者に一任するものとする。これらの事項は利用者の運用や関連する法規にかかわるものである。

4 参照スタンダード

4.1 SEMI スタンダード

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

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

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

注: SEMI による書面での了承なき場合は、本文書の全部もしくは一部の転載を禁じる。

注: 本仕様は SEMI による書面での了承なき場合は、本文書の全部もしくは一部の転載を禁じる。

©SEMI Japan 〒102-0074 東京都千代田区九段南 4-7-15

5 用語

5.1 定義

5.1.1 アタッチメント (Attachment) — 治具および工具の汎用的な呼び方。

5.1.2 工具 (Implement) — 検査または製造装置の組み立て、保守、改良の際に使用するある種のアタッチメント。

5.1.3 治具 (Jig) — 装置または複数の装置で材料の検査、あるいは加工を支援する用途のある種のアタッチメント。

6 表記法

6.1 本項は本文書で使う表記法を定義する。

6.2 オブジェクトに関する表記法 — 本文書は SEMI E39 で定着したオブジェクトの表記法に従うものとする。それにはオブジェクト図類、オブジェクトの用語、標準化オブジェクトの要件も含まれる。それに従い、表記法は Unified Modeling Language (UML) を基にするものとする。

6.2.1 オブジェクトの正式名 — オブジェクトの正式名を引用する場合は語頭を大文字とする。同様な語頭の大文字表現はエンティティ(実体や概念)を論じる際も通常、用いられる。一般物(市街: cities の様な)に関して記述する際は小文字が用いられるが、特定のもの(ニューヨーク市: New York City)を注目する場合は、語頭は大文字とする。

6.2.2 複合属性の要素 — 表中で定義するオブジェクトの属性名は左詰めとする。複合属性の個々の要素は右詰めとし、出現順にその複合属性の下欄に記すものとする。

6.3 状態モデルの表記法

6.3.1 本文書はオブジェクトに規定される動的な挙動を表現するためにハーレル状態図の記法を用いるものとする。本表記法の概要は SEMI E30 の付属書に記される。本記法の公式な規定は “State charts: A Visual Formalism for Complex Systems”¹ に記される。

6.3.2 ハーレルの記法には状態図に一時的なものに伴う「生成」や「消滅」の概念はない。本文書に登場する Secondment[※]はそのようなもので、個々に新たなオブジェクトが生成される際に同一状態モデルのコピーが使われる。本文書では黒丸を円で囲んだ記号をエンティティの消滅の表現に使用する。黒丸は状態図への入り口を表す(エンティティの生成)。

¹ D. Harel, “State charts: A Visual Formalism for Complex Systems”, *Science of Computer Programming* 8, 1987.

[※] この部分は綴り誤りで、原文では「Secondent」となっていますが、正しくは「Secondment」です。

6.3.3 状態図の補完に遷移表を用い、各々の状態遷移の特質を示す。遷移表には遷移番号、元の状態、トリガ、遷移後の状態、動作、備考記事の欄を設ける。「元の状態」の間に遷移を導く「トリガ(3列目の欄)」が発生する。ここで行われる「アクション(5列目の欄)」は以下の組み合わせである:

1. 元の状態を飛び出す際に行われる動作
2. 遷移後の状態に飛び込む際に行われる動作
3. 遷移に強く関連して行われる動作
4. ただし、上記は区別されない。

番号	元の状態	トリガ	遷移後の状態	動作	備考記事

6.4 サービスメッセージの表記 — サービスは装置やホストが提供することになる関数やメソッドである。サービスメッセージは、常に返答を必要とする要求メッセージか、返答が特に必要ではない通知メッセージのいずれかである。

6.4.1 サービスの定義

6.4.1.1 サービス定義表は想定するサービスリソースに対応するメッセージ揃えを、以下の表の様に仕様するものである:

サービスメッセージ名	形態	説明

6.4.1.2 形態は“N”=通知 か“R”=要求である。

6.4.1.3 通知の形態をとるメッセージはサービスの提供側(例えば装置)から発せられ、提供側は利用側の返答を貰えるとは望まないものである。要求メッセージはサービスの利用者(例えばホスト)から発せられる。要求メッセージは提供側にデータ、あるいは動作を求めるものである。要求メッセージは決められた返答メッセージ(メッセージの内容は予め分かっている)を待つものである。

6.4.2 サービスのパラメータの定義集

6.4.2.1 サービスのパラメータ定義表はサービスで使うパラメータの説明、形式、取り得る値を下記の表の様に定義するものである:

パラメータ名	説明	形式: 取り得る値

6.4.2.2 サービスのパラメータ毎に表の1行分を充てる。

6.4.3 サービスメッセージの定義

6.4.3.1 サービスメッセージ定義表は下記の表の様に、サービスのパラメータを仕様するものである:

