

# Open MES 仕様書 Ver 1.0

ドラフト 1

生産システムモデル専門委員会

## 目次

1. 要約.....	1
2. はじめに .....	2
2.1 想定している読者 .....	3
2.2 なぜフレームワークなのか.....	4
2.3 なぜ CORBA なのか .....	5
2.4 フレームワークの範囲 .....	6
2.5 フレームワークのメリット.....	7
2.5.1 エンドユーザ.....	7
2.5.2 システムインテグレータ .....	7
2.5.3 コンポーネントプロバイダ.....	7
2.5.4 装置メーカー .....	7
2.6 仕様に関する規定・記述法.....	8
2.7 この仕様書の使用法.....	9
2.8 関連活動.....	10
2.8.1 OMG .....	10
2.8.2 ISO.....	10
2.8.3 MSTC/JOP .....	10
3. Open MES Objects.....	11
3.1 工場管理機能グループ .....	14
3.1.1 工場構成管理コンポーネント .....	14
3.1.2 設備スケジュール管理コンポーネント.....	16
3.2 製造指示管理機能グループ.....	23
3.2.1 製造指示管理コンポーネント .....	23
3.2.2 製造ロット作成管理コンポーネント .....	25
3.3 製品仕様管理機能グループ.....	27
3.3.1 製品仕様管理コンポーネント .....	27
3.4 工程管理機能グループ .....	31
3.4.1 製造ロット管理コンポーネント .....	31
3.4.2 工程内作業管理コンポーネント .....	35
3.4.3 投入指示管理コンポーネント .....	38
3.4.4 搬送指示管理コンポーネント .....	40
3.5 資材管理機能グループ .....	42
3.5.1 資材管理コンポーネント .....	42
3.6 搬送管理機能グループ .....	45
3.6.1 搬送管理コンポーネント .....	45

3.7 工程仕様管理機能グループ.....	49
3.7.1 工程経路管理コンポーネント.....	49
3.8 設備管理機能グループ.....	58
3.8.1 設備管理コンポーネント.....	58
3.9 スケジュール管理機能グループ.....	74
3.9.1 スケジュールラインタフェースコンポーネント.....	74
3.10 共通機能グループ.....	75
3.10.1 イベント通知管理コンポーネント.....	75
3.10.2 リモートエンティティコンポーネント.....	76
3.10.3 状態管理コンポーネント.....	77

## 改定履歴

版	99/05/27	JOP/FAM 内リリース
1 版	99/08/26	
2 版	99/09/14	JOP 内リリース
1 版	99/12/29	JOP/FAM 成果報告書

## 1. 要約

本仕様書は加工組み立て生産ラインのための MES アプリケーションフレームワーク(以下 Open MES フレームワーク)の仕様を規定しており、本書の読者として、フレームワーク、コンポーネントおよびアプリケーション開発者を想定している。

ディスクリット系 MES の開発においては、これまで共通したインタフェースが存在せず、またビジネスの競争上、各企業・工場ごとに管理ノウハウが異なっていたため、個別の作り込みを行っていた。しかし実は MES には共通なモデルが存在し、そのモデルに従ったソフトウェア部品を用意することにより、個々の生産現場にあった MES の構築を、必要なソフトウェア部品の組み合わせとそのカスタマイゼーションによって行うことが可能である。

本書に記述された MES フレームワークを用いるメリットは以下のとおりである。

- SCM/ERP との連携によって全社的な最適化が可能になる。現場の情報が SCM/ERP に反映されるため、販売、生産、調達、物流にわたる最適化が行なえるようになる。
- 生産ライン改善のための Plan Do Check Action に必要なデータが得られる。それぞれの局面において必要なデータが生成・加工される。
- Web Browser によって生産指示情報、生産実績情報にどこからでもアクセスすることが可能になり、意志決定が迅速・正確になる。
- 生産装置のマルチベンダ環境を容易に構築できる。
- これまでに開発・使用されたコンポーネントを用いることで短時間で高品質な MES を構築できる。
- ノウハウを組み込んだコンポーネントの提供によるビジネスが可能となる。

本書で記述されているフレームワークの対象範囲は、

- 工場管理
- 製造指示管理
- 製造仕様管理
- 工程仕様管理
- 工程管理
- 設備管理
- 搬送管理
- 資源管理
- スケジュール管理

であり、この仕様をもとに GUI 開発、アプリケーション開発、データのカスタマイズ、CORBA インタフェースのカスタマイズ、フレームワーク機能の追加を行なうことができる。

## 2. はじめに

本仕様書は、電子商取引共通基盤整備事業の一環として情報処理振興事業協会より製造科学技術センターが開発を請け負った OpenMES アプリケーションフレームワークの仕様を記述している。

一般に仕様書を記述する目的は、プロジェクトの成果を普及させるために

- 標準化すべき仕様を規定する。
- プロジェクトで開発したプログラムをベースにして、個々の MES を開発するために必要な仕様をまとめる。
- プロジェクトの成果をまとめる。

ことである。ただしこれらは一意の仕様にはならない。たとえば「プロジェクトの成果をまとめ」た仕様には、そこで開発した全ての仕様が記述されており、必ずしも「標準化すべき仕様」とは一致しない。「標準化すべき仕様」とはインターオペラビリティを確保し、プラグインを可能にすることにより、オープンな世界を実現するために必要十分な仕様である。実装に依存する仕様は含まれない。

本仕様書では2番目の「プロジェクトで開発したプログラムをベースにして、個々の MES を開発するために必要な仕様」を規定する。プロジェクト全体の仕様に関しては、これをベースにして、フレームワークのいわゆる“Cooking Book”を後日執筆したい。「標準化すべき仕様」に関しては、製造科学技術センターの FA オープン推進協議会生産システムモデル専門委員会での今後の議論にゆずる。

## 2.1 想定している読者

本書が想定している読者は次の3者である。

- アプリケーションプログラマ
- コンポーネントプロバイダー
- フレームワークプロバイダー

アプリケーションプログラマーは Javadoc の仕様を参照してアプリケーションを開発することができる。

コンポーネントプロバイダーは同様に Javadoc の仕様を参照して再利用を前提としたアプリケーションフレームワークを開発することができる。

フレームワークプロバイダーは Javadoc の仕様および UML による仕様を参照にしてフレームワークを実装し、また拡張することができる。ただし、実装のための詳細な情報は本書に記述されていないので、各プロバイダでパフォーマンス、パーシステンス、セキュリティ、その他実用時に必要とされる事項について設計を行なう必要がある。

## 2.2 なぜフレームワークなのか

MES ( Manufacturing Execution System ) とは生産システムにおいて ERP 等からの生産計画および生産指示を受け、製造設備に作業指示を出し、実績収集、進捗管理を行なう工程管理を中心とした統制システムである。MES の構築、調達において次の問題が存在する。

- 生産システムは、数多くのサブ・システムから構成されるが、標準化されたインターフェースが存在せず有機的に統合することが難しい。
- 生産システムは、他種類の製造設備から構成されるマルチ・ベンダー環境が普通であるが、製造設備を管理する標準的なインターフェースが存在しないため、個別の作り込みによって製造設備をシステムに取り込む必要がある。
- 生産現場の環境の多様さに起因する様々な特殊要件が存在するため、システムのパッケージ化が難しく、生産現場ごとに個別に作り直す必要がある。

こうした理由から、生産システムの開発・保守には多大なコストがかかり、また真に統合された生産システムが存在しないために、システム全体にわたる生産の最適化を計ることが困難となっている。

しかし実は MES には共通なモデルが存在し、そのモデルに従ったソフトウェア部品を用意することにより、個々の生産現場にあった MES の構築を、必要なソフトウェア部品の組み合わせとそのカスタマイゼーションによって行うことが可能であると我々は考えている。このようなアプローチの実現のためにオブジェクト指向を採用した。オブジェクト指向によって、生産システムをモデル化しソフトウェア部品に分解することができる。また同時に、これらのソフトウェア部品間の連携を定義することにより、構築される生産システムのテンプレートを予め与えることができる。このテンプレートは MES アプリケーションフレームワークと呼ばれ、新たな機能部品の追加や、既存機能部品の振る舞いの変更などにより、必要なカスタマイゼーションを容易に行うことが可能である。

MES アプリケーションフレームワークには機能面以外にも次の要件が存在する。

- 小規模な生産ラインから大規模な生産ラインまで対応できるスケーラビリティを持つこと
- ハードウェア、OS 等のプラットフォームに依存しないオープン性を持つこと

このため CORBA の分散オブジェクトを用い、複数のサーバにオブジェクトを分散させ、また実装言語として JAVA を採用することによりスケーラビリティとオープン性を確保する。

こういった MES アプリケーションフレームワークを導入することにより、現場の情報化が実現し、必要なデータにどこからでもアクセスできるようなシステムを短期間・低コストでユーザ自身が構築することが可能になる。さらにこのような情報化を推し進めることにより、ERP ( Enterprise Resource Planning )、SCM ( Supply Chain Management )、設備制御システムと協調して最適な生産が実現可能となる。

### 2.3 なぜ CORBA なのか

MES は複数の設備、サーバ、クライアント端末から構成され、本質的に分散処理が要求される。また分散処理によって小規模な生産システムから複数の工場を含む生産システムまでサポートできるスケーラビリティを実現することができる。さらに拡張性、オープン性の観点からソフトウェアのコンポーネント化、プラグイン性が要求されている。このような環境を実現するプラットフォームとして最近は

- DCOM
- CORBA

が注目されている。MES の場合、ハードウェアに関しては、計算機の種類が多岐に渡り、PC、ワークステーション、ビジネスコンピュータ、大型コンピュータから構成される。このような環境でシームレスなシステムを構築していくためにはプラットフォームに依存しない CORBA が適している。

## 2.4 フレームワークの範囲

MES が管理対象とする製造工程には、一般的に下記の三つの形態があると考えられる。

- 連続プロセス
- バッチ・プロセス
- ディスクリート・プロセス

オープン MES では、この内のディスクリート・プロセスをモデル化の対象として選択した。ディスクリート・プロセスとは、ロットなどの単位でまとめられたワークが、工程ごとに区切られた製造設備間を移動しながら、加工を施されるものである。機械加工や組立の工場がこれに相当する。

ディスクリート・プロセスには、さらにコンベヤ・ラインの様に固定された工程経路をとるフロー・ショップと、工程経路に自由度のあるジョブ・ショップに分かれる。オープン MES では、ジョブ・ショップに対応したモデルを用意し、その特殊形としてフロー・ショップに対応する。

フレームワークに含まれる機能は以下の通りである。詳しくは以下の章で説明する。

- 工場管理
- 製造指示管理
- 製造仕様管理
- 工程仕様管理
- 工程管理
- 設備管理
- 搬送管理
- 資源管理
- スケジュール管理

## 2.5 フレームワークのメリット

### 2.5.1 エンドユーザ

MES 導入によるメリットは次の通りである。

- Web Browser によって生産指示情報、生産実績情報にどこからでもアクセスすることが可能になり、意志決定が迅速・正確になる。例えば営業員は在庫状況、仕掛かり状況から納期を返答することが可能になり、また現場管理者は来期の生産スケジュールを知ることにより、負荷の集中や仕掛かり在庫を減少させることが可能になる。
- SCM/ERP との連携によって全社的な最適化が可能になる。現場の情報が SCM/ERP に反映されるため、販売、生産、調達、物流にわたる最適化が行なえるようになる。
- 生産ライン改善のための Plan Do Check Action に必要なデータが得られる。Plan はスケジューリング、Do は作業指示、Check は進捗管理、Action は実績データであり、それぞれの局面において必要なデータが生成・加工される。また作業日報等をわざわざ作成することはない。
- MES による指示により複数の工場があたかも一つの工場として機能することができる。製品種別ごとに工場の組み合わせを柔軟に変更することができる。
- MES フレームワークのメリットとしては既存のソフトウェアコンポーネントを調達してシステムを構築できることがあげられる。

### 2.5.2 システムインテグレータ

MES アプリケーションフレームワーク導入によるメリットは次の通りである。

- 生産装置のマルチベンダ環境を容易に構築できる。
- これまでに開発・使用されたコンポーネントを用いることで短時間で高品質な MES を構築できる。
- Web を用いた先進的なソリューションを提案、提供できる。

### 2.5.3 コンポーネントプロバイダ

MES アプリケーションフレームワーク導入によるメリットは次の通りである。

- 自社のノウハウを組み込んだコンポーネントの提供によるビジネスが可能となる。

### 2.5.4 装置メーカー

MES アプリケーションフレームワーク導入によるメリットは次の通りである。

- 生産システムごとの、装置の繋ぎ込みが不要となる。

## 2.6 仕様に関する規定・記述法

通常 CORBA を採用したフレームワークの仕様は IDL で記述されることが多い。しかし、本プロジェクトでは、次の 2 つの理由から Java の API として仕様を定める。

- アプリケーションプログラマが CORBA の仕様に精通していなくとも開発できるようにする。
- フレームワークのパフォーマンスを高めるため、必要なオブジェクトだけを CORBA オブジェクトとする。サーバとクライアントとで共有する必要がある非 CORBA オブジェクトは両者の間での値渡しとする。この機能は現在 CORBA2.0 では規約化されていない。Visibroker のベンダ固有の機能として提供されている。したがって、この機能は標準的な IDL では記述できない。

Java クラスは階層的なグループに分類されている。最上位のグループを機能グループ、それを分割したものをコンポーネントと呼ぶ。コンポーネントはクラスおよびインタフェースから構成される。本仕様書はクラスまたはインタフェースの機能について記述している。メソッドおよびその引数についての詳細は付録（別冊？）の Javadoc による仕様書を参照していただきたい。

グラフィカル表記は UML に準じている。

クラス階層図、シーケンス図において導出クラスの例が現れることがある。本仕様書では導出クラスの Prefix として “FMS” を用いる。

## 2.7 この仕様書の使用法

フレームワークのカスタマイズには次の5段階が考えられる。

- GUI 開発  
Visual Age Java, Jbuilder, Visual Cafe 等の開発環境あるいは JDK を用いて、フレームワーク API を呼び出す Graphical User Interface の開発・変更を行なう。
- アプリケーション開発  
フレームワーク API を呼び出すアプリケーションのロジックをクライアント側あるいはサーバ側の実装する。サーバ側の場合、クライアントとの HTML の授受にはサーブレットや JSP を用いるのであれば、その設計・実装も行なう。
- データのカスタマイズ  
個々の MES ごとにデータをカスタマイズすることを想定しているクラスが用意されているので、必要なメンバ変数・メソッドを追加する。
- CORBA インタフェースのカスタマイズ  
設備インタフェース、スケジューラインタフェースなど CORBA インタフェースにメンバ変数・メソッドを追加する場合は、CORBA インタフェースの導出インタフェースを定義し、そのインタフェースを実装する。それに伴い、新しいメソッドを呼び出すアプリケーションを開発することになる。また CORBA クラスに変数を追加する場合、実装クラスの導出クラスを定義・実装する。CORBA インタフェースを変更せず、ロジックを変更する場合は、実装クラスの導出クラスを定義し、メソッドを実装する。
- フレームワーク機能の追加  
仕様書で規定されているクラス定義を呼び出すことにより、再利用を考慮した MES アプリケーションフレームワークの機能グループ・コンポーネントを追加する。分散処理が必要な場合、あるいは適切な場合は CORBA インタフェースを新規に定義し、実装する。

## 2.8 関連活動

### 2.8.1 OMG

Manufacturing Domain Task Force では MES の RFI が提出され、Response が 6 個提出された。現在それを基に 3 つの RFP が準備されている。

### 2.8.2 ISO

TC 184/SC5/WG1 (Enterprise Integration) で New Work Item を審議中であり、日本国内対策委員会として OpenMES を検討している。

### 2.8.3 MSTC/JOP

生産システムモデル専門委員会で仕様について審議中である。

### 3 . Open MES Objects

Open MES フレームワークは次の 10 機能グループから構成される。

- 工場管理機能グループ
- 製造指示管理機能グループ
- 製造仕様機能グループ
- 工程仕様機能グループ
- 工程管理機能グループ
- 設備管理機能グループ
- 搬送管理機能グループ
- 資源管理機能グループ
- スケジュール管理機能グループ
- 共通機能グループ

本章では、これらの機能グループに含まれるクラスおよびインターフェースについて記述する。

図 3-1 に製造指示クラス ( MesProductionOrder )、ロットクラス ( MesProductionLot )、ロットジョブクラス ( MesLotJob )、プロセスジョブクラス ( MesProcessJob )、作業指示クラス ( MesWorkOrder )、抽象工程設備クラス ( MesProcessEquipment ) の関係を示す。製造指示を表現する MesProductionOrder オブジェクトはロットを表現する複数の MesProductionLot オブジェクトから構成される。MesProductionOrderManager オブジェクトはシステム内に存在する複数の MesProductionOrder オブジェクトを管理している。MesProductionLot オブジェクトに関する作業を MesLotJob オブジェクトは表現しており、個々の工程 ( 加工、検査、洗浄等 ) を表現する MesProcessJob オブジェクトから構成される。作業指示を表現する MesWorkOrder オブジェクトは MesLotJob オブジェクトと関連付けられており、ディスパッチャ ( MesDispatcher ) によって抽象工程設備 MesProcessEquipment オブジェクトに差し立てられる。

図 3-2 にこれらのクラス間のコラボレーション図を示す。アプリケーション ( 図中 AP 製造指示管理 ) が MesProductionOrderManager の addProductionOrder メソッドを呼び出し、製造指示 ( MesProductionOrder オブジェクト ) を作成する。次に createProductionLots メソッドを呼び出すことによって、MesProductionLot オブジェクトが生成され、releaseProductionOrders メソッドの呼び出しによって、MesLotJobManager に対し MesLotJob の生成を指示する。次にアプリケーション ( 図中 AP 製造ロット管理 ) が createSchedule メソッドを呼び出すことにより、スケジュールを作成し、outputSchedule メソッドの呼び出しにより、作成されたスケジュールをフレームワークに取り込む。releaseLotJobs メソッドにより、生産ラインにロットが投入され、addWorkOrder メソッドにより、作成されたスケジュールに基づき MesWorkOrder ( 作業指示 ) を MesProcessEquipment ( 工程設備 ) に割り付ける。作業が終了したら finishWorkOrder メソ

ッドが呼び出され、setWorkResult メソッドにより、関係している MesWorkOrder , MesLotJob , MesProcessJob , MesProductionOrder オブジェクトに作業実績が関連づけられる。