パラメータ	Req/Ind	Res/Cnf	記事

6.4.3.2 REQ/IND や RSP/CNF 記された欄はメッセージの流される方向に対応する。開始側から発せられたメッセージを「要求(Request)」と言う。受信側ではこのメッセージを「指示(Indication)」と呼ぶが要求である。受信側は「応答(Response)」を送ることがあるが、これは最初の送信側で言えば「確認(Confirmation)」である。

6.4.3.3 下記の記号は REQ/IND と RSP/CNF 欄に現れるものであり、パラメータの定義に使われる(例えば、各々のパラメータは各方向でどう使われるか):

M	必須なパラメータ — 有効な値を用いなければならない。
C	条件依存パラメータ — ある状況下では存在するが、そうでない場合は存在しない。値があり得るかは任意であったり、別のパラメータに依存することもある。
U	ユーザ定義のパラメータ。
-	パラメータは使われない。
=	(応答のみ) 応答でのこのパラメータの値は元々の発信側のパラメータ(がある場合)と同等でなければならない。

7 一般要求事項

7.1 識別手段 — 装置は治工具を物理的に識別する何らかの手段を用意しなければならない。

7.2 通信手段 — 個々の装置およびそれを管理するシステムはその間で何らかの通信手段を用意しなければならない。

8 概要

8.1 半導体製造におけるアタッチメント — 半導体製造ラインは多くの種類の多数の装置で構成される。ある種の装置では半導体素子の生産に治工具の様なアタッチメントを必要とする。こういったものの中には非常に重要な役割を担っているため、品質や歩留まり、あるいはプロセスの生産性がそのアタッチメントに依存することがある。

8.2 アタッチメントの管理 — 製品のより良い品質、歩留まり、生産性を保つには、ある種のアタッチメントは管理が必要である。本文書は、装置上や工場内でのアタッチメントの管理の概念、振る舞い、サービスを仕様する。しかしながら、必ずしもすべてのアタッチメントにその様な管理が必用な訳でもなく、特に重要プロセスに使われたり、精細で装置との組み合わせによっては違いが出てしまうものなどのアタッチメントに限定して本文書は適用されるべきである。

8.3 様々なアタッチメントに有効な概念であること — 幾多の種類の装置があるのと同じくアタッチメントにも多くの種類がある。そのほとんど総てに適用できるようにするため、本文書の規定は物理的制約や運用の詳細に立ち入らない汎用で理想的なものとなっている。このことは急激に変化する技術や生産環境に左右されない上でも重要である。

8.4 論理的かつ情報に主眼を置いた規定 — 概念を規定するドキュメントから逸脱せず、共通に使える部分を可能な限り最大限にするため、関連情報の章以外は論理的な事項および情報に関する事項のみを規定することとする。物理的な規定は別文書で規定されるものとする: これらの規定は別のスタンダード、会社固有の仕様書、或いはその組み合わせが考えられる。

8.5 トラッキングに即した管理 — 工場の戦略に関わる非常に特殊な管理や運営方法と言うものがあって、それらは多くの工場間では共用できないため、特定の工場あるいは特定の製品ののための戦略的管理をスタンダード化するのとは望ましくない。本文書の多くの箇所に出てくる管理事項はトレーサビリティに係わるものである。言い換えれば「治工具のトラッキング」は「アタッチメントの管理」の部分がスタンダード化されたものである。

8.5.1 治工具のトラッキング — 半導体工場では任意の治工具がトラックされることが必要である。

8.5.1.1 治工具をトラックする情報には2種類ある。その1つは伝達される情報である: 例えば今どこにあって誰(どの装置)が使っているのか。論理情報と現物のアタッチメントを分けて考えることが求められる。もう1種類の情報は履歴情報である: 例えば、アタッチメントはどう使われたのか、プロセスの条件はどうだったか、ならびに何時リペアされたか。

8.5.1.2 アタッチメントは治具または工具として、保管場所から装置へ運ばれる、あるいは逆方向に、さらに装置間を移動することもあり得る。移動先に向けて移動元から取り出される時に、この事実がトラッキングシステムに報告されるべきである。移動先に到着した時もこの事実はそのシステムに報告されるべきである。照合はそこで使われ

る前になされるべきである：例えば、プロセス、保守作業、あるいは保管。移動中に異常事態が発生した場合は例外事象が警告される。

8.5.1.3 アタッチメントが装置で使用される時には(例えばプロセス用や保守に)履歴情報は蓄積され、その情報は適当な区切り毎にシステムに報告される。履歴情報の例としては使用期間やサイクルタイム、温度や圧力などのプロセスパラメータ、材料の特性、およびプロセス中に発生し気付いている損傷がある。

8.5.1.4 トラッキングシステムは報告された情報を集計する。システムのクライアントはその情報の問い合わせを行う。クライアントはスケジューラ/ディスパッチャの場合もあり、その場合、材料・製品によっては品質を上げるために製造装置を換えるかも知れない。

8.5.1.5 クライアントは意図する装置が、その装置上の治具が使用限度に近づいたことを表すイベントが発生した際に対処するよう求めることもある。求められる対処方法の例としては処理を止めることがあげられる。もし治具の繰り返し使用回数が許容限度に近い場合にはロットの処理を始める前に一旦停止する様、そのクライアントは求めるかも知れない。

8.5.2 トラッキングシステムのアプリケーション — 本文書は治工具の基本的な機能を規定するものであるため、半導体素子製造メーカーや装置メーカー個々のアプリケーションやセキュリティ事項のトラッキングなどの踏み込んだアプリケーションは本文書の範囲外である。この基本仕様と共に、装置ユーザや装置メーカー、あるいはシステム供給者が個々に機能を追加することにより、求められる踏み込んだ機能が実現できるものである。

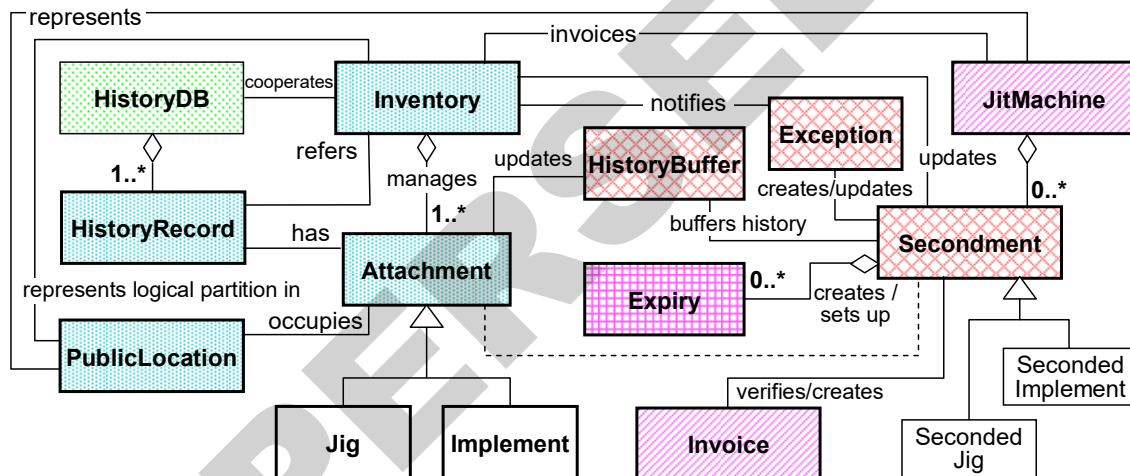


図 1 治工具トラッキングのモデル化

9 概念

9.1 ドメイン分析とオブジェクト指向モデリング — 複雑系をモデル化し設計するため、あるいは様々なアプリケーションで広範囲に使われるシステムのある部分を共通の仕様にするには何らかの洗練された手法が必要である。問題解決すべきドメインをオブジェクト指向方法論で分析することは取り得る最良の常套手段である。本技術は多くの SEMI スタンドアードでしばしば使われており、本文書で踏襲するにあたり特に支障はない。

9.1.1 UML による表現 — 上図は UML による治工具トラッキングのモデルである。本仕様に準拠するに際しては、オブジェクト指向プログラミング言語を用いてシステムを組む必要はない。UML が通常のオブジェクト指向の表現手段であるものの、オブジェクト指向の実装が望ましいが、必須と言うわけではない。

9.1.2 系全体を見据えたモデリング — トラッキングシステムとは一体何であるかを明確にするため、上図はトラッキング情報の収集のみならず、トラッキングシステム全系の基本的な管理事項をも説明するものである。

9.1.2.1 最小限のトラッキング情報の収集 — アタッチメントの確認、取り外される時までの履歴の蓄積、あるいはその報告の様な機能は装置上の治工具の仕様履歴を得るのに必要である。薄い赤色の交斜線を施した部分

(HistoryBuffer と Secondment)とその関連動作は収集機能に対応する。こう言った機能は通常、装置上に実装される。

9.1.2.2 許容限度到達時の動作 — 治工具であるアタッチメントのトラッキングの大きな目的の一つは適当ではないアタッチメントの使用による処理の誤りを防ぐことである。想定した時間や使用回数を越えてアタッチメントを使用した場合、そのアタッチメントは使用に適しないものと言えるだろう。それには使用に適さない状況になってアタッチメントが失効した時に求められる何らかの動作が行われるなどの機能が相当する。明るい赤色で格子模様を施したクラス(Expiry)とその関連動作はこの特別動作機能に対応するものである。本機能は通常、装置上に実装される。