クラス図/製造関係

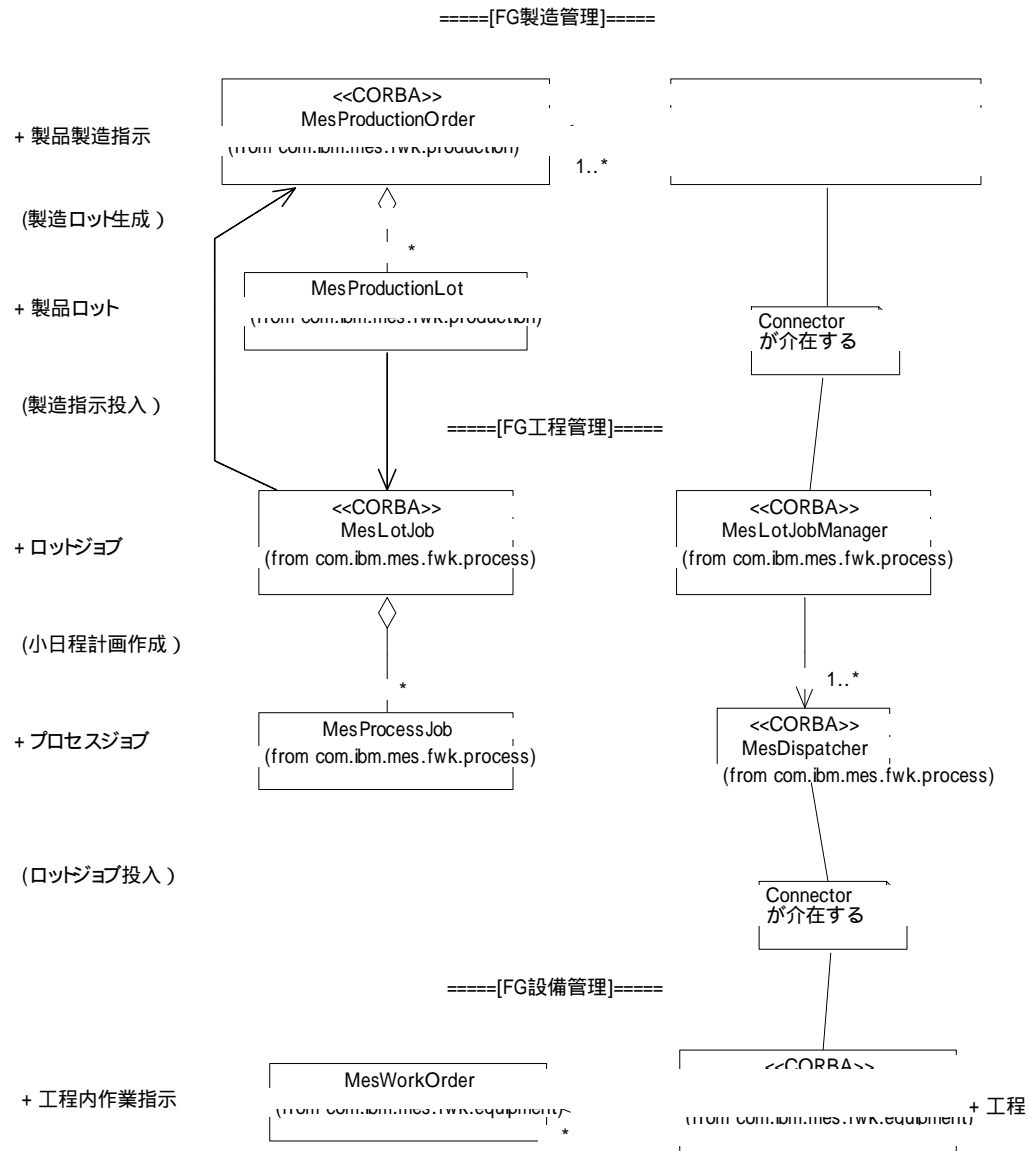


図 3-1 製造関連クラス図

コラボレーション図/UG計画管理

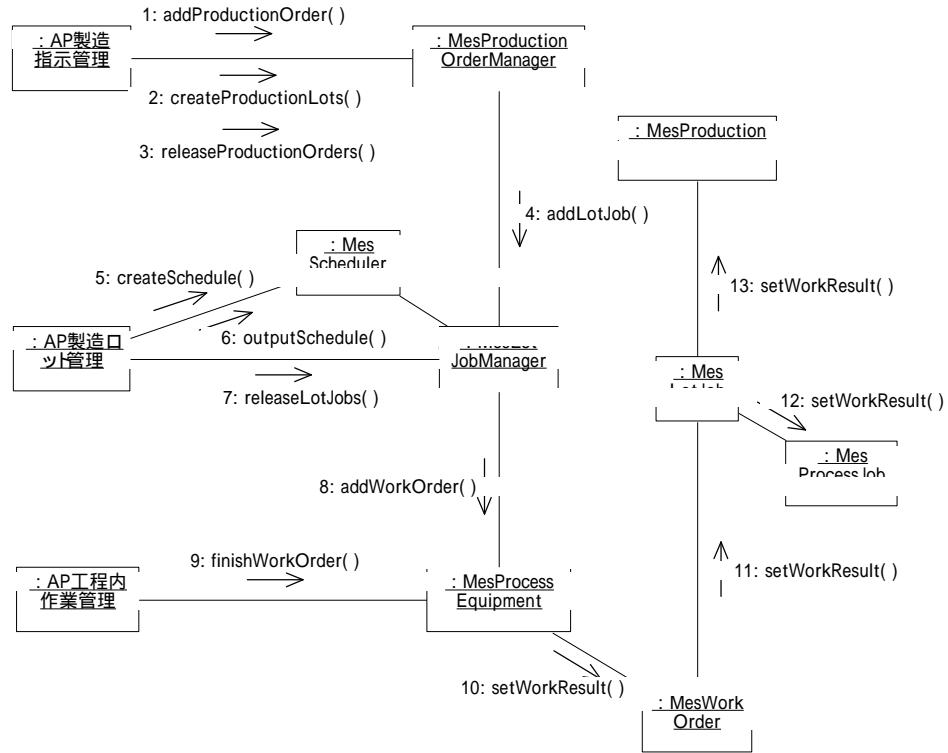


図 3-2 製造関連クラスコラボレーション図

実際にフレームワークを稼働させるために CORBA インタフェースを実装したクラスが必要である。CORBA インタフェース名を “ Foo ” とした場合、実装クラス名を “ FooImpl ” そのオブジェクトを立ち上げ管理するクラスを “ FooServer ” としている。“ FooImpl ” に関してはメソッド定義が “ Foo ” と同一のため、本仕様書では記述しない。ただし、クラス図やシーケンス図においては理解を容易にするため出現することがある。

### 3.1 工場管理機能グループ

本機能グループは工場の構成と稼働スケジュールを管理する。

- 工場構成管理コンポーネント
  - 設備スケジュール管理コンポーネント
- から構成される。

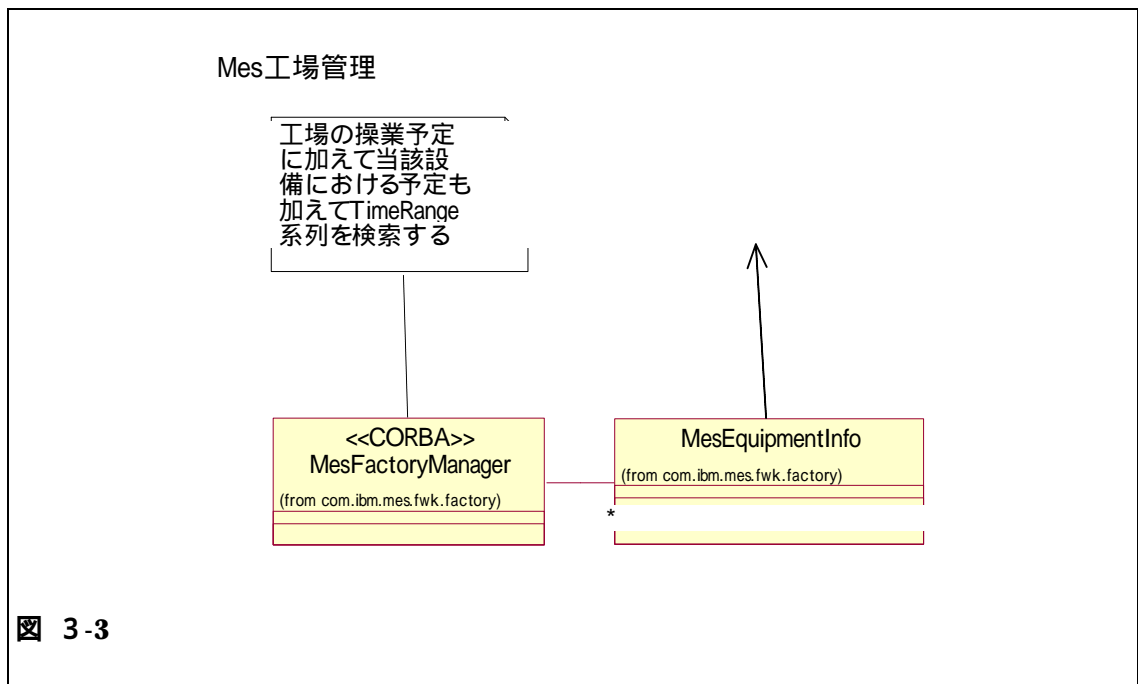


図 3-3

#### 3.1.1 工場構成管理コンポーネント

本コンポーネントは工場に設置されている設備情報を管理する。

##### (1) MesFactoryManager

本インタフェースは製造設備情報、工場オペレーション・プラン、設備オペレーション・プランおよび保守計画に対する操作（設定、検索、変更、削除）を定義する。さらに、設備休止時間帯、ある時間帯における使用可能設備を取得するメソッドを定義している。

##### (2) MesEquipmentInfo

本クラスは製造設備の情報すなわち種別、型番、設置場所の値を管理する。MES ごとにカスタマイズすることを想定している。

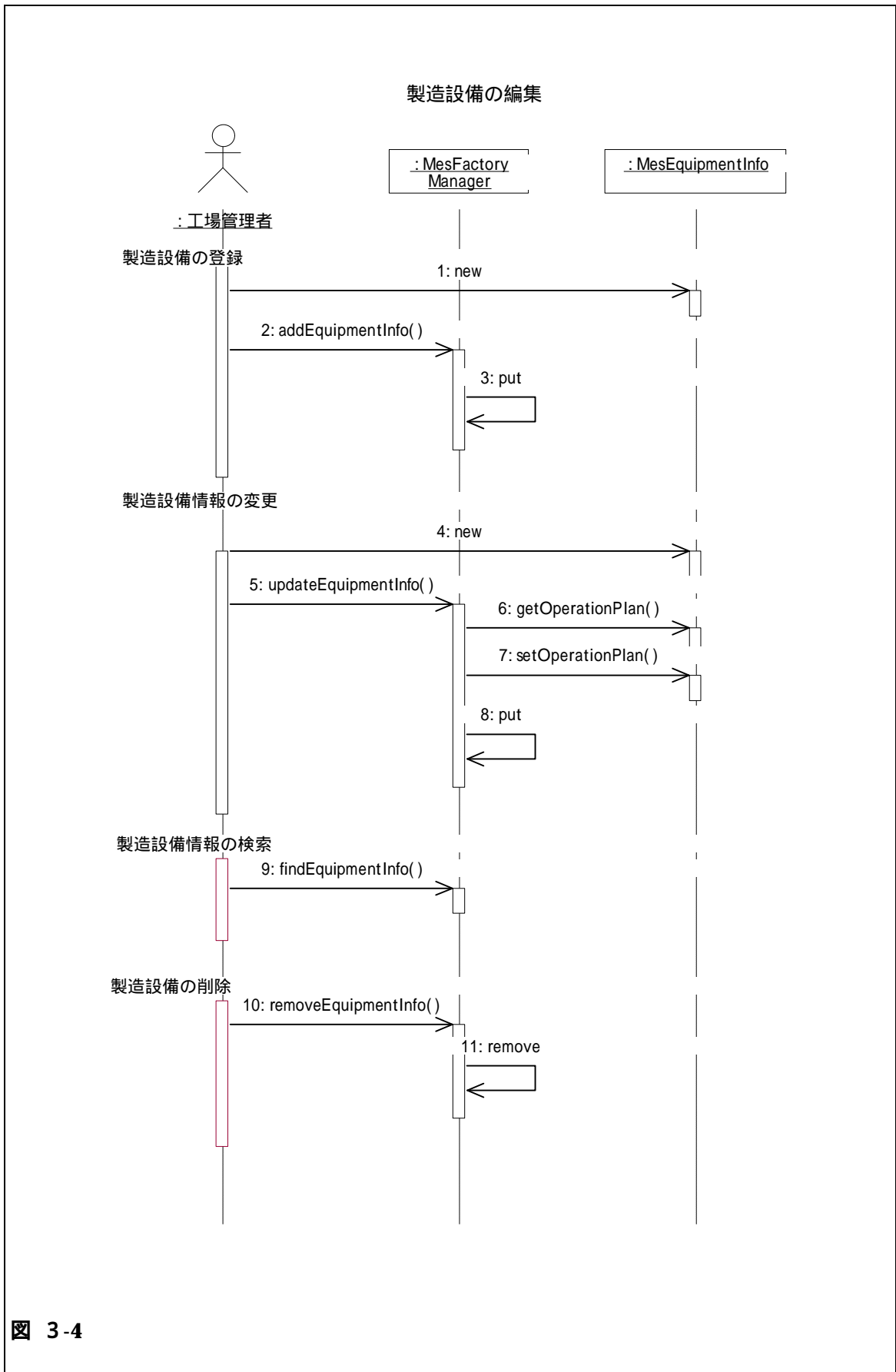
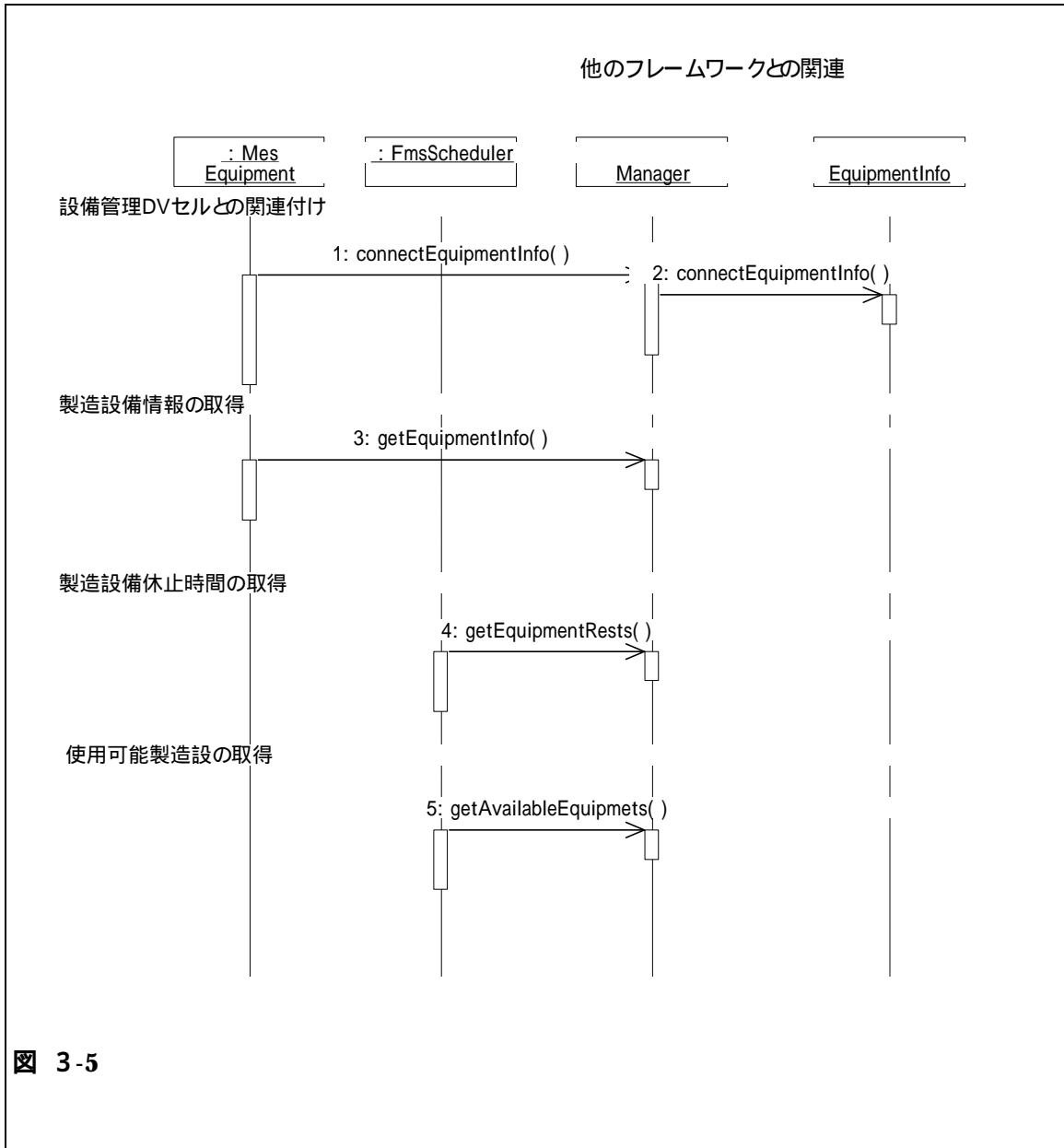


図 3-4



### 3.1.2 設備スケジュール管理コンポーネント

本コンポーネントは工場および個々の設備スケジュールを管理する機能を提供するクラスから構成される。設備の保守時期も本コンポーネントを用いて表現される。またスケジュールに対する実績値も管理する。

スケジュールは休止期間の集合として定義される。工場スケジュールは工場の休止期間、設備スケジュールは設備の休止期間の集合を表す。保守に関しては保守期間を表す。スケジューラが稼動期間として認識するのは、工場の休止期間と設備の休止期間の論理和の補