9.1.2.3 アタッチメントの識別と照合 — 装置の制御部などの情報系は、たとえアタッチメントの ID が分かっていたとしても、そのアタッチメント使用区分は分からないかも知れない。それを理解する能力、ならびに、情報と実物のアタッチメントとを照合することは上記の基本的な機能の手助けになるのみならず、トラッキングを確実なものにするのにも役立つ。ピンク色の斜線を施したクラス(JitMachine と Invoice)およびその関連動作は本機能に相当する。本機能もまた、通常は装置に実装される。

9.1.2.4 トラッキング情報の管理 — インベントリ(保管管理データ)の様な情報系は収集されたアタッチメントトラッキング情報を効果的に管理し、利用するためには必要なものである。青色でやや濃い目に陰影を施したクラス(Attachment, HistoryRecord, Inventory と PublicLocation)およびその関連動作は本機能に相当する。本機能は通常、親装置やセルコントローラ、あるいはホストコンピュータに実装される。若しくはこれらプラットフォームに分散実装される。

9.1.2.5 さらに踏み込んだアプリケーションのインターフェース — 治工具であるアタッチメントのトラッキングの最終目標はアタッチメントの寿命を管理することや、アタッチメントの有効利用を高めること、処理の仕損じを防ぐこと、ならびに、望ましい品質の製品を作ることである。この意味で、ここでのインターフェース機能はアプリケーションを支援するものである。薄めの緑色の陰影を施したクラス(HistoryDB)は本機能に相当する。本機能はホストやアプリケーションサーバに実装されるだろう。

9.1.3 クラスモデルにおける協力関係 — 連係動作の例を以下に示す。

9.1.3.1 アプリケーションはある種類の治具を HistoryDB から探す。高価な製品の製造に適したものが幾つか見つかった。Inventory にそれがどこにあるか問い合わせる。

9.1.3.2 それを使用する装置へ治具を配送する。Inventory を更新し、Inventory は送付状を JitMachine に送る。Invoice オブジェクトが JitMachine 上で生成される。治具が個々にそれぞれの装置に到着した時点で Invoice は照合を行う。照合結果が正常であれば、その事実は Inventory に知らされて、その位置情報が更新され、Secondment が JitMachine に生成され、送られた治具が何であったかが分かる。同時に HistoryBuffer が Secondment により生成される。

9.1.3.3 装置が移動する。HistoryBuffer はイベントやプロセス条件や使用回数を一時蓄積する。ある時点で蓄積した内容が Inventory に報告される。Inventory は HistoryRecord を対応する更新を行う。

9.2 アタッチメント(Attachment) — アタッチメントは治具または工具を表す。この概念は両者に共通の性質を取り扱うために導入されるものである。

9.2.1 Attachment オブジェクト — Attachment オブジェクトは治具や工具などの補助用具を論理的に表したものである。Attachment オブジェクトは後述する Attachment クラスのインスタンス(個々に具体化したもの)である。Attachment オブジェクトは対応する実際のアタッチメントとその情報を一体にして他と識別するものである。このオブジェクトは本文書では「Attachment」と簡略化され、しばしば「オブジェクト」を抜かして記すことがある。

Attachment
showId showName changeName showType showHistory updateHistory showLocation updateLocation

図 2 Attachment クラス

9.2.2 *Attachment* クラス — *Attachment* クラスはそのアタッチメントに固有のデータ、対処される振る舞い、提供されるサービスを定義するものである。本クラスは *Inventory* に登録される治工具を一般化したものである。参考として本クラスを上記に図解する。

9.2.2.1 *Attachment* の使命 — *Inventory* に属す治具または工具の履歴などの情報を保持・整理する役目を負う。

9.2.2.1.1 この役目を遂行するにあたり、*HistoryRecord* の記録項目の作成／削除、イベントの追記、使用回数や時間の更新の依頼や問い合わせを受け付ける。これらは本仕様の中では *Attachment* のサービスとして定義する。サービスの名称は上記の機能とは異なることがある。

9.2.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で仕様される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.2.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスに明らかに状態を考えられるならば状態モデルは規定されるべきであるが、本クラスにはそれがなく、状態モデルはここでは規定しない。

9.3 *例外状態* — 例外状態は治工具やトラッキング能力に起こった異常な状態で、危険な状態を含む: 材料やアタッチメントや装置を壊す危険、あるいは既に壊している場合、爆発、人に障害を与えるか死に至らしむ場合など。例外状態は装置や、場合によっては保持管理部簿に対応する記憶機能で発生する。その事実は履歴に追記され得るものである。

9.3.1 *Exception* オブジェクト — *Exception* オブジェクトは通常、装置に起こった異常状態を表す。オブジェクトはそれが起こったときに生成され、異常状態から正常状態へ回復した後、消滅する。*Exception* オブジェクトは後述する *Exception* クラスのインスタンスである。本オブジェクトは本文書では「*Exception*」と簡略化され、しばしば「オブジェクト」を抜いて記すことがある。

Exception
showId showCategory showCode showDescription showSubject showState updateState destruct