集合に相当する期間である。

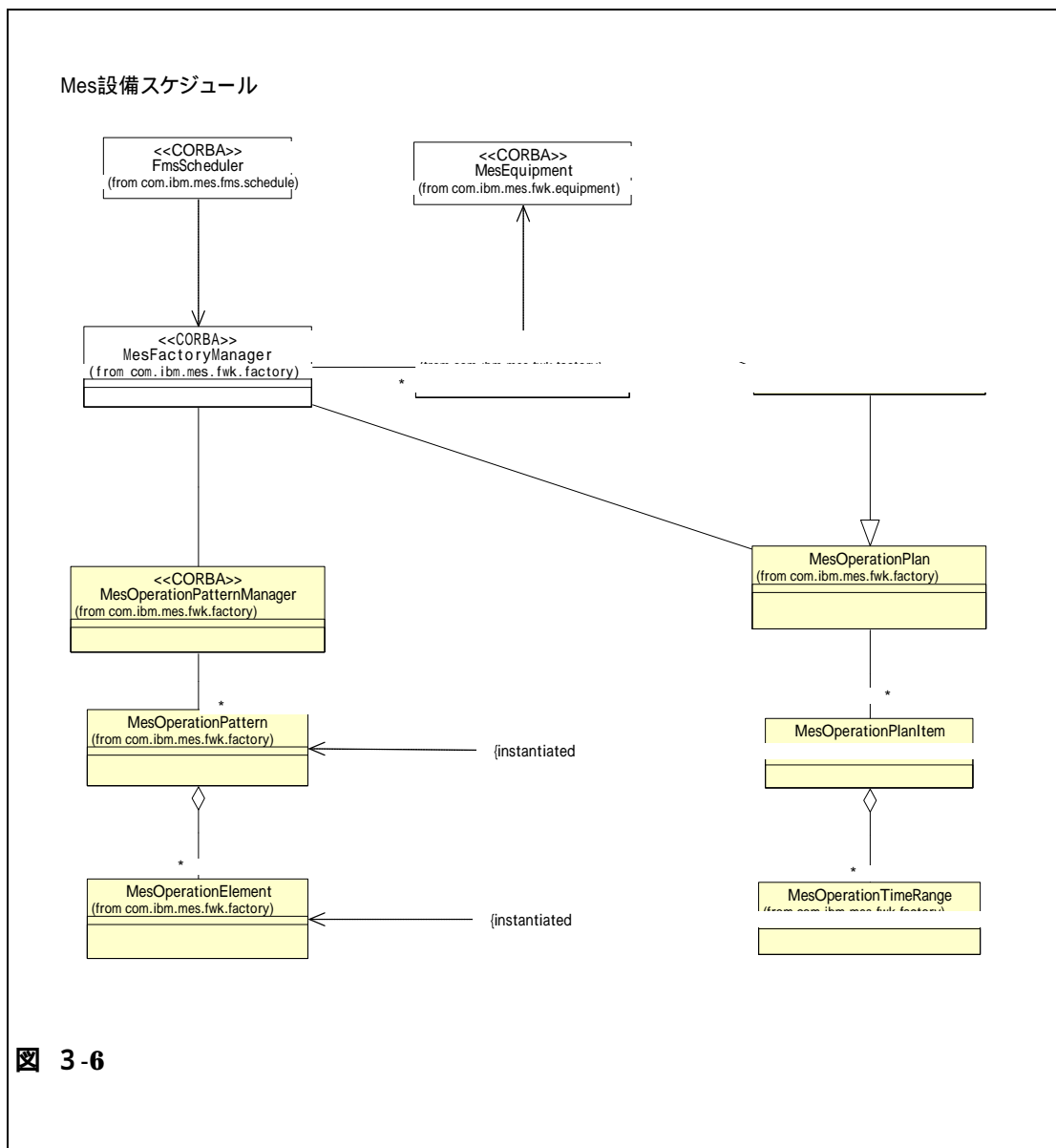


図 3-6

( 1 ) **MesOperationPatternManager**

本インタフェースは製造設備の管理と工場オペレーションの計画パターン に対する操作（設定、取得、検索、変更、削除）をおこなうメソッドを定義する。

( 2 ) **MesOperationPattern**

本クラスは MesOperationElement の集合体である。MesOperationElement 間の Interval を指定することにより、スケジュールのパターンを規定する。Interval の単位は分である。週毎であれば、 $7*24*60=10080$  となる。

( 3 ) **MesOperationElement**

本クラスはスケジュールの時間パターン要素を表す。開始時間からの Offset と期間で定義される。単位は両者とも分である。

( 4 ) **MesOperationPlan**

本クラスはスケジュールに対する操作（設定、取得、削除）を提供する。異なるパターンを持つ MesOperationPlanItem の集合を保持する。

( 5 ) **MesEquipmentOperationPlan**

本クラスは設備スケジュールおよび保守管理 に対する計画を持つ。

( 6 ) **MesOperationPlanItem**

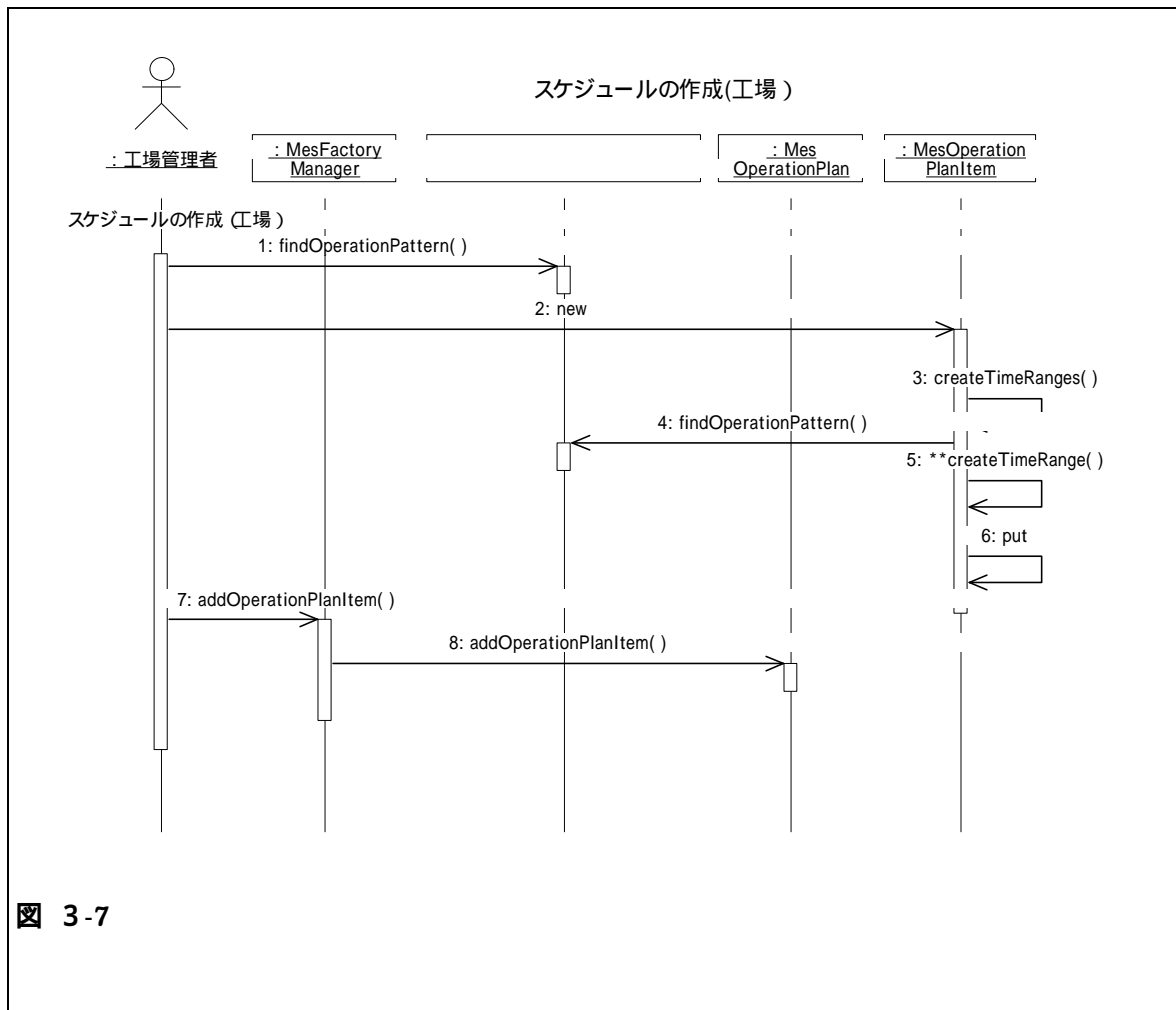
本クラスは MesOperationPattern を参照して、生成される一つのスケジュール要素である。開始日時と繰り返し回数が指定される。MesOperationTimeRange の集合体を保持する。

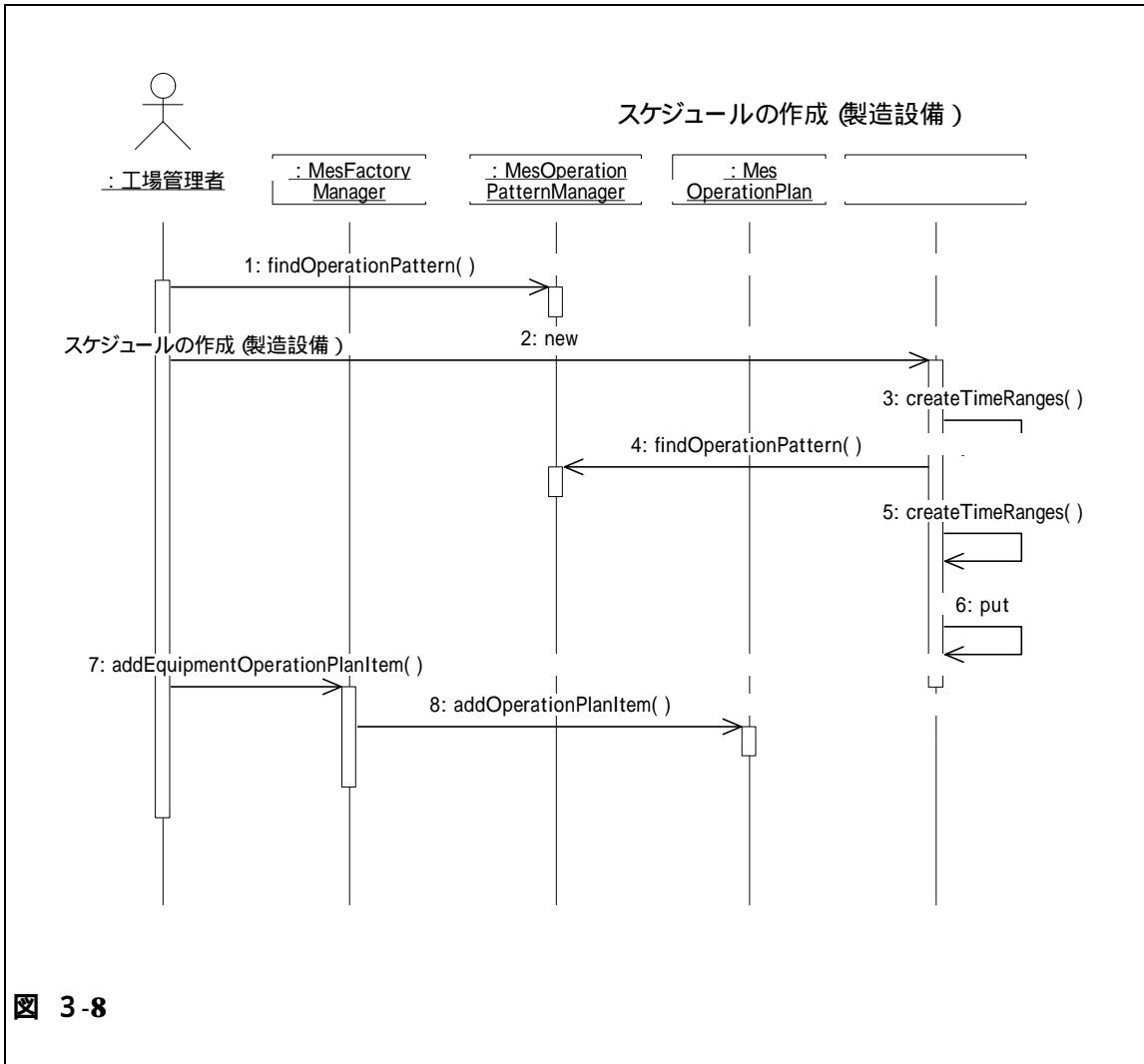
( 7 ) **MesOperationTimeRange**

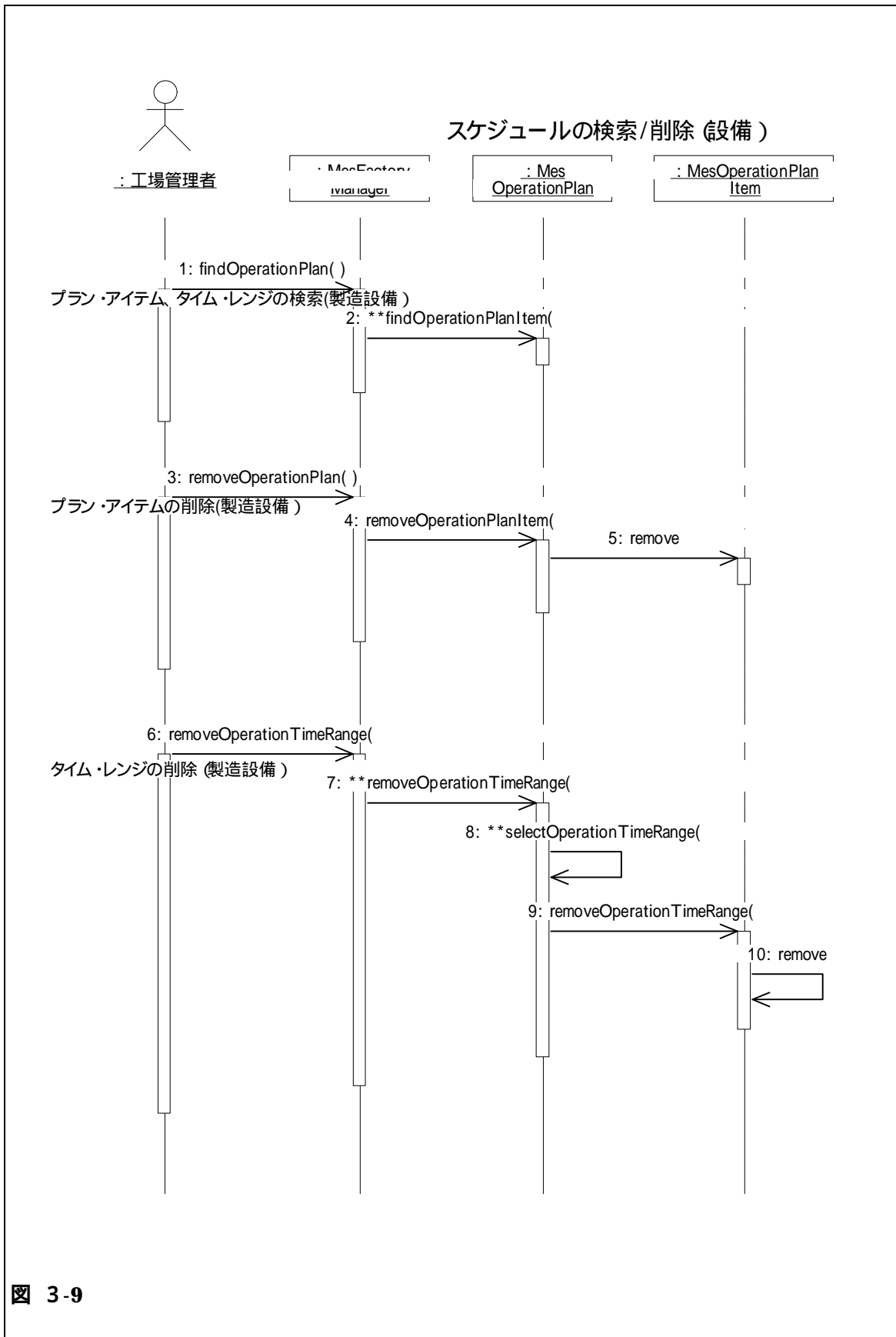
本クラスはタイム・レンジを表し、予定と実績を管理する。開始日時、期間で定義される。期間の単位は分である。

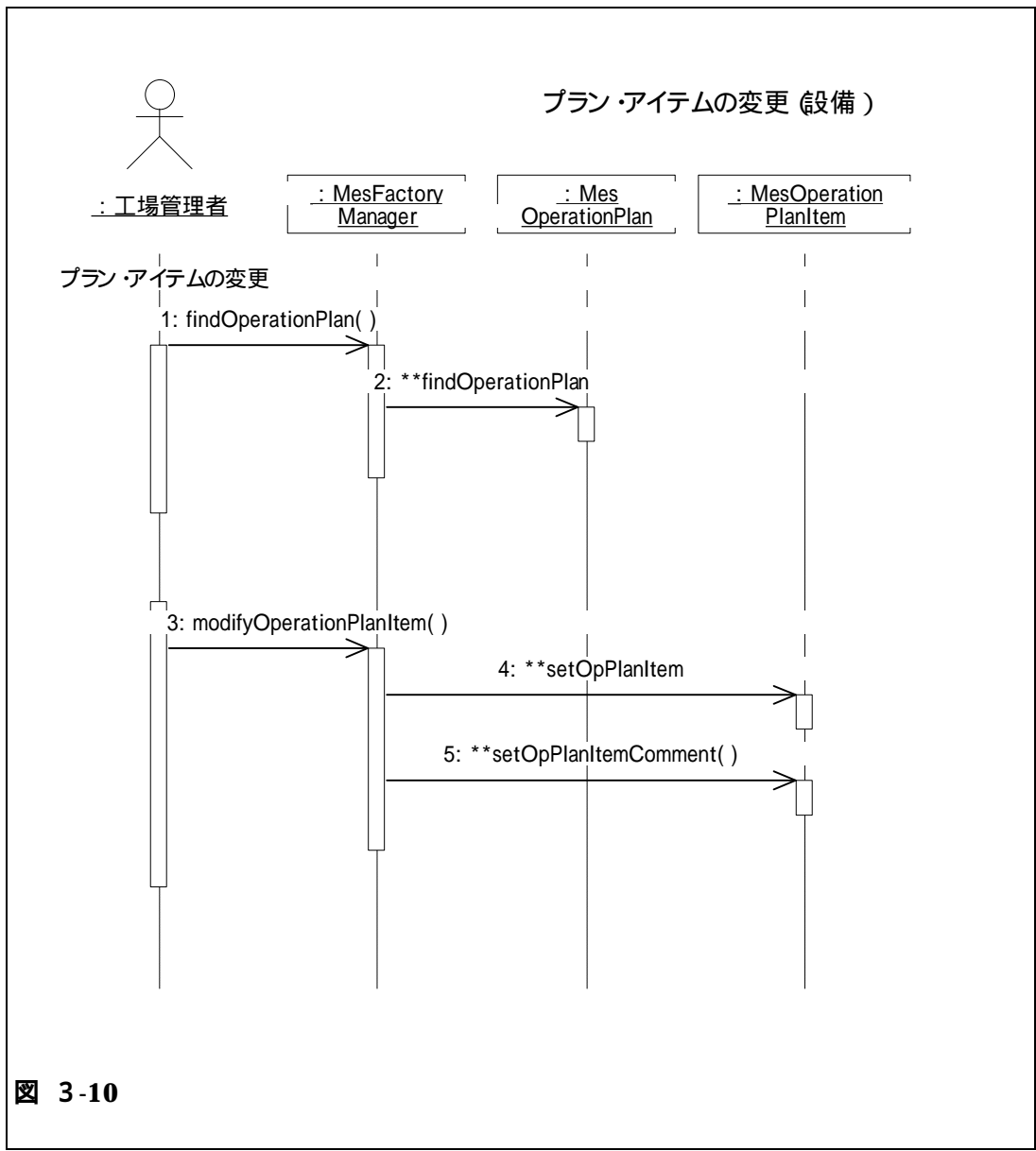
( 8 ) **MesPlanTimeRange**

タイム・レンジを定義する。休止開始時刻、期間で定義される。このクラスは、休止時間の計算結果として Hashtable の要素として返される。









### 3.2 製造指示管理機能グループ

製造指示管理機能グループは、生産計画システムとのインターフェースを提供する。製造指示を受けて必要な数のロットを作成し、工程に投入し、その製造指示の進捗を監視することができる。すなわち、製造指示と工程内を流れるロットの間の紐付けを管理する。

この紐付けは、受注生産か見込み生産かといった生産形態、MRP や製番管理といった管理方式、ロット分割など、様々な要因で変わってくる可能性がある。

#### 3.2.1 製造指示管理コンポーネント

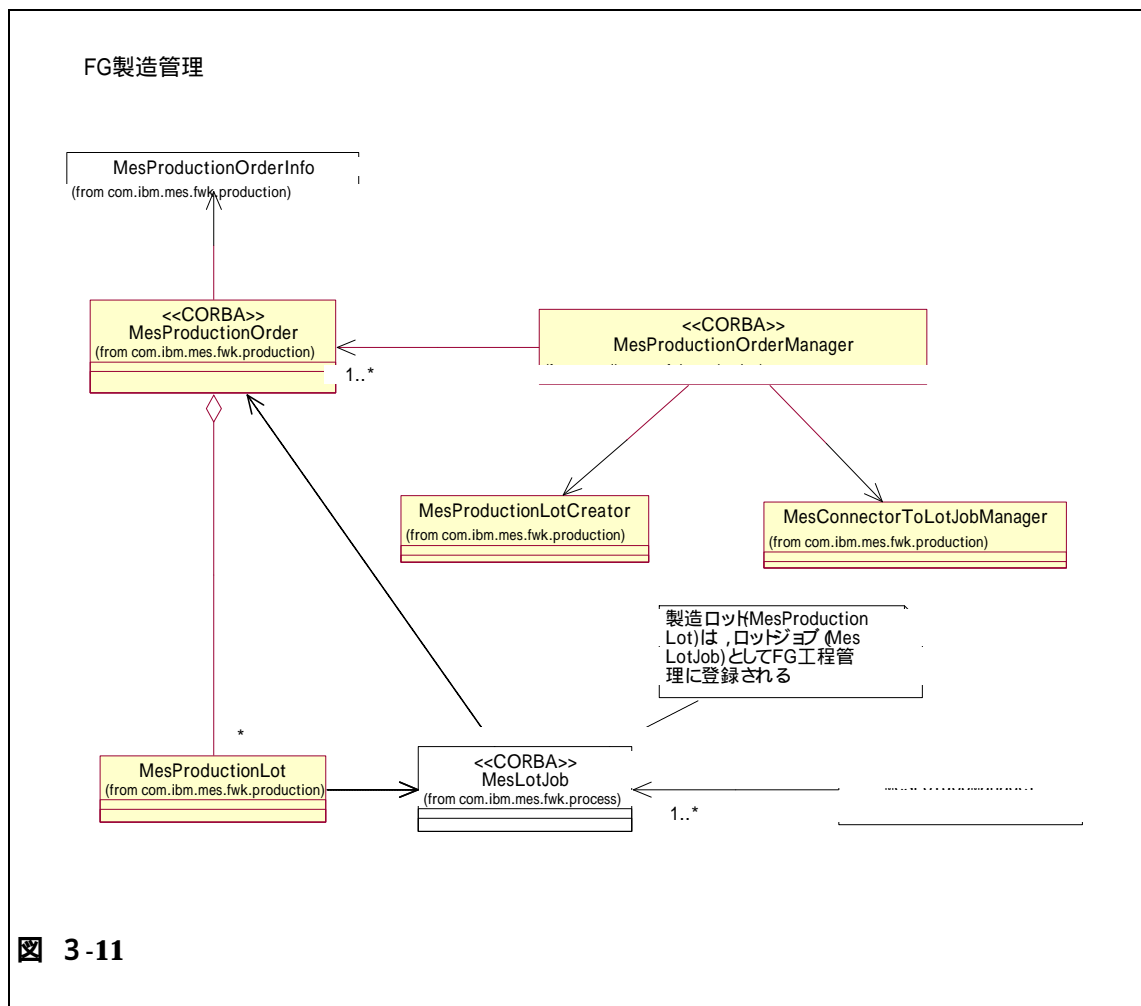


図 3-11

#### (1) MesProductionOrderManager

本インタフェースは MesProductionOrder を管理（登録、削除、検索、編集）するメソッドを定義する。

#### (2) MesProductionOrder

本インタフェースは製造指示を表す。以下の機能を持つメソッドが定義されている。

- 製造指示 ID の取得

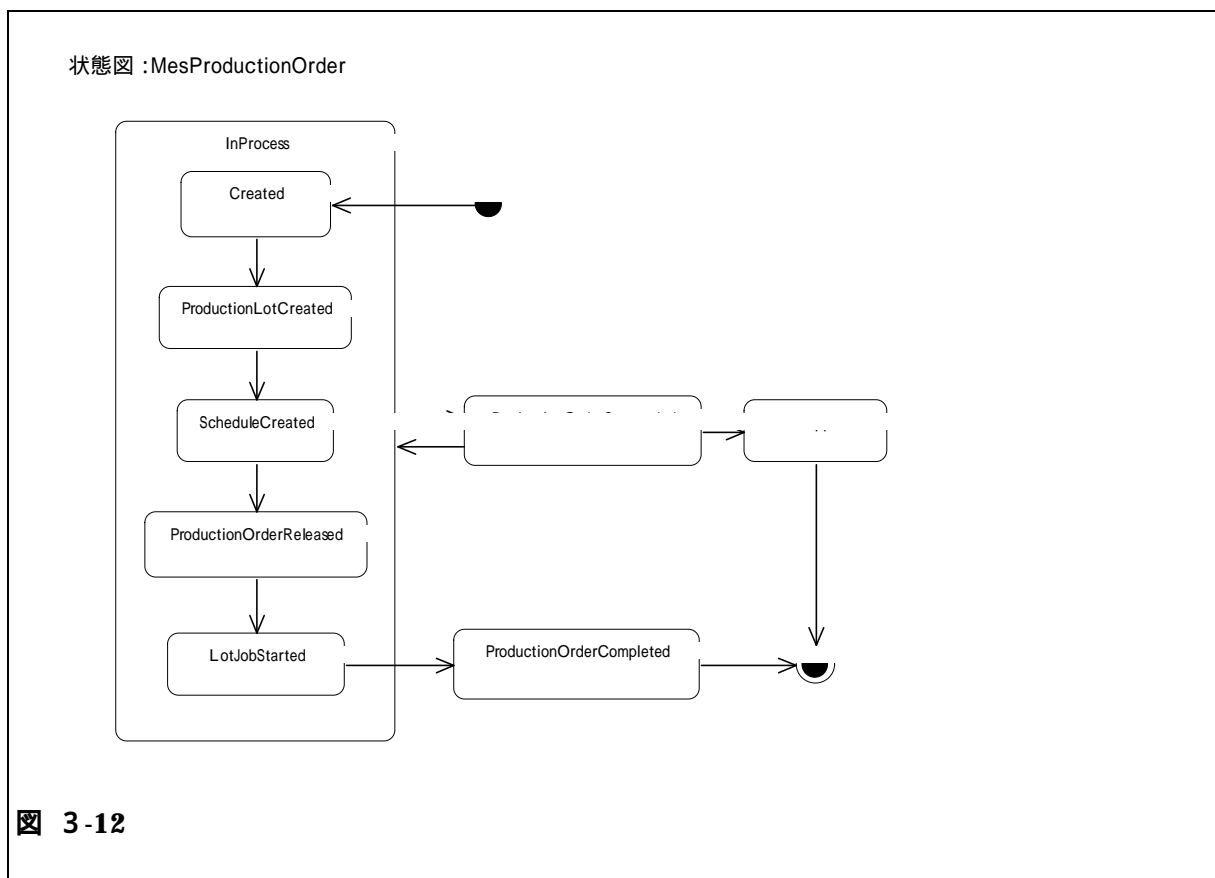
- 製品 ID の設定、取得
- 製品仕様 ID の取得
- 製品仕様バージョンの設定、取得
- 製造ロットの登録、取得
- 製造ロット数の取得
- 製造ロットの選択
- 製造指示優先順位の設定、取得
- 製造指示の状態の設定、取得
- 製造指示情報の設定、取得
- 現在の製造実績数量のカウント
- 製造予定数量の設定、取得
- 製造実績数量の設定、取得
- 着手予定日時の設定、取得
- 着手実績日時の設定、取得
- 最早製造指示投入時刻の設定、取得
- 最遅製造指示完了時刻の設定、取得
- 終了予定日時の設定、取得
- 終了実績日時の設定、取得
- 製造指示が登録された時刻の設定、取得。
- 製造指示状態集合の取得。
- すべての製造ロットの終了実績日時が設定されたかチェックする。
- すべての製造ロットの着手実績日時が設定されていないかチェックする。
- 全ての製造ロットの削除
- 特定の製造ロットの削除
- 製造指示を一時停止状態にする。
- 製造指示の一時停止状態を解除する。
- ProductionLot の納期設定。
- ProductionLot の着手予定日時の設定
- ProductionLot の着手実績日時の設定
- ProductionLot の終了予定日時の設定
- ProductionLot の終了実績日時の設定
- ProductionLot の実績数量の設定

### ( 3 ) MesProductionOrderState

本クラスは製造指示状態集合を表す。製造指示には以下の状態が存在する。

- 未処理

- 製造ロット生成済
- 製造指示投入済
- 小日程計画生成済
- 製造指示投入済
- ロットジョブ開始後
- 製造指示完了
- 製造指示一時停止
- 破棄



#### ( 4 ) MesProductionOrderInfo

本クラスは製造指示情報を表す。このクラスは適用事例において派生させることによって、MesProductionOrder では提供できない個別の属性やそれに関わる操作を定義するために用意されている。

### 3.2.2 製造ロット作成管理コンポーネント

#### ( 1 ) MesProductionLot

本クラスは製造ロットを表す。以下の機能を持つメソッドが定義されている。

- 製造ロット ID の設定、取得
- 製造指示 ID の設定、取得
- 納期の設定、取得
- 与えられた製造ロットがこの製造ロットと同じかどうか判定する。
- 予定数量の設定、取得
- 実績数量の設定、取得
- 着手予定日時の設定、取得
- 着手実績日時の設定、取得
- 終了予定日時の設定、取得
- 終了実績日時の設定、取得
- ロットジョブの優先順位を設定
- ロットジョブを一時停止状態に設定
- ロットジョブの再開

## ( 2 ) **MesProductionLotCreator**

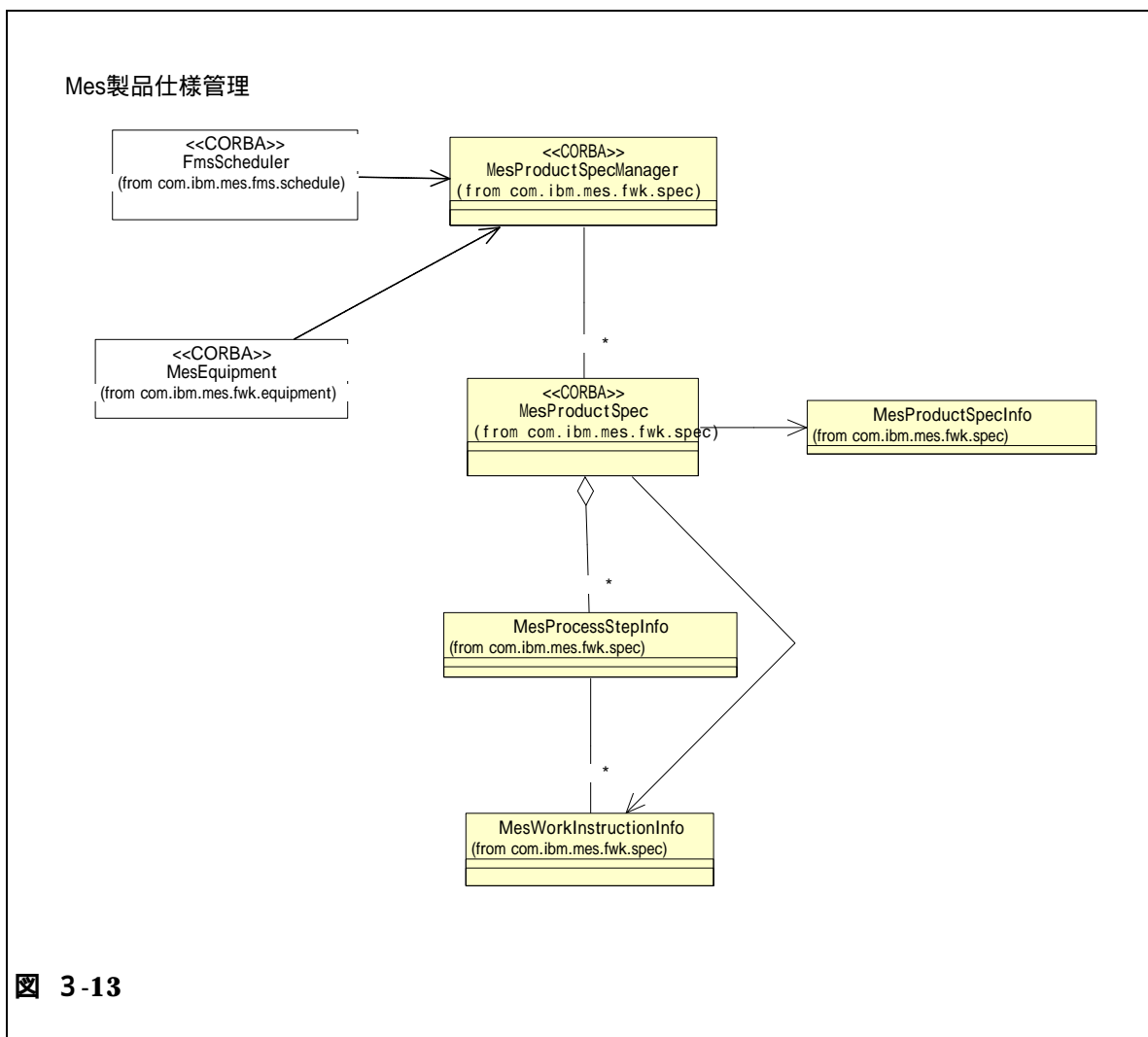
本クラスは MesProductionLot を生成する。

## ( 3 ) **MesConnectorToLotJobManager**

本クラスは製造指示管理機能グループと工程管理機能グループとのインタフェース機能を持つ。MesProductionLot を MesLotJobManager に送付する。

### 3.3 製品仕様管理機能グループ

#### 3.3.1 製品仕様管理コンポーネント



#### (1) MesProductSpecManager

本インタフェースは製品仕様 (MesProductSpec) に対する操作 (登録、削除、検索、変更) を行なうメソッドを定義する。

#### (2) MesProductSpec

本インタフェースは製品仕様 (MesProductSpecInfo) と工程ステップ情報 (MesProcessStepInfo) を管理する次のメソッドを定義する。

- 製品 ID の設定、取得。
- 製品仕様名の設定、取得
- MesProcessStepInfo の登録、削除、検索
- MesWorkInstructionInfo の登録、削除

- MesProductSpecInfo の設定、取得
- 仕様の参照回数を 1 増減する。参照回数が 0 のときのみ当該 MesProductSpec オブジェクトを削除できる。
- 製品仕様の参照回数設定、取得
- この製品仕様が使用中であるかどうかチェックする。
- 工程経路の設定、取得
- 製品バージョンの設定、取得

### ( 3 ) MesProcessStepInfo

本クラスは工程内作業指図の登録と工程ステップ属性の変更をする。

- MesWorkInstructionInfo 工程内作業指図の登録、取得、削除
- 関連先 MesProductSpecInfo オブジェクト・リファレンスを設定、取得する。
- 工程ステップ番号の設定、取得
- 工程ステップ名の設定、取得