図 3 Exception クラス

9.3.2 *Exception* クラス — *Exception* クラスは例外状態に固有のデータ、対処される振る舞い、提供されるサービスを定義するものである。参考として本クラスを上記に図解する。

9.3.2.1 *Exception* の使命 — 本クラスは状況を識別し、状態を判明させ、対象となる治部または工具を特定する役目を負う。

9.3.2.1.1 この役目を遂行するにあたり *Exception* は状態変化の通知と状態の追補情報を受け付ける。これらは本仕様では *Exception* のサービスとして規定する。サービスの名称は上記の機能とは異なる場合がある。

9.3.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.3.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスには明確に状態が存在し、下記の状態と、状態図と状態遷移定義表による状態モデルが定義される。

9.3.2.1.3.1 *SET* — 異常な状況がまだ解決されていない状態。

9.3.2.1.3.2 *CLEARED* — 異常な状況から脱し、その例外状態の経緯から見ると正常な状態。

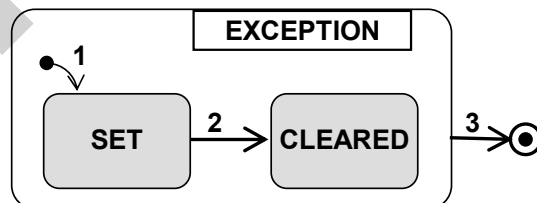


図 4 Exception の状態

表 1 *Exception* の状態遷移定義表

#	元の状態	トリガ	遷移後の状態	動作	記事
1	(No state)	異常状態が発生する	SET		
2	SET	異常状況から脱する	CLEARED	場合によって、例外状態の情報が履歴に追記される。	必ずしも終結とは限らない
3	CLEARED	終了の指示がでる	(No state)		

9.4 *Expiry* — *Expiry* は、治具または工具のある種の性質である使用に適する明確な期間を表現するものである：プロセスに使用した時間、負担の高いプロセス時間、使用回数、等々。*Expiry* は決められた期間について用意され、そのアタッチメントに関してそれが終了した際に特別な動作ができるようにするものである。

9.4.1 *Expiry* オブジェクト — *Expiry* オブジェクトはあるアタッチメントに対する有効期限切れの条件の1つに関する一連の情報を蓄えるものである。*Secondment* すなわち、装置上でのアタッチメントの表現、が生成された際、あるいは、有効期限切れ条件を指定する指示を *Secondment* が受け付けた際にこのオブジェクトが生成され、*Secondment* が装置から取り外された時に消滅する。有効期限が切れた際に特別な操作がなされた後に有効期限切れ条件を取り下げる場合にもこのオブジェクトは消滅する。*Expiry* オブジェクトは後述する *Expiry* クラスのインスタンスである。本オブジェクトは本文書では「*Expiry*」と簡略化され、しばしば「オブジェクト」を抜いて記すことがある。

Expiry
showId showExpiration changeExpiration showAction changeAction showDescription changeDescription showSubject showState destruct

図 5 *Expiry* クラス

9.4.2 *Expiry* クラス — *Expiry* クラスは有効期限切れ条件に係わる固有のデータ、対処される振る舞い、提供されるサービスを定義するものである。参考として本クラスを上記に図解する。

9.4.2.1 *Expiry* の使命 — 本クラスは治具または工具の有効期限切れに関する維持管理の役目を負うものである。

9.4.2.1.1 この役目の遂行にあたり、このクラスは対応するアタッチメントの有効期限切れ条件項目をチェックする。この作業は本クラスが生成された時から始められる。本仕様はこれを支援するサービスを定義するものである。サービスの名称は上記の機能と異なる場合がある。

9.4.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.4.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスには明確に状態が存在し、下記の状態と、状態図と状態遷移定義表による状態モデルが定義される。

9.4.2.1.3.1 *VALID* — アタッチメントは条件項目に関して有効条件内である。

9.4.2.1.3.2 *EXPIRED* — 有効期限の条件項目を外れているのでアタッチメントの使用は適当ではない。

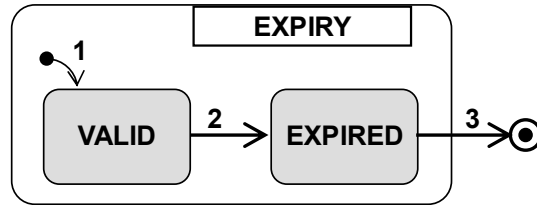


図 6 Expiry の状態

表 2 Expiry の状態遷移定義表

#	元の状態	トリガ	遷移後の状態	動作	記事
1	(No state)	Expiry オブジェクトが生成される	VALID	開始を開始する	
2	VALID	条件項目の期限に達する	EXPIRED	登録しておいた特別動作が実行される	
3	EXPIRED	対処動作がなされる	(No state)		