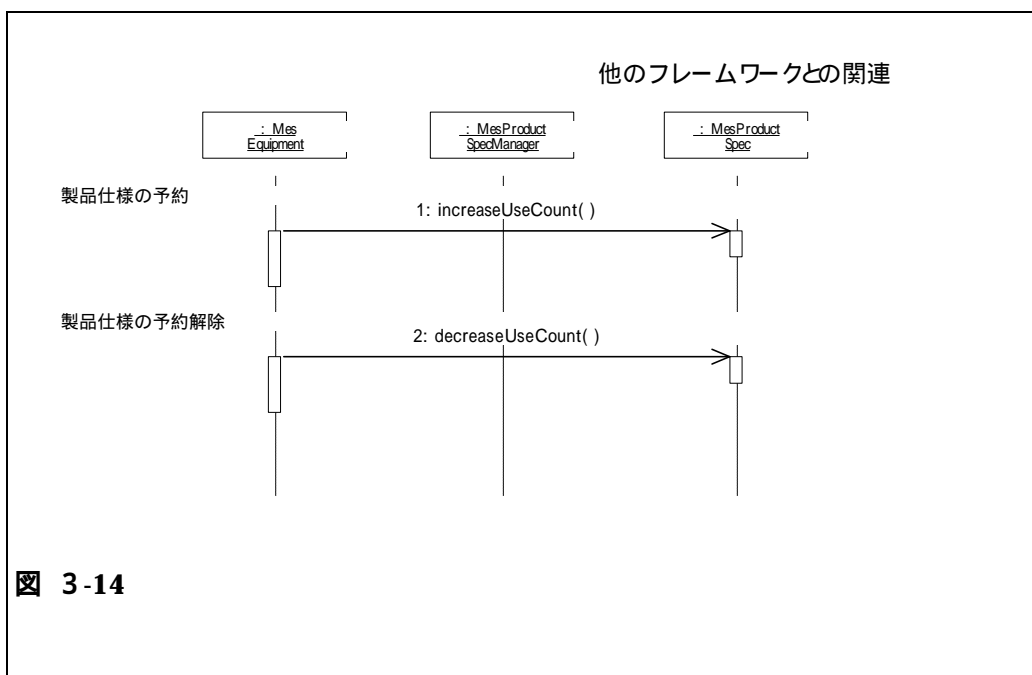
### ( 4 ) MesWorkInstructionInfo

本クラスは工程内作業指図を表す。MES ごとにカスタマイズすることを想定している。

- 製造設備 ID の設定、取得
- 製品設備型番の設定、取得
- 関連 ProcessStepInfo オブジェクトの設定、取得
- 設備モードの設定、取得

### ( 5 ) MesProductSpecInfo

本クラスは製品仕様の追加情報を表しており、MES ごとにカスタマイズすることを想定している。



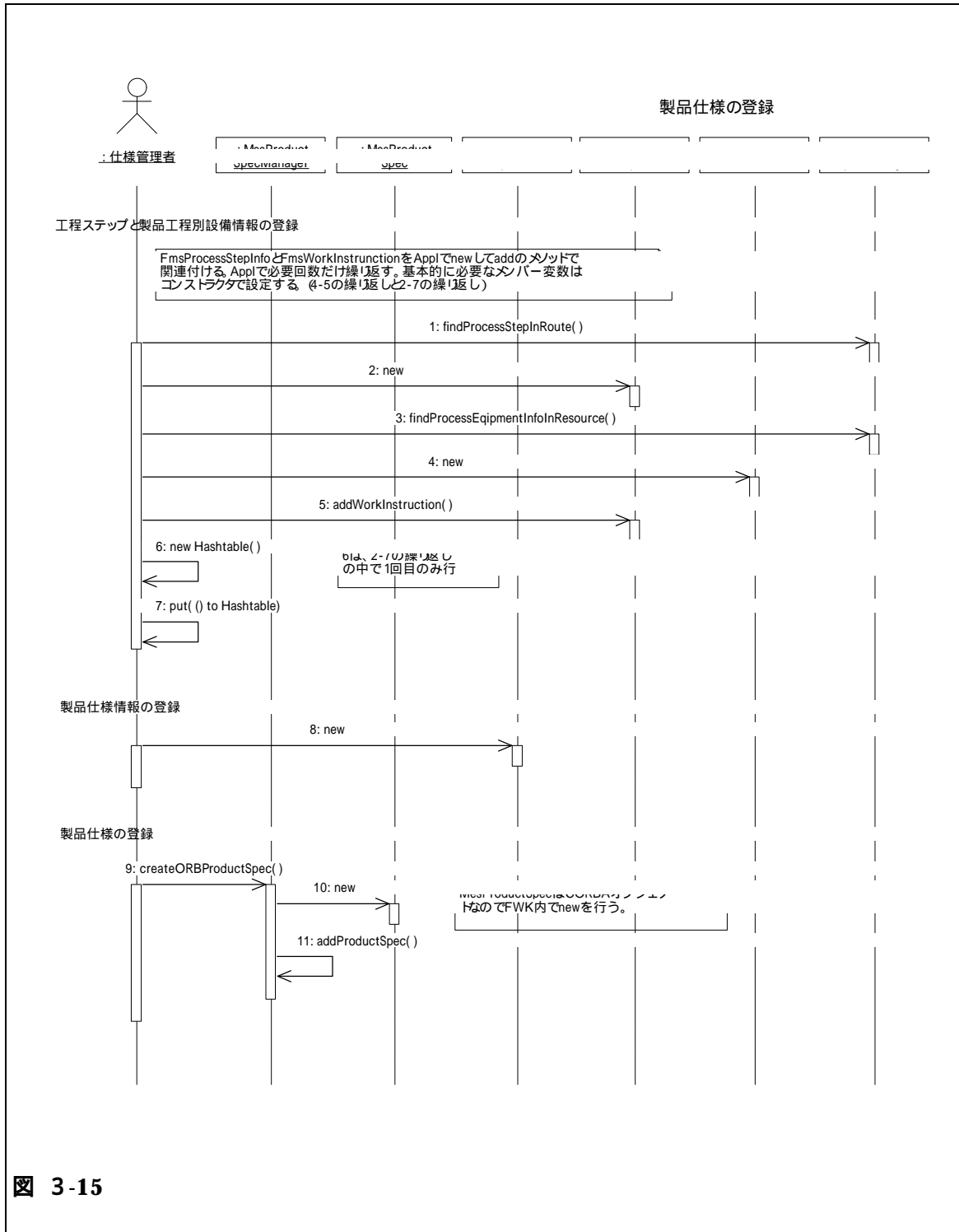


図 3-15

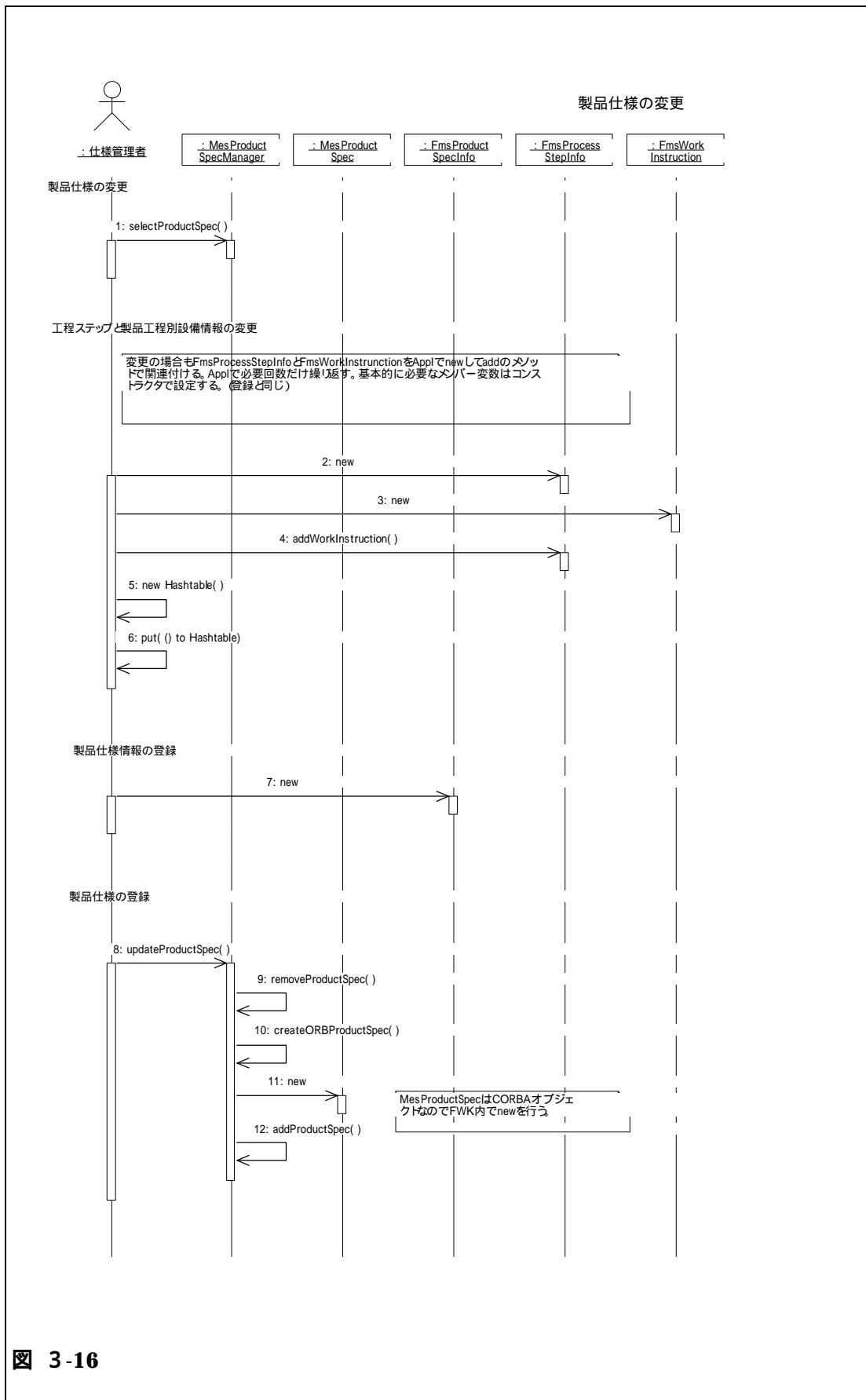


図 3-16

### 3.4 工程管理機能グループ

工程管理機能グループは、ロットに対する製造工程の実行すなわち設備に対し作業指示を行ない、作業進捗を追跡する。工程管理の中心になるのは、ロットに対する作業計画と作業実績を管理する LotJob と呼ばれるクラスと、作業計画に基づいて作業指示を各設備に差し立てる Dispatcher と呼ばれるクラスである。

ここでロットと呼ばれるのは、工程内を流れる物理的なロットと対応する製造ロットである。今回は、投入から完了までロットの統合や分割は行わないものとしてモデル化した。また、製造指示との紐付けは、製造指示管理機能グループで取り扱うものとした。

次のコンポーネントから構成される。

- 製造ロット管理コンポーネント
- 工程内作業指示管理コンポーネント
- 投入指示管理コンポーネント
- 搬送指示管理コンポーネント

#### 3.4.1 製造ロット管理コンポーネント

##### (1) MesLotJobServer

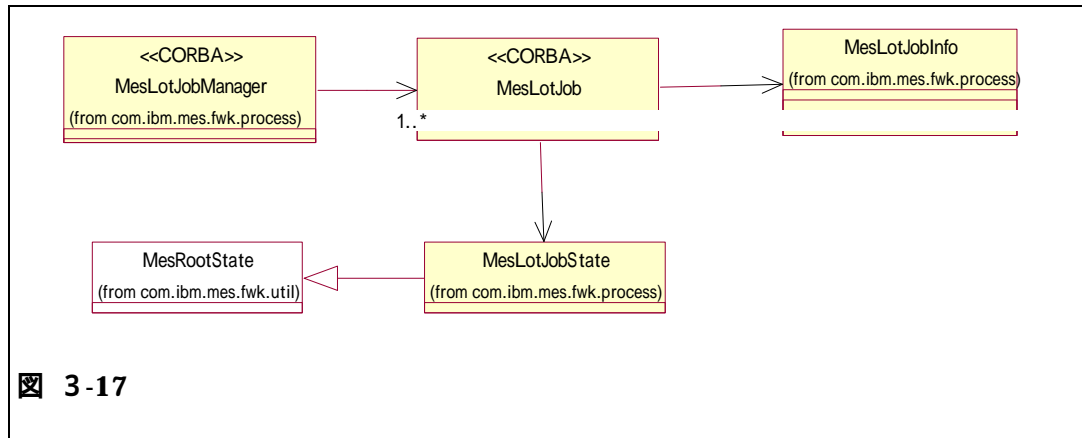


図 3-17

本クラスは CORBA オブジェクトである MesLotJob を実装したオブジェクトを立ち上げ、管理する。

##### (2) MesLotJobManagerServer

本クラスは CORBA オブジェクトである MesLotJobManager を実装したオブジェクトを立ち上げ、管理する。

##### (3) MesLotJobCreator

本クラスはロットジョブを生成する。Creator に関しては共通機能グループを参照していただきたい。

#### ( 4 ) **MesLotJobHandlerServer**

本クラスは CORBA オブジェクトである MesLotJobHandler を実装したオブジェクトを立ち上げ、管理する。

#### ( 5 ) **MesLotJobHandler**

このインタフェースは MesLotJob を異なるアドレス空間に配置するために用意されている。本仕様書では詳述しない。

#### ( 6 ) **MesLotJobManager**

MesLotJob を管理するインタフェースであり、以下の機能を提供するメソッドが宣言されている。

- ロットの作成、削除、検索
- プロセスジョブの検索
- 全ロットジョブの投入 ( 実行されるのはディスパッチ以後 )
- 特定のプロセスジョブの投入
- 全プロセスジョブのディスパッチ
- 優先順位の高いプロセスジョブのディスパッチ

#### ( 7 ) **MesLotJob**

ロットジョブを表現するインタフェースであり、以下の機能を提供するメソッドが宣言されている。

- プロセスジョブの登録、検索
- 優先順位の設定・取得
- 開始予定時刻、終了予定時刻、開始実績時刻、終了実績時刻の設定・取得
- 最早投入時刻、最遅完了時刻の設定・取得
- 製造予定数量の設定・取得
- 製造実績数量の取得
- 実績欠陥数量の取得
- ロットジョブ情報の設定・取得
- 状態の設定・取得
- 実行開始要求
- 一時停止、再開通知
- プロセスジョブ完了通知
- プロセスジョブ中断通知
- プロセスジョブ取り消し通知
- プロセスジョブ一時停止通知
- プロセスジョブ再開通知

変更内容が MesWorkOrder(作業指示)まで伝達されるものとして以下のものがある。

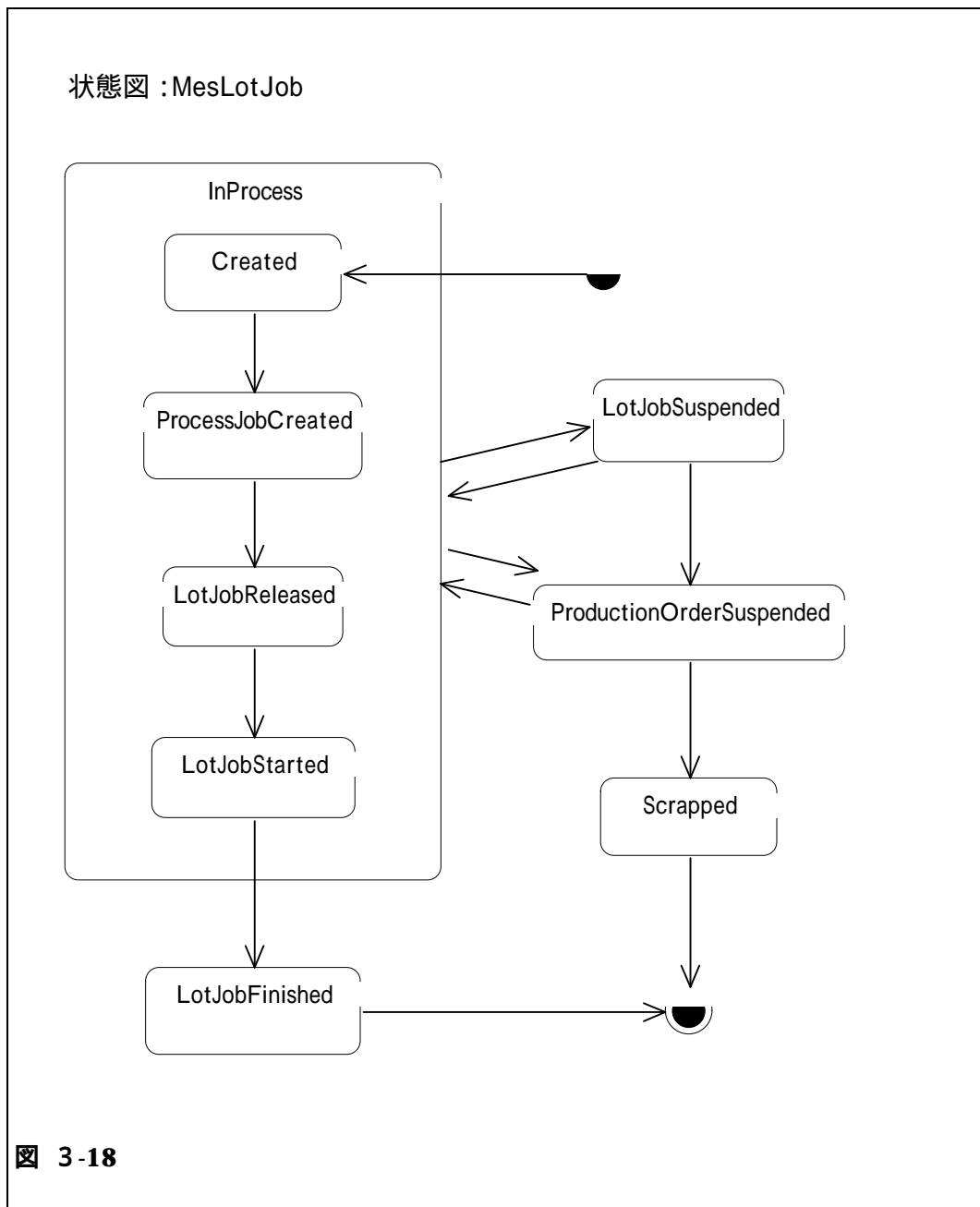
- 本オブジェクトを構成している全プロセスジョブの優先順位
- 本オブジェクトを構成している特定のプロセスジョブの優先順位
- 本オブジェクトを構成している特定のプロセスジョブの開始予定時刻

以下のものについては実行時に履歴情報をデータベースに記録することができる。

- 実行開始要求
- プロセスジョブ完了通知
- プロセスジョブ中断通知
- プロセスジョブ取り消し通知
- プロセスジョブ一時停止通知
- プロセスジョブ再開通知

#### ( 8 ) **MesLotJobState**

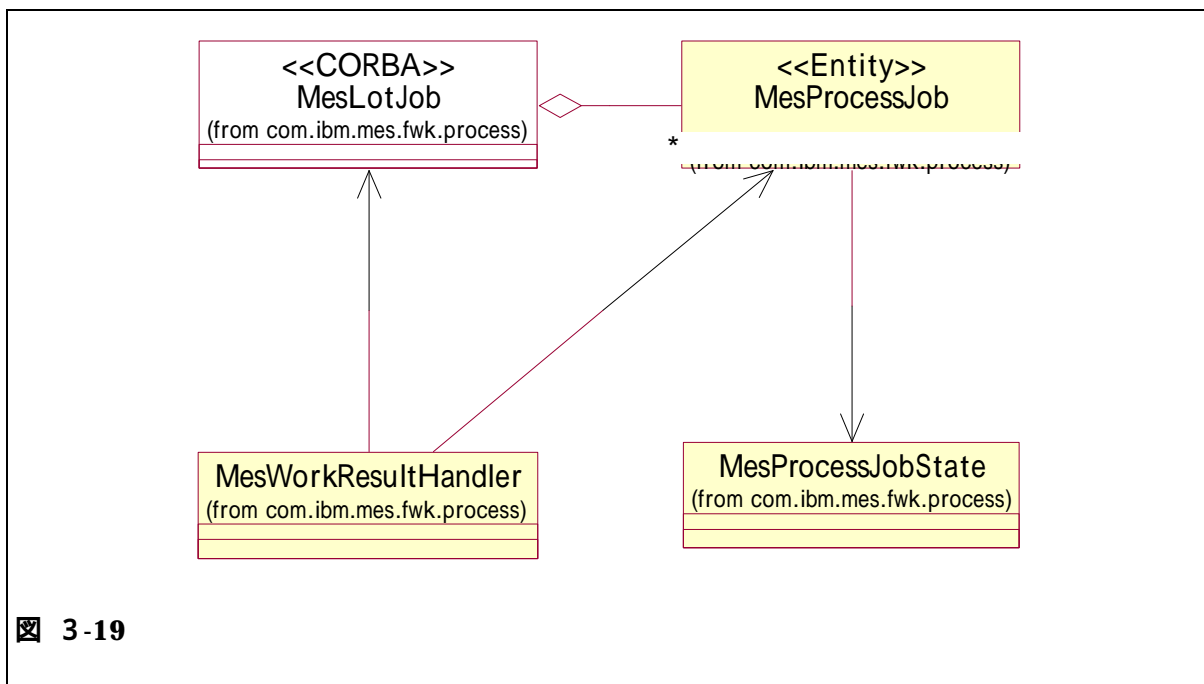
本クラスはロットジョブの状態を表現する。



( 9 ) MesLotJobInfo

このクラスは適用事例において派生させることによって、MesLotJob では提供できない個別の属性やそれに関わる操作を定義するために用意されている。アプリケーション開発者は CORBA オブジェクトである MesLotJob をカスタマイズする必要がない。

### 3.4.2 工程内作業管理コンポーネント



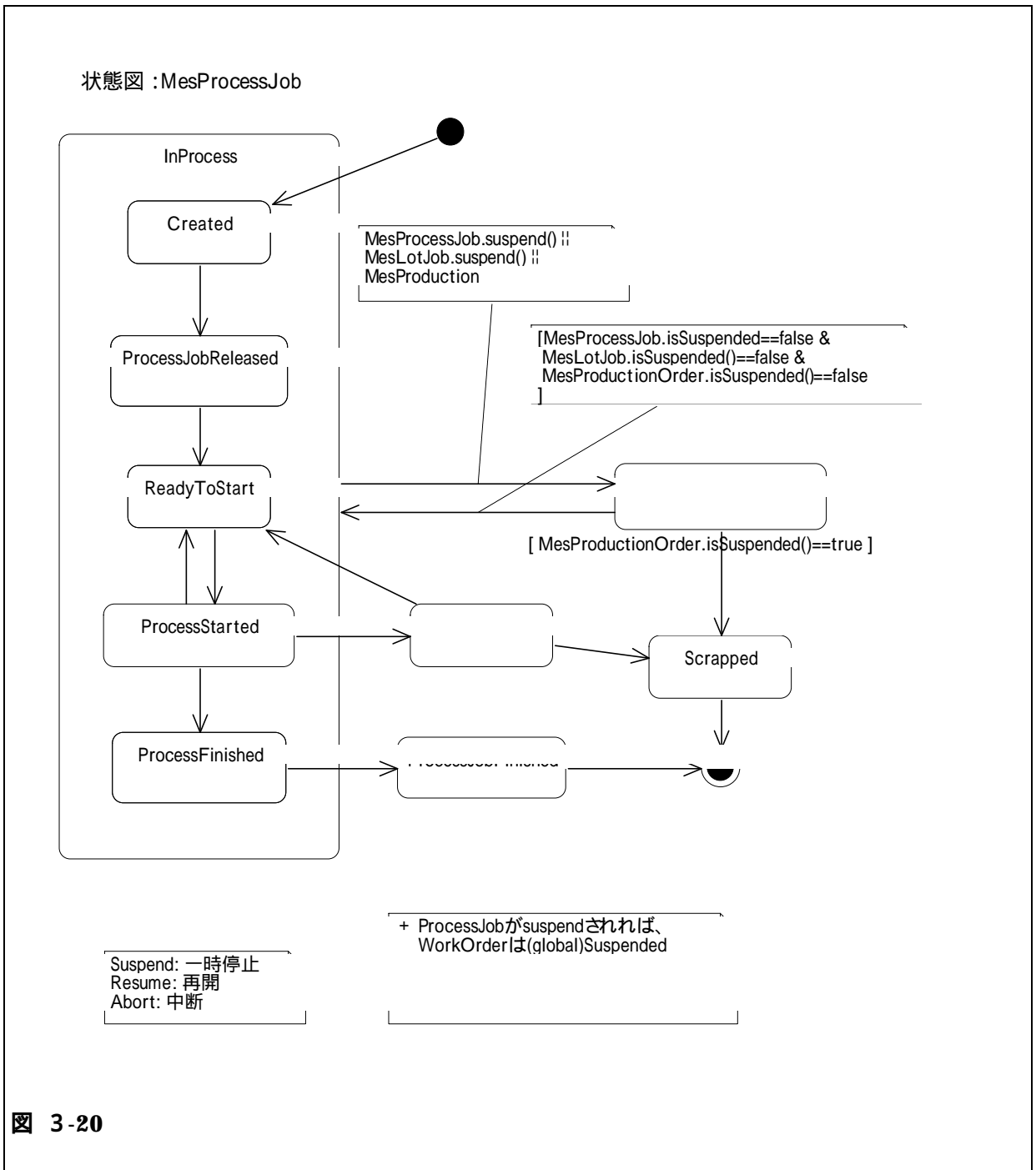


図 3-20

( 1 ) MesProcessJob

本クラスはプロセスジョブを表現する。以下の機能を持つ。

- 作業予定数量の設定、取得
- 作業実績数量の設定、取得
- 作業予定設備の設定、取得

- 作業実績設備の設定・取得
- 優先順位の設定、取得
- 工程資源の設定、取得
- 作業開始予定時刻の設定、取得
- 作業終了予定時刻の設定、取得
- 作業開始実績時刻の設定、取得
- 作業終了実績時刻の設定、取得
- 作業実績データの設定
- 実績欠陥数量の設定、取得
- 作業取り消し時刻の設定、取得
- 状態の設定、取得
- 投入要求
- 開始要求
- 中断要求
- 取り消し要求
- 終了要求
- 一時停止通知
- 再開通知

## ( 2 ) **MesProcessJobState**

本クラスはプロセスジョブの状態を表現する。

## ( 3 ) **MesWorkResultHandler**

本クラスは MesProductionOrder に、関連を辿って次のものを通知する。

- 最初の MesProcessJob の開始
- 最後の MesProcessJob の完了
- MesWorkResult ( 作業開始と作業完了と両方ある )

### 3.4.3 投入指示管理コンポーネント

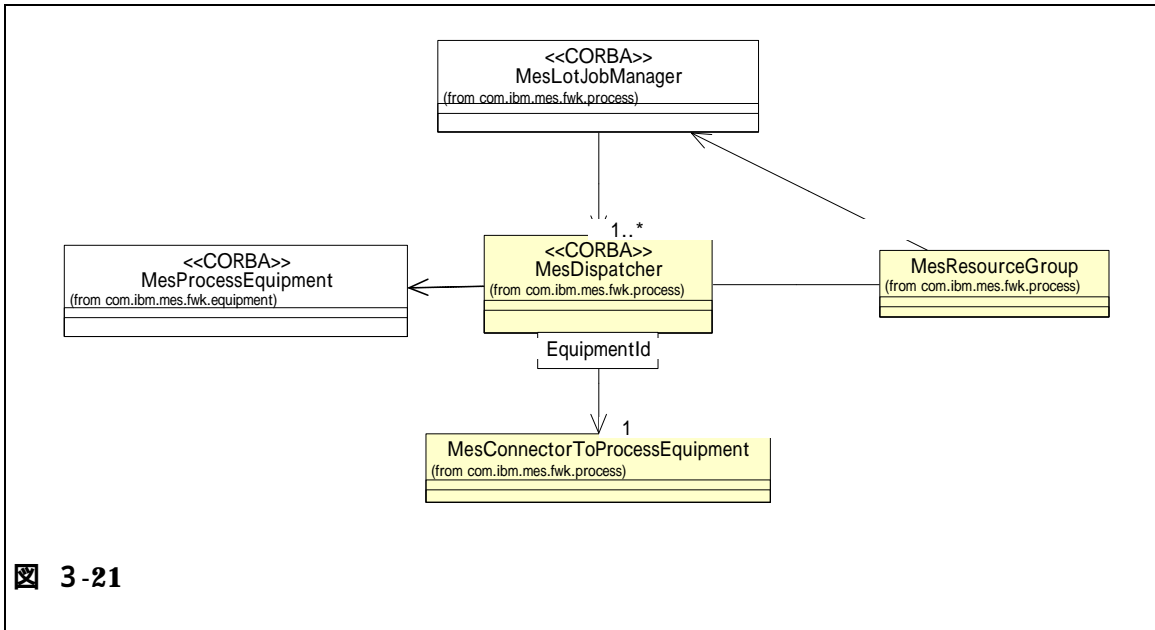


図 3-21

#### (1) MesDispatcherServer

本クラスはCORBA オブジェクトである MesDispatcher を実装したオブジェクトを管理する。

#### (2) MesDispatcher

本インタフェースは MesWorkOrder を MesProcessEquipment に差し立てるメソッドを定義する。

- ディスパッチングに関する履歴情報を登録する。
- プロセスジョブを工程設備コネクタに登録する。
- リリースされた全プロセスジョブをディスパッチする。
- リリースされた全プロセスジョブを指定された工程設備にディスパッチする。
- ディスパッチング順位の最も高いものから指定された数のプロセスジョブを工程設備にディスパッチする。
- ディスパッチング順位の最も高いものから指定された数のプロセスジョブを特定の工程設備にディスパッチする。
- 指定された工程設備に登録されたプロセスジョブを検索する。
- ロットジョブ・マネジャーID を返す。
- プロセスジョブの比較演算子を返す。
- 指定された工程設備コネクタに登録されているかどうかを返答する。
- 代替設備に該当するコネクタに登録されたプロセスジョブ状態を開始不可とする。
- プロセスジョブが削除されたことを工程設備コネクタに通知する。
- 代替設備に該当するコネクタに登録されたプロセスジョブを削除する。

- 工程設備コネクタ・クラス名を設定する。
- ロットジョブ・マネジャーID を設定する。
- プロセスジョブの比較演算子を設定する。
- プロセスジョブのパラメタが更新されたことを工程設備コネクタに通知する。

### ( 3 ) MesConnectorToProcessEquipment

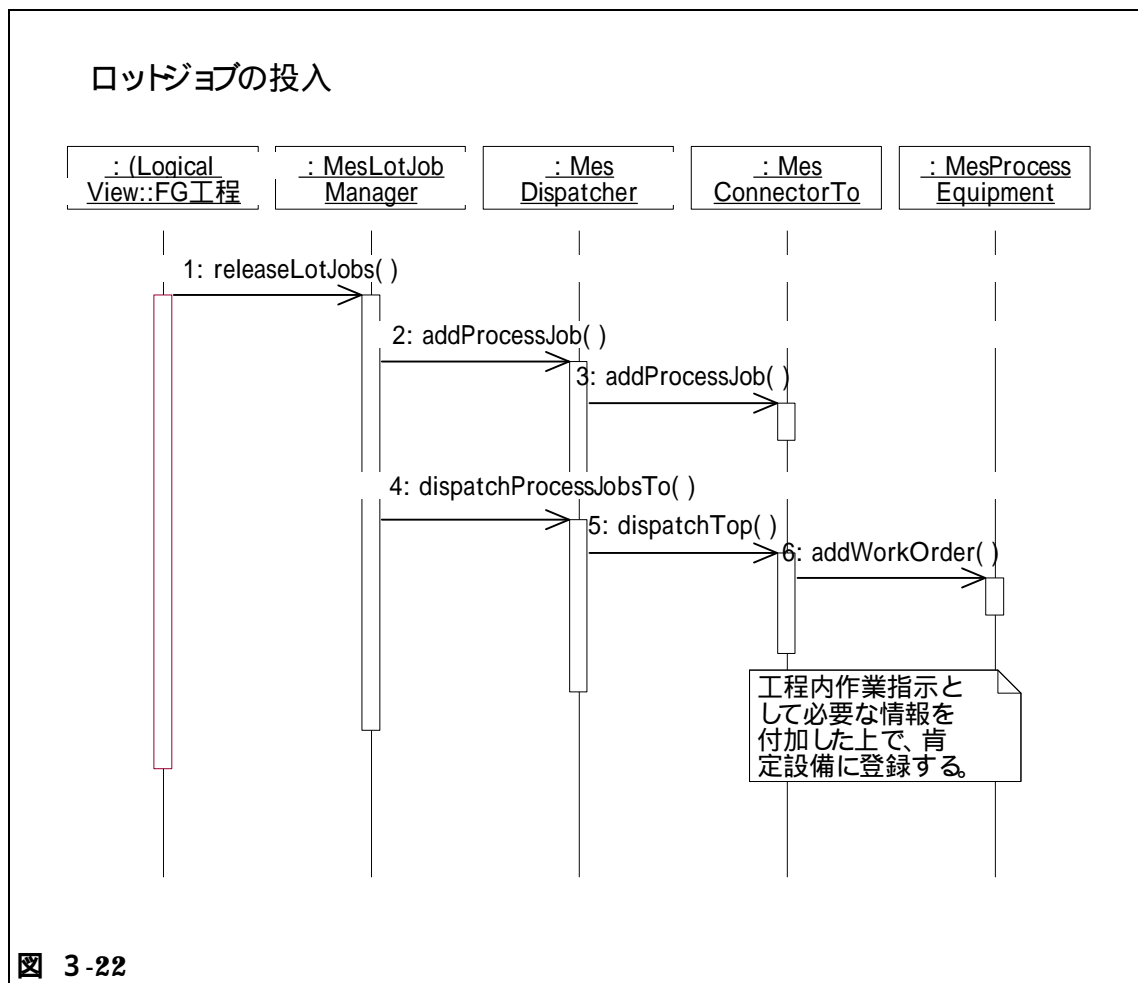
本クラスは工程管理機能グループと設備管理機能グループとのインタフェースを取る。この両者の接続は特に MES でカスタマイズする必要性が高く、機能グループ間での依存関係を単純化するために用意されている。

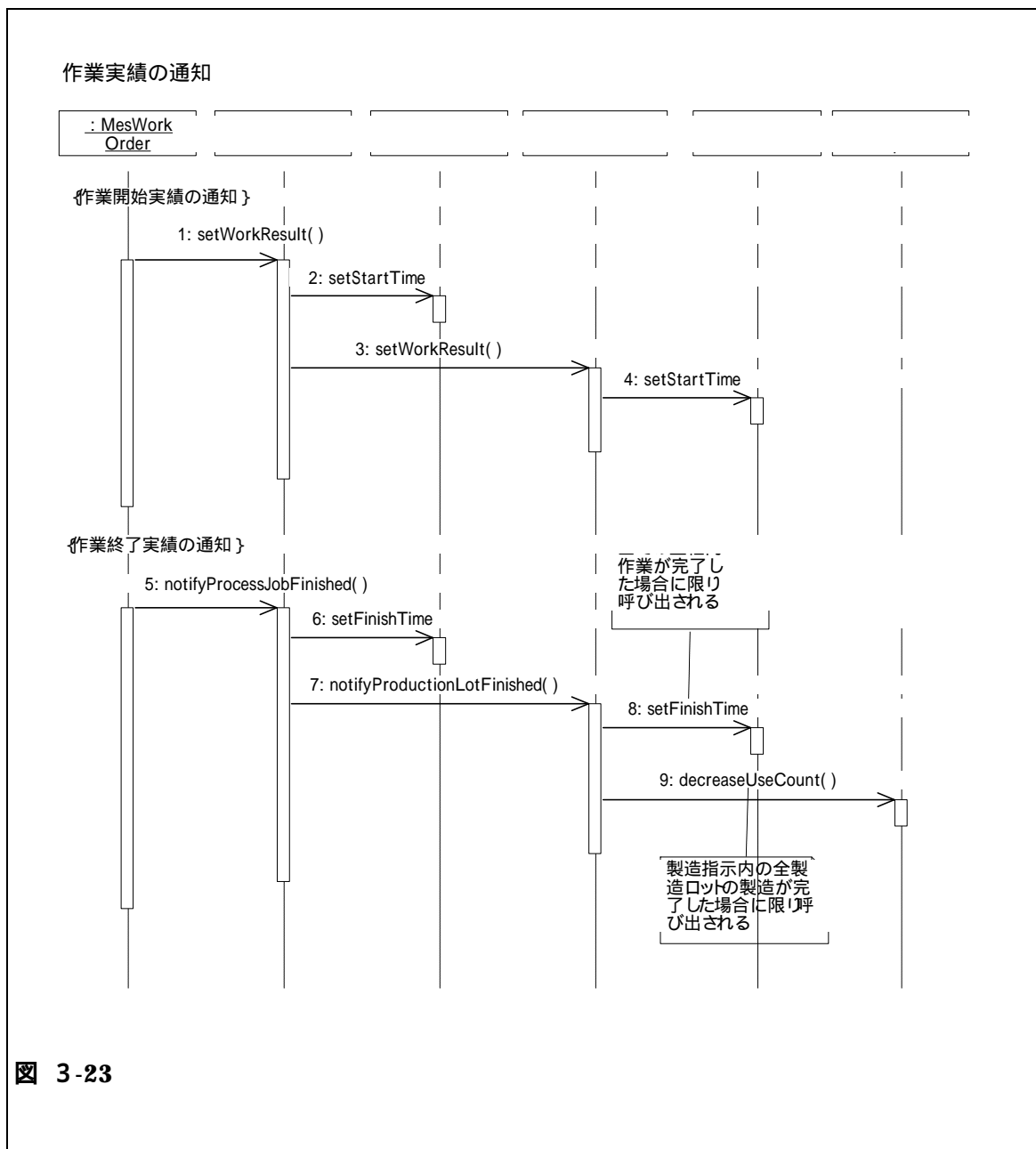
カスタマイズする必要性があるのは、次ぎの 2 つである。

- 作業指示 ( MesWorkOrder ) の工程設備への登録・削除
- プロセスジョブに対する優先順位、作業開始予定時刻、一時停止状態の設定・解除について、作業指示への反映

### ( 4 ) MesResourceGroup

代替設備を表す。





### 3.4.4 搬送指示管理コンポーネント

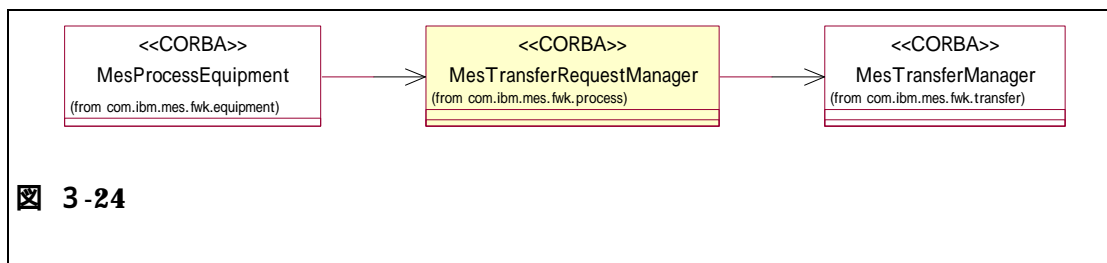
本コンポーネントは搬送指示を管理する機能を提供する。

#### (1) MesTransferRequestManager

本インターフェースは、ロットを設備に対して配送する要求または設備からの引き取りの要求を受け付け、実行可能になったら、例えば対象設備の作業が完了したら、搬送設備に

対し搬送を要求する以下の機能を提供するメソッドを定義している。

- ロットを設備に配送するように要求する。
- セットアップのために、空パレットの搬入を要求する。
- 空になったパレットの搬出を要求する。



- ロットを設備から引き取るように要求する。
- 搬送完了の通知を受け取る。

### 3.5 資材管理機能グループ

資材管理機能グループは、資材の利用実績を管理する。Consumable クラスと呼ぶ消費財と Durable クラスと呼ぶ耐久財を管理対象として定義している。ただし、資材の在庫管理全体まではカバーしておらず、例えば資材の種別ごとのマスタ情報などは持っていない。これは、資材の在庫管理は ERP などの外部システムが行うものと仮定しており、ここで用意したクラスは、外部システムに対して利用状況を通知するためのインタフェースとして使われることを想定しているためである。