9.5 **History** — History は治工具に起こったことと、どう使われたかを追跡するための概念である。その機能を全うするために数種類のクラスオブジェクトをモデル化する。

9.6 **HistoryBuffer** — HistoryBuffer は装置にある治具または工具の一時的な履歴情報のことである。この情報は定期的に、あるいは何らかの動作の通過点毎に Inventory に報告される。報告されるとその情報は消去され、新たな情報に備える。

9.6.1 **HistoryBuffer オブジェクト** — HistoryBuffer オブジェクトは装置上でのアタッチメントの一時的な履歴情報のバッファ領域を表す。このオブジェクトはアタッチメントが装置上で生成された時に生成され、アタッチメントが取り外された時に消滅する。HistoryBuffer オブジェクトは後述する HistoryBuffer クラスのインスタンスである。本オブジェクトは本文書では「HistoryBuffer」と簡略化され、しばしば「オブジェクト」を抜いて記すことがある。

HistorBuffer
showAttachment showHistory updateHistory showReleaseCondition setReleaseCondition showState destruct

図 7 HistoryBuffer クラス

9.6.2 **HistoryBuffer クラス** — HistoryBuffer クラスはこの一時記憶に関わる固有のデータ、対処される振る舞い、提供されるサービスを定義するものである。参考として本クラスを上記に図解する。

9.6.2.1 **HistoryBuffer の使命** — 本クラスは装置にある治具または工具の一時的な履歴情報を保持し、Inventory に報告して消去する役目を負うものである。

9.6.2.1.1 その役目を遂行するにあたり、本クラスはアタッチメントに起こったこと、および使用時間や使用回数を含めどのように使われたかの情報を受け付ける。この操作は本仕様では HistoryBuffer のサービスとして定義する。サービスの名称は上記の機能と違う場合がある。

9.6.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.6.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスには明確に状態が存在し、下記の状態と、状態図と状態遷移定義表による状態モデルが定義される。

9.6.2.1.3.1 *Empty* — 履歴情報は蓄積されていない。

9.6.2.1.3.2 *Buffered* — 履歴情報が蓄積されており、それはまだ報告されていない。

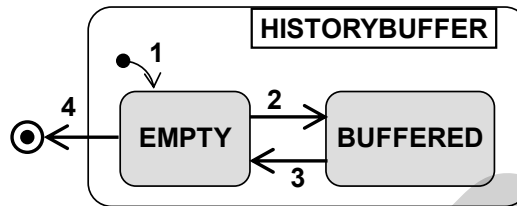


図 8 HistoryBuffer の状態

表 3 HistoryBuffer の状態遷移定義表

#	元の状態	トリガ	遷移後の状態	動作	記事
1	(No state)	Secondment が生成される	EMPTY		
2	EMPTY	履歴情報が記入される	BUFFERED		何らかの機会に情報は報告される
3	BUFFERED	履歴情報は報告され、情報は消去される	EMPTY	情報は消去の前に報告される必要がある。	
4	EMPTY	対応するアタッチメントが取り外される	(No state)		

9.7 履歴データベース — HistoryDB はアプリケーションから見て、履歴のデータベース全体へのインターフェースである。

9.7.1 履歴データベース・オブジェクト — 履歴データベース・オブジェクトは保管管理簿 (Inventory) に登録されている治具と工具すべての履歴データを保存する。履歴データベース・オブジェクトは後述する履歴データベース・クラスのインスタンスである。保管管理簿に対して1個だけ、履歴データベース・オブジェクトが存在する。本オブジェクトは本ドキュメントでは「HistoryDB」と簡略化され、しばしば「オブジェクト」が略される。

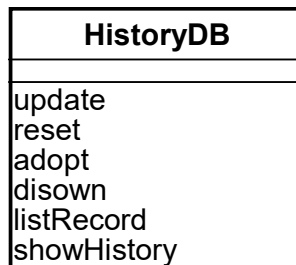


図 9 HistoryDB クラス

9.7.2 履歴データベース・クラス — HistoryDB クラスはデータベースに関する固有のデータ、振る舞い、およびサービスを規定する。本クラスは保管管理簿に登録されているすべての治具と工具の履歴を管理する。本クラスを参考のため上記に図示する。

9.7.2.1 *HistoryDB* の役目 — 本クラスは保管管理簿(Inventory)に登録されているすべての治工具の履歴を管理し整理する責務を負うものである。

9.7.2.1.1 その責務を全うするにあたり、アタッチメントの履歴を引継ぎ／リセット、更新の要求、問い合わせを受け付ける。本仕様では、これらは *HistoryDB* のサービスとして定義する。サービスの名称は上記の機能と同じでない場合がある。

9.7.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.7.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスに明確な状態が存在するならば、それは規定されなければならないが、本クラスには状態がないので、ここでは状態モデルは規定しない。