#### 3.5.1 資材管理コンポーネント

##### (1) MesInventoryManagerServer

本クラスは MesInventoryManager を実装したオブジェクトを立ち上げ、管理する。

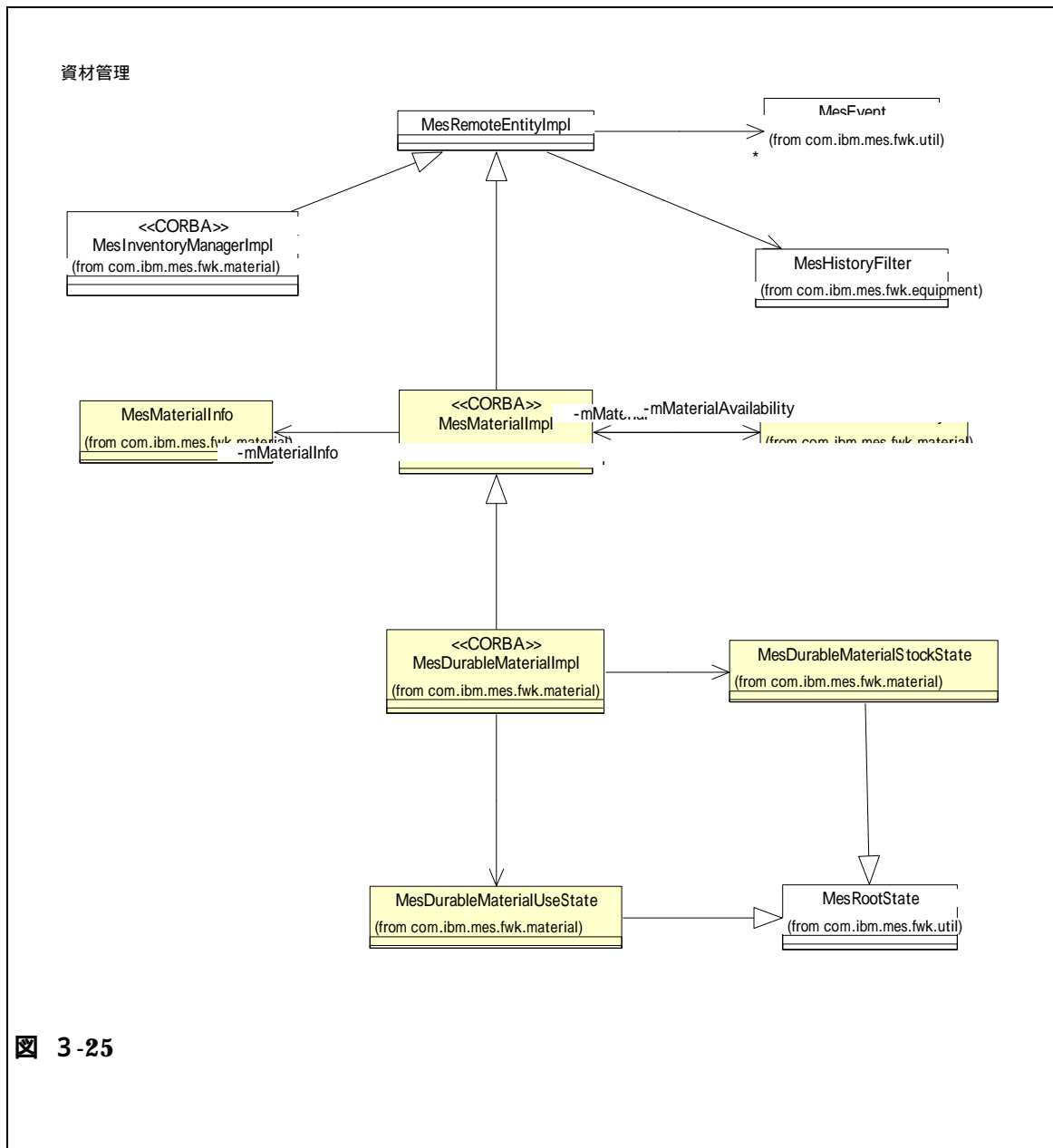


図 3-25

## ( 2 ) **MesInventoryManager**

このインタフェースは、資材 ( MesMaterial ) を管理 ( 登録、検索、削除 ) するメソッドを定義する。

## ( 3 ) **MesMaterial**

このインタフェースは、資材を定義する。

- 所在の設定、取得
- MesMaterialInfo の設定、取得
- 資材名の設定、取得
- 資材タイプの取得
- 寿命に達したかどうかを調べる。

## ( 4 ) **MesMaterialAvailability**

この抽象クラスは、資材の寿命を調べる機能を定義する。

## ( 5 ) **MesMaterialInfo**

この抽象クラスは、資材情報を定義する。

## ( 6 ) **MesDurableMaterial**

このインターフェースは、資材から継承した耐久資材の機能を定義する。

- 使用回数の実績を設定、取得する。
- 使用時間の実績を設定、取得する。
- 残りの可能使用回数を取得する。
- 残りの可能使用時間を取得する。
- MesDurableMaterialStockState を設定、取得する。
- 使用回数の上限値を設定、取得する。
- 使用時間の上限値を設定、取得する。
- MesDurableMaterialUseState を設定、取得する。
- 実績使用回数を 1 回加算する。
- 使用時間の実績を加算する。

## ( 7 ) **MesDurableMaterialUseState**

このクラスは、耐久資材の使用状態集合を定義する。

- 使用可 (Ready)
- 使用中 (BeingUsed)
- 予約中 (BeingReserved)
- 使用不可 (NotUsable)
- 廃棄 (Disposed)

## ( 8 ) **MesDurableMaterialStockState**

このクラスは、耐久資材のストック状態集合を定義する。

- 保管中 (InStock)

- 出庫中 (BeingStockOut)
- 設備内 (InEquipment)
- 入庫中 (BeingStockIn)
- 廃棄 (Disposed)

#### ( 9 ) **MesDurableMaterialAvaibility**

このクラスは、耐久資材の寿命を調べる機能を定義する。寿命判定のロジックを実装する必要がある。

### 3.6 搬送管理機能グループ

搬送管理機能グループは、ロットを搬送する機能を提供する。

ただし搬送システムは、通常専門の搬送装置メーカーによって一括してインストールされることが多い。従って、搬送システム内の AGV や自動倉庫の標準モデルが存在し得るかどうかは不明である。このためオープン MES では、搬送システムのインタフェースの抽象度に段階を設け、最も高い抽象度では、搬送システム全体をブラック・ボックスとして扱えるように考慮した。

#### 3.6.1 搬送管理コンポーネント

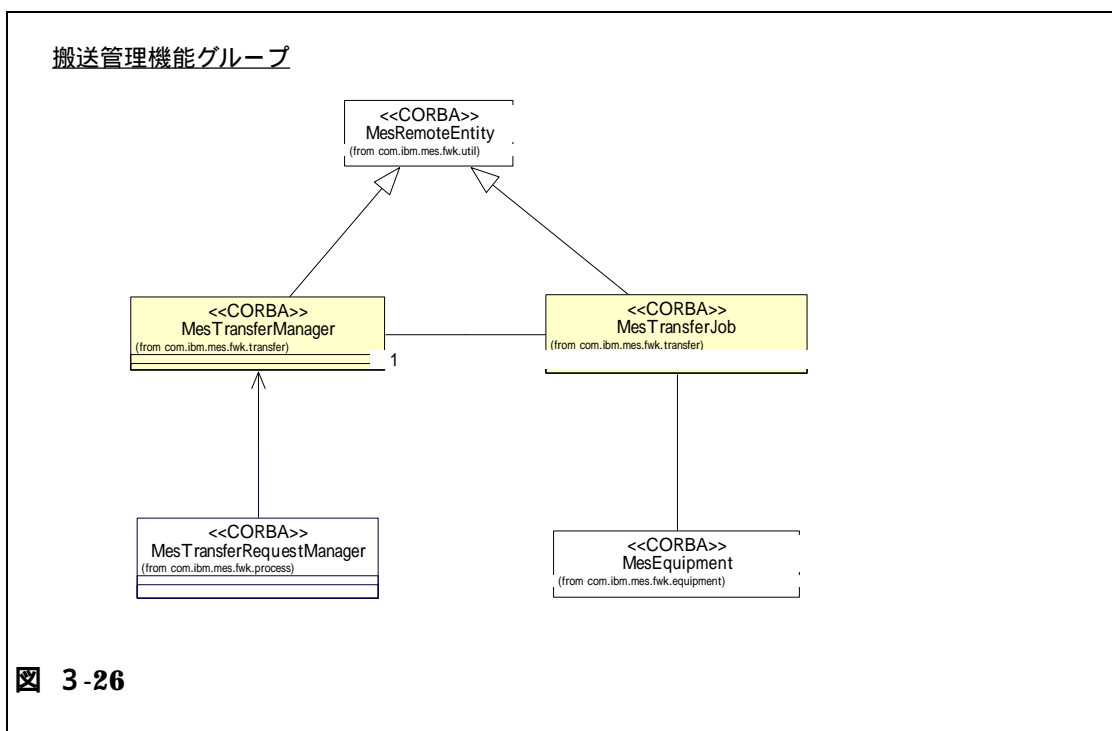


図 3-26

#### (1) MesTransferManager

このインターフェースは、搬送管理機能を定義する。

- MesTransferRequestManager を設定、取得する。
- 貯蔵庫から設備へロットの配送を行う。
- 設備から貯蔵庫へロットの配送を行う。
- ロットの設備間搬送を行う。
- 実行中の全ての搬送ジョブを取得する。
- 搬送が完了した通知を受け取る。
- 実行中の搬送ジョブの個数を取得する。
- 搬送ジョブを投入する。
- MesTransferJob オブジェクトを取得する。

## ( 2 ) MesTransferJob

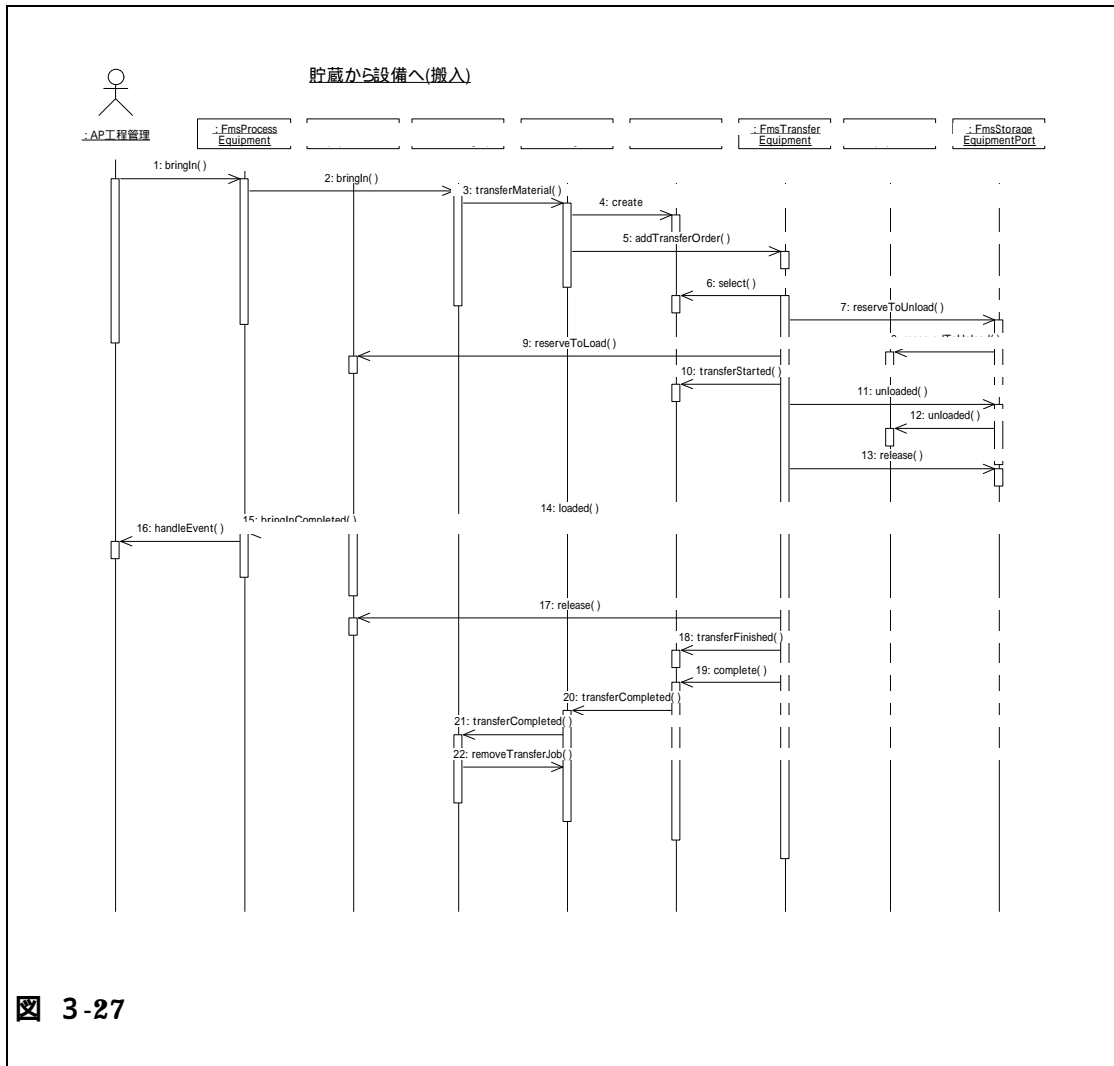
このインターフェースは、搬送ジョブの機能を定義する。

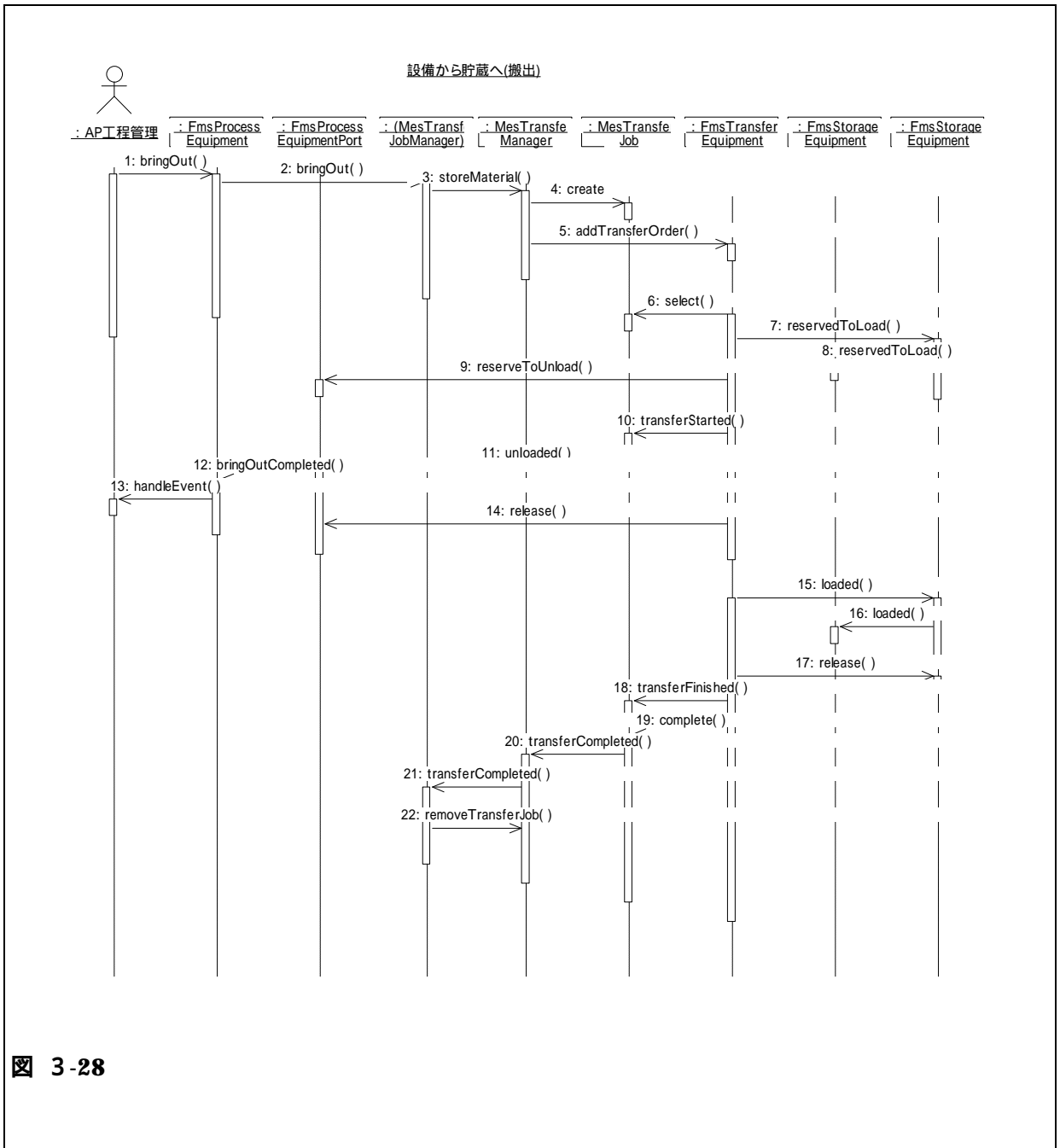
- 搬送元設備を取得する。
- 搬送先設備を取得する。
- ロットを取得する。
- 優先順位を設定、取得する。
- 搬送ジョブを要求した日時を取得する。
- 搬送ジョブを開始した日時を取得する。
- 搬送ジョブを中断する。
- 搬送ジョブを取り消す。
- 搬送ジョブを完了する。
- 搬送ジョブを完了した日時を取得する。
- 搬送ジョブの状態を取得する。
- 搬送ジョブの終了通知を受け取る。
- 搬送ジョブの開始通知を受け取る。
- 搬送ジョブを選択する。
- 搬送ジョブの選択を解除する。

## ( 3 ) MesTransferJobState

本クラスは搬送ジョブの状態を表す。搬送ジョブには以下の状態が存在する。

- 搬送可能 (ReadyForTransfer)
- 搬送待ち (WaitingForTransfer)
- 搬送中 (UnderTransfer)
- 搬送アボート (TransferAborted)
- 搬送中止 (TransferCanceled)
- 搬送完了 (TransferCompleted)





### 3.7 工程仕様管理機能グループ

本機能グループは工程経路、工程資源を管理する機能を提供する。MesProcessRoute(工程経路)は MesProcessStep(工程ステップ)のシーケンスであり、この情報をもとに MesLotJob から MesProcessJob が生成される。