9.8 *History Record* — *HistoryRecord* は保管管理簿に登録されている個々のアタッチメントに対応した記録である。

9.8.1 *History Record* オブジェクト — *History Record* オブジェクトは特定の治具か工具の履歴データを保持する。*History Record* オブジェクトは後述する *History Record* クラスのインスタンスである。*History Record* オブジェクトは *HistoryDB* オブジェクトによって取りまとめられる(集約)。本文書では本オブジェクトは「*HistoryRecord*」と簡略化され、しばしば「オブジェクト」が略される。

HistoryRecord
fluctuate addEvent listEvents showEvent showHistory destruct

図 10 *HistoryRecord* クラス

9.8.2 *History Record* クラス — *HistoryRecord* クラスはアタッチメントの記録に関する固有のデータ、振る舞い、およびサービスを定義する。本クラスは *HistoryDB* に集約されている。本クラスは、*HistoryDB* がまとめる治具または工具の履歴を管理するものである。本クラスは参考のため上記に図示する。

9.8.2.1 *HistoryRecord* の役目 — 本クラスは *HistoryDB* に登録されている、ある治具または工具の履歴情報を保持し、整理する責務を負うものである。

9.8.2.1.1 その責務を全うするにあたり、本クラスは *HistoryRecord* の記録の削除、記録へのイベントの追加や使用回数/時間の更新の依頼、記録の問い合わせを受け付ける。To complete the responsibility it accepts removal of a *HistoryRecord*, adding events or increment time/duration requests and inquiry. これらは本仕様では *HistoryRecord* のサービスとして定義するものである。サービスの名称は上記の機能と違うことがある。

9.8.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.8.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスに明確な状態が存在するならば、それは規定されなければならないが、本クラスには状態がないので、ここでは状態モデルは規定しない。

9.9 *Inventory* — 工場内の治工具であるアタッチメントすべてにわたってトラックし管理するインベントリ(保管管理簿)なるものが存在する。インベントリは治具と工具の二つに分けた分割インベントリとしても良い。あるいは、管理の都合に合わせ、治工具の種類や分類によって幾つかの部分的なインベントリに分けても良い。たとえそうしたとしても、装置の種類毎に、あるアタッチメントの種類に関してインベントリのユーザからは一体に見える必要がある。インベントリは物理的な実態を言うのではなく論理上のものであるから、ある亜アタッチメントが他の種類の装置用の別のインベントリと共用されることもあり得、双方のインベントリも混乱なく管理されなければならない。

9.9.1 *Inventory* オブジェクト — そのインベントリは本仕様の多くの箇所で *Inventory* オブジェクトとして取り扱われる。それは時折「*Inventory*」と略記されるが、その場合は必ず大文字の「I」で書き始める。*Inventory* オブジェクトは後述の *Inventory* クラスのインスタンスである。トラッキングを行うシステムには唯1個だけ *Inventory* オブジェクトが存在する。

Inventory
update install uninstall showNamespace changeNamespace listUp showData

図 11 *Inventory* クラス

9.9.2 *Inventory* クラス — *Inventory* クラスは保管管理簿に係わる固有のデータ、振る舞い、ならびにサービスを定義するものである。本クラスはある定まった範囲の治工具総てを管理する。個々で言う範囲とは必ずしも空間的なものを言う訳ではなく、装置の種類やアタッチメントの種類、サプライヤやそれらの分類の組み合わせなどによる論理上のものである。本クラスを参考のため上記に図示する。

9.9.2.1 *Inventory* の役目 — 本クラスは登録されている工具の管理や整理を行う責務を負うものである。

9.9.2.1.1 その責務を遂行するにあたり、アタッチメントの登録／登録削除や更新依頼、問い合わせを受け付ける。これらは本仕様では *Inventory* のサービスとして定義される。サービスの名称は上記と異なる場合がある。

9.9.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンドード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するのである。

9.9.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスに明確な状態が存在するならば、それは規定されなければならないが、本クラスには状態がないので、ここでは状態モデルは規定しない。

9.10 **Invoice** — インボイスは装置に届く治具や工具などであるアタッチメントについてひとまとまりの情報である。その情報は実際にアタッチメントが届く前、あるいは後に装置に送られる。もしアタッチメントが届いた情報に問題がある場合、装置は新たなものを要求する。代わるアタッチメントか情報が届くと、装置は改めてチェックを行う。

9.10.1 **Invoice オブジェクト** — Invoice オブジェクトはまとまったインボイス情報を装置が受け取った時に生成される。もし装置がそのアタッチメントに関して複数のロードポートを持つ場合、その情報にはポート ID も明示される場合がある。ポート ID が明示されない場合、届け先のロードポートは不特定となる。もし、複数のアタッチメントやロードポートが指定されると、それらは可能性を示すが、組み合わせを指示したり関連性を示すものではない。しかし、アタッチメントとその履歴には当然、関連がある。情報がすべて照合されると、その結果の如何に拘わらず、本オブジェクトは消滅する。Invoice オブジェクトは後述する Invoice クラスのインスタンスである。本文書では本オブジェクトは「Invoice」と略記され、しばしば「オブジェクト」が抜かされる。