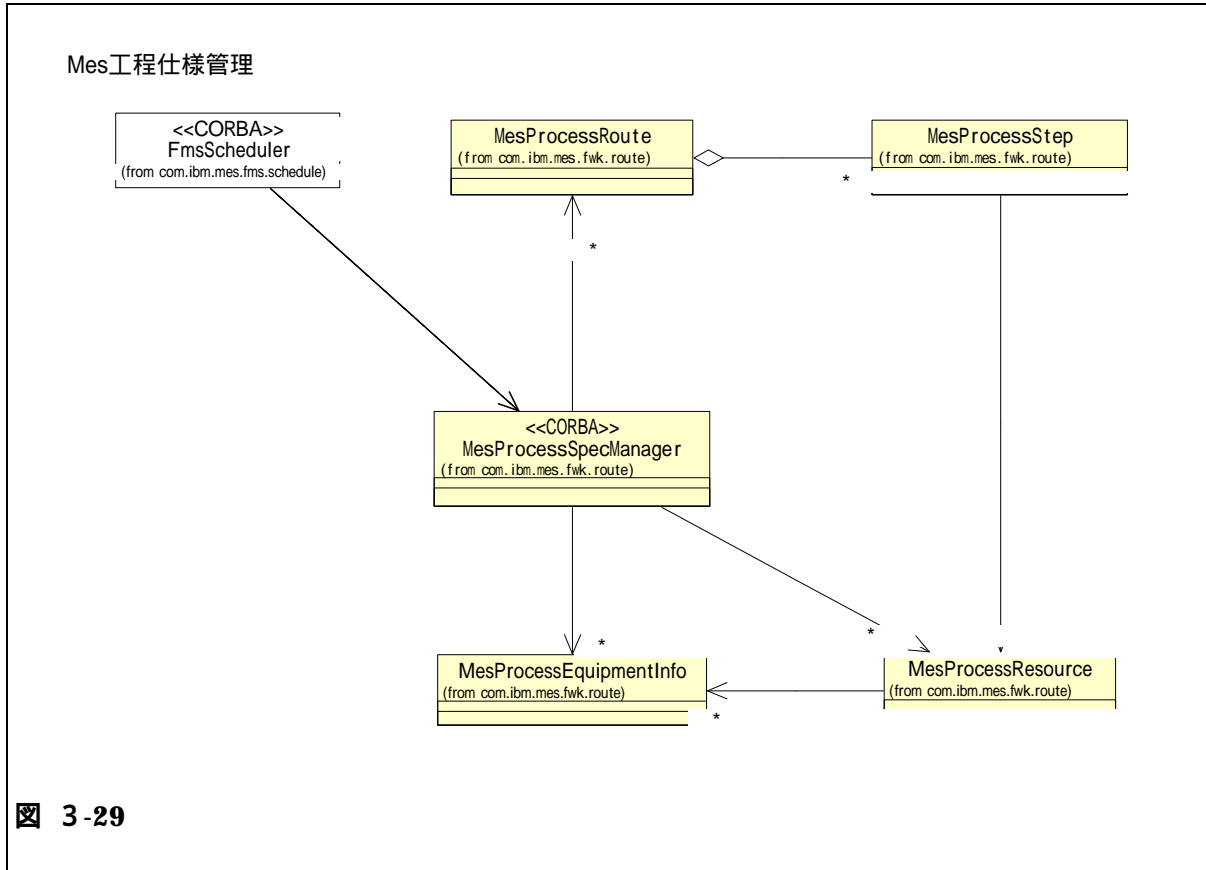


図 3-29

#### 3.7.1 工程経路管理コンポーネント

##### (1) MesProcessSpecManager

本インタフェースは工程経路、工程ステップ、工程資源および製造設備のオブジェクトに対し操作するメソッドを定義する。

- MesProcessResource(工程資源)を登録、検索、削除する。
- MesProcessRoute(工程経路)を登録、検索、削除する。
- MesProcessStep(工程ステップ)を工程経路に登録する。
- 工程経路内の工程ステップを検索する。
- 工程経路の参照カウントを1つ増減する。
- MesProcessEquipmentInfo(工程設備情報)を設定、検索、変更、削除する。
- 工程設備情報を工程資源に関連づける。
- 工程資源に登録されている工程設備情報を検索、削除する。

- 工程経路に対する参照が存在するかどうかチェックする。
- 工程ステップに工程資源を割り当てる。
- 工程ステップから工程資源の割り当てを削除する。
- 工程ステップを工程経路から削除する。

#### ( 2 ) MesProcessRoute

本クラスは工程経路を表す。

- 工程ステップを工程経路に登録する。
- 工程経路の工程ステップを削除する
- 工程ステップを取得する
- 参照カウンタを 1 増減する。
- 参照カウンタを設定、取得する。
- 使用中かどうかを返す。

#### ( 3 ) MesProcessStep

本クラスは工程ステップを表す。

- 関連先 MesProcessRoute を設定、取得する。
- 関連工程資源を設定、取得する。
- 工程資源を工程ステップに登録する。
- 工程ステップの工程資源を削除する

#### ( 4 ) MesProcessResource

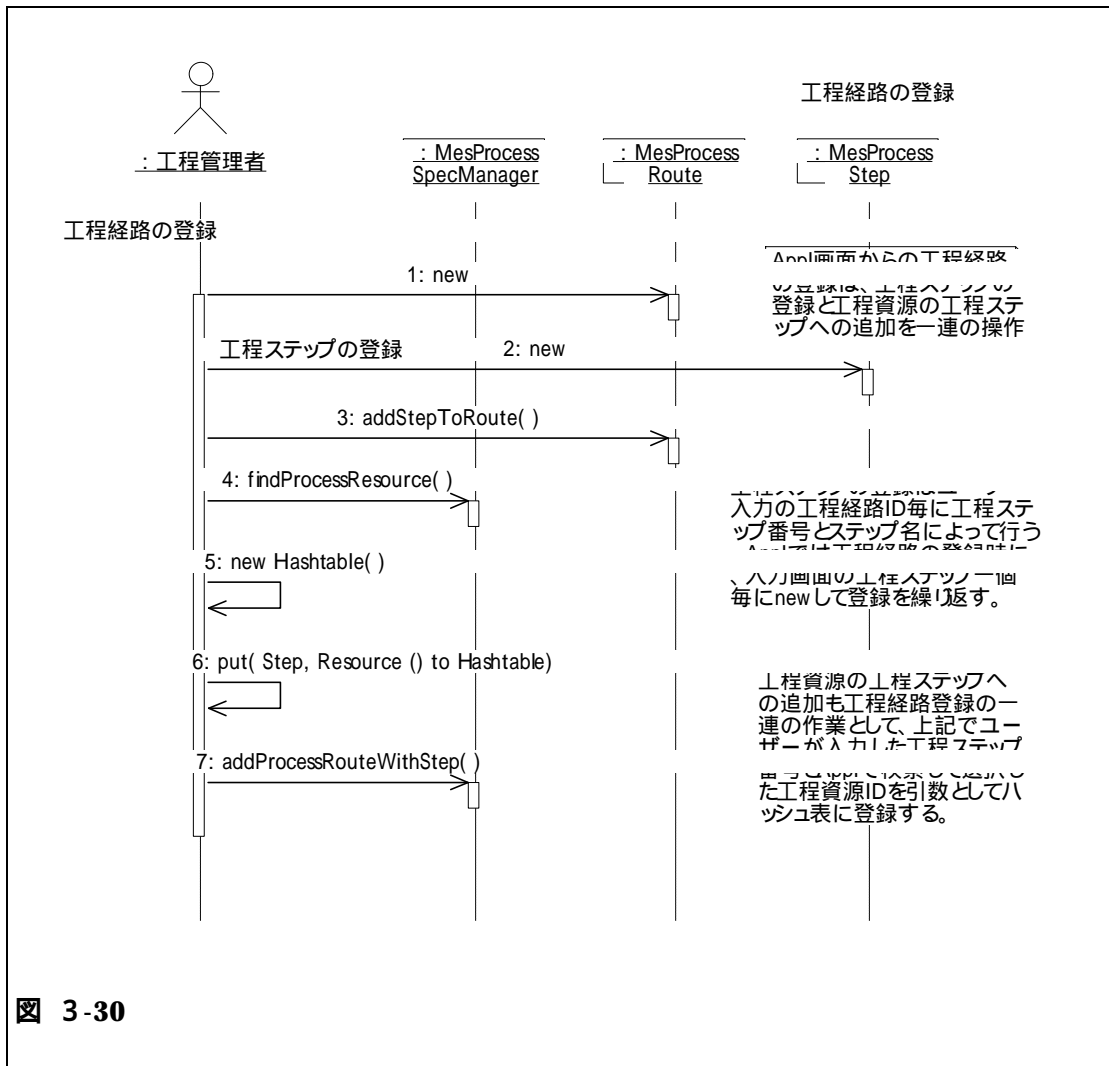
本クラスはある工程ステップを実行できる製造設備のグルーピング機能を提供する。

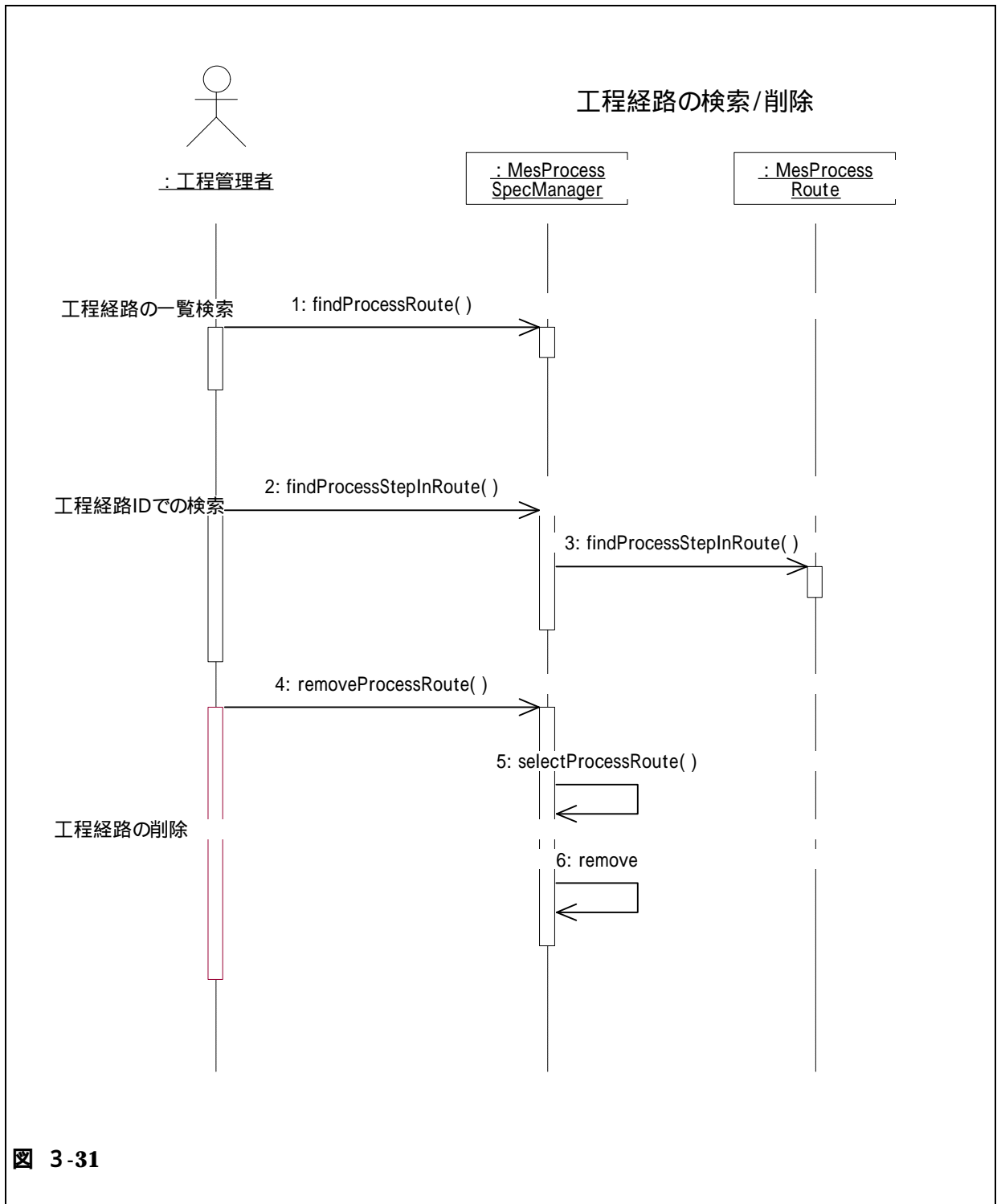
- 製造設備資源情報を設定、取得する。
- 製造設備種別を設定、取得する。
- 関連先 MesProcessStep を設定、取得する。
- 製造設備種別を取得する。
- 工程資源の製造設備を削除する
- 製造設備情報を工程資源に登録する。

#### ( 5 ) MesProcessEquipmentInfo

本クラスは製造設備情報を表す。

- 関連先 MesProcessResource を取得、設定する。
- 製造設備型番を設定、取得する。
- 製造設備種別を設定、取得する。





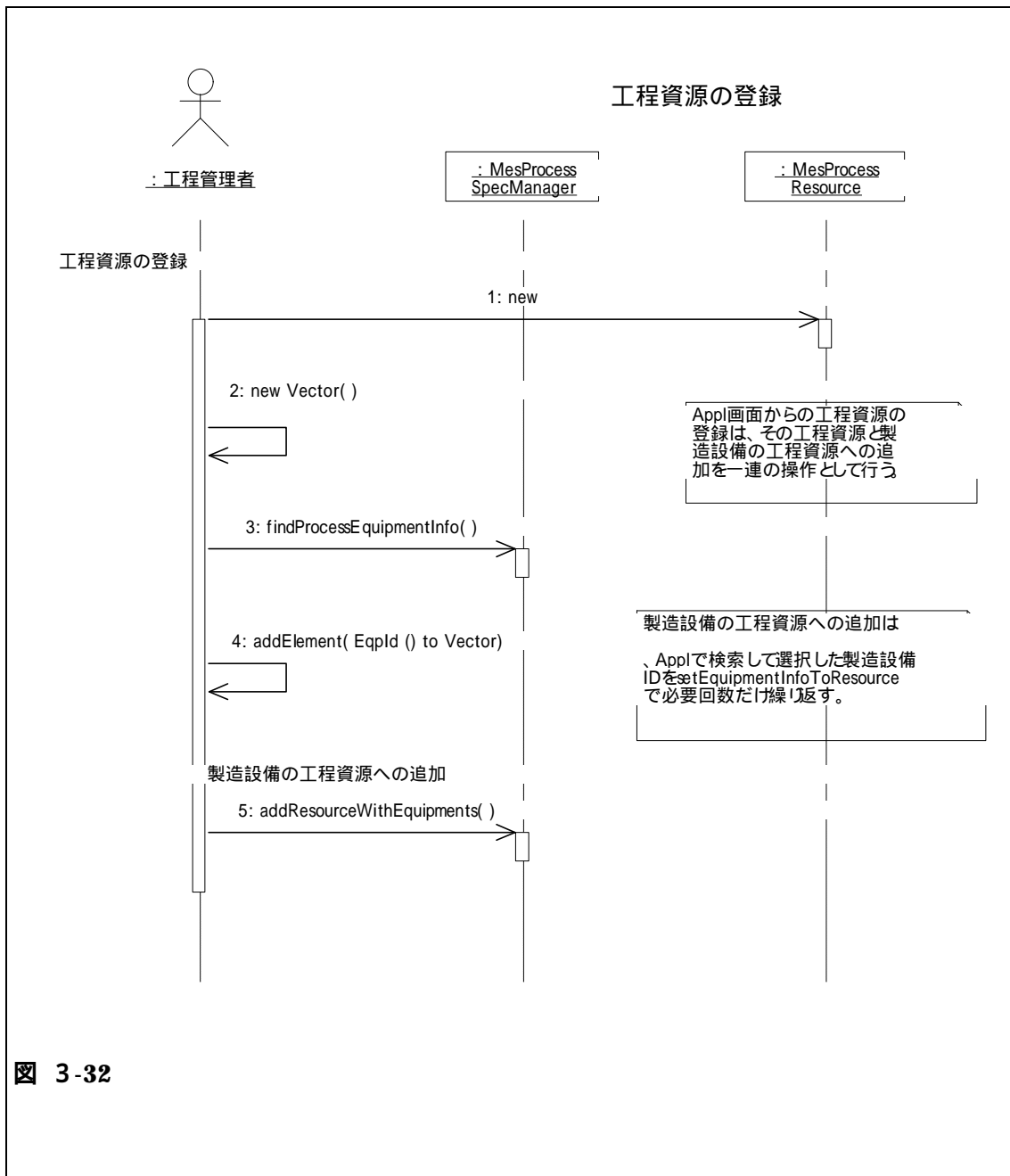
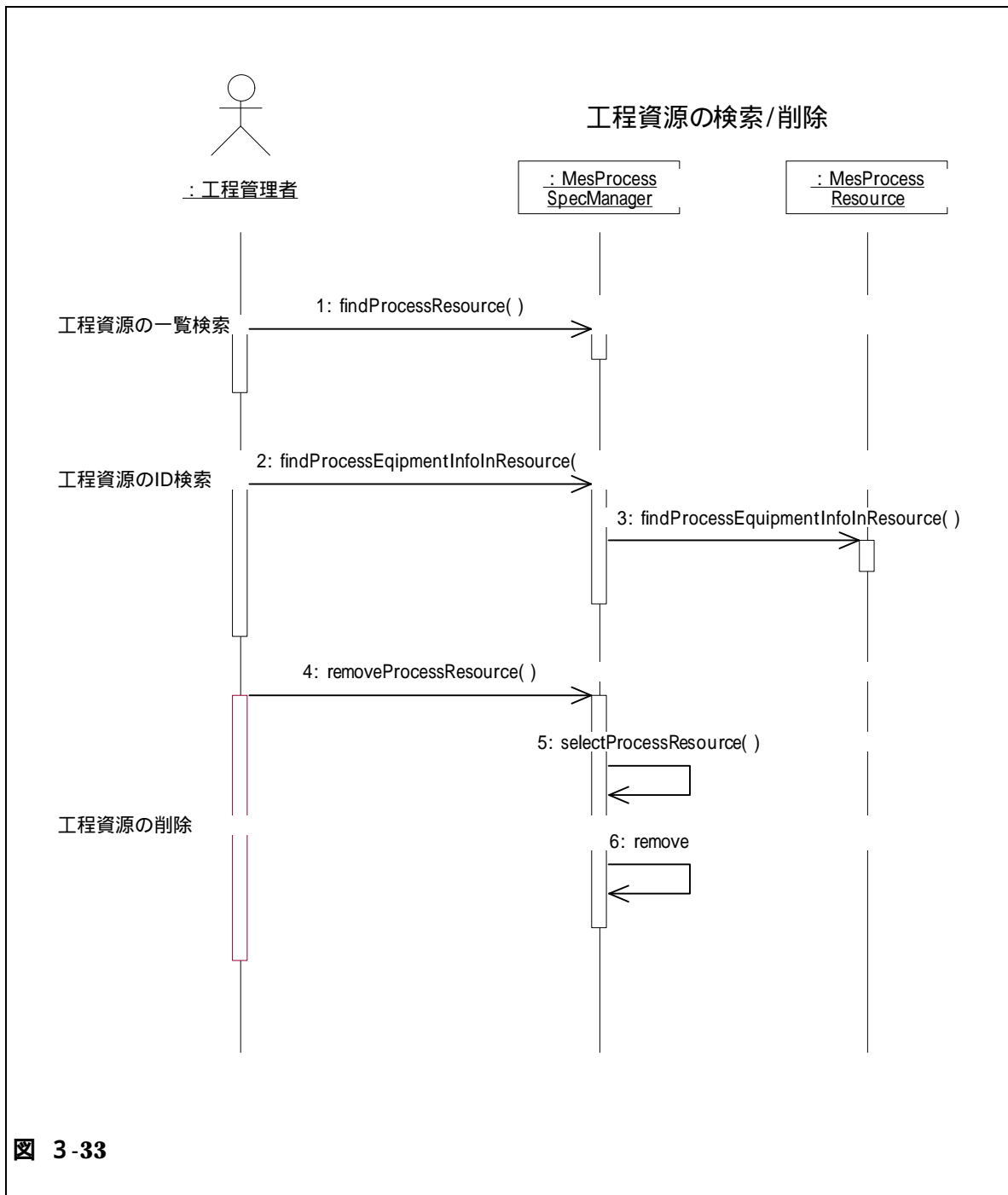