Invoice
showId showAttachments showTypes showLoadPorts showHistory

図 12 Invoice クラス

9.10.2 **Invoice クラス** — Invoice クラスはこのクラス固有のデータや振る舞い、およびサービスを定義するものである。

9.10.2.1 **Invoice の役目** — 本クラスはインボイス情報を保持し、整理し、照合する責務を負う。このオブジェクトは装置上の未照合のアタッチメントに関してこのオブジェクトが生成された時、あるいは新たに実物のアタッチメントは到着して本オブジェクトが生成された時、実物のアタッチメントをインボイス情報で照合する。

9.10.2.1.1 その責務を全うするにあたり、持っている情報への問い合わせを受け付ける。本仕様ではそれは Invoice のサービスとして定義する。サービスの名称は上記の機能と異なる場合がある。

9.10.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.10.2.1.3 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との関係動作と、状態モデルとで表現される。関係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスに明確な状態が存在するならば、それは規定されなければならないが、本クラスには状態がないので、ここでは状態モデルは規定しない。

9.11 **JitMachine** — JitMachine は完全な治工具のトラッキング能力をもつ1台の装置である。インベントリ(保管管理簿)がその役目を果たせる様、JitMachie はインベントリと強調動作を行う。

9.11.1 **JitMachine オブジェクト** — JitMachine オブジェクトは装置の治工具であるアタッチメントをトラッキングする役目を負う1台の装置を表わす。本オブジェクトは届けられたアタッチメントがその装置で使われるのが適当か、あるいは想定していた種類や同等品であっても適当ではないかの照合をする。同オブジェクトはあるアタッチメントが装置に届いたとかインベントリに向かったことをインベントリに通知する。JitMachine オブジェクトは後述する JitMachine クラスのインスタンスである。本文書ではこのオブジェクトは「JitMachi」と略記され、しばしば「オブジェクト」が抜かされて表現される。

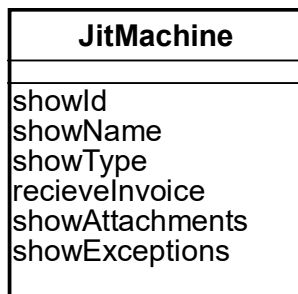


図 13 JitMachine クラス

9.11.2 JitMachine クラス — JitMachine は対応装置固有のデータや振る舞い、あるいはサービスを定義するものである。本クラスを参考のため上記に図示する。

9.11.2.1 JitMachine の役目 — 本クラスはアタッチメントの照合をし、装置への出入りを報告し、治具や工具を管理する責務を担う。

9.11.2.1.1 この責務を遂行するにあたり、本オブジェクトはどんなアタッチメントが届くのかを示すインボイスを受け取るとともに関連する要求を受け付ける。これらは本仕様では JitMachine のサービスとして定義されるものである。サービスの名称は上記の機能と異なることがある。

9.11.2.1.2 固有のデータは役目を全うするために使われる。このデータは属性とも呼ばれ、何らかの方法で規定される必要がある。本文書は一部の SEMI スタンダード文書に現れるような属性定義表によって直接、属性を定義することはせずに、サービスのパラメータに盛り込むことにより間接的に定義するものである。

9.11.2.1.2.1 振る舞いもまた、役目を全うするために定義される。振る舞いはしばしば、クラス外部との連係動作と、状態モデルとで表現される。連係動作は他のクラスとの関連の一部が反映される。関連動作は、関連動作するオブジェクトに定義されているサービスを依頼することによって動作するものであるが、依頼先のオブジェクトはクラス図において関連の相手側であるクラスのオブジェクトである。状態モデルはこのクラスのもう一つ別の振る舞いの表現である。本クラスは明確な状態が存在するので、下記の状態を定義し、状態図と状態遷移定義表で状態モデルを規定する。

9.11.2.1.2.2 INSERVICE — JitMachine はアタッチメントのトラッキングができる。

9.11.2.1.2.3 OUTOFSERVICE — JitMachine はアタッチメントのトラッキングができない状態。

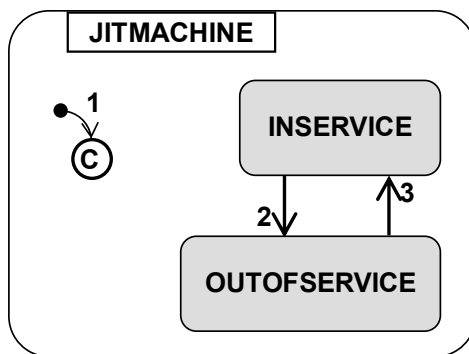


図 14 JitMachine の状態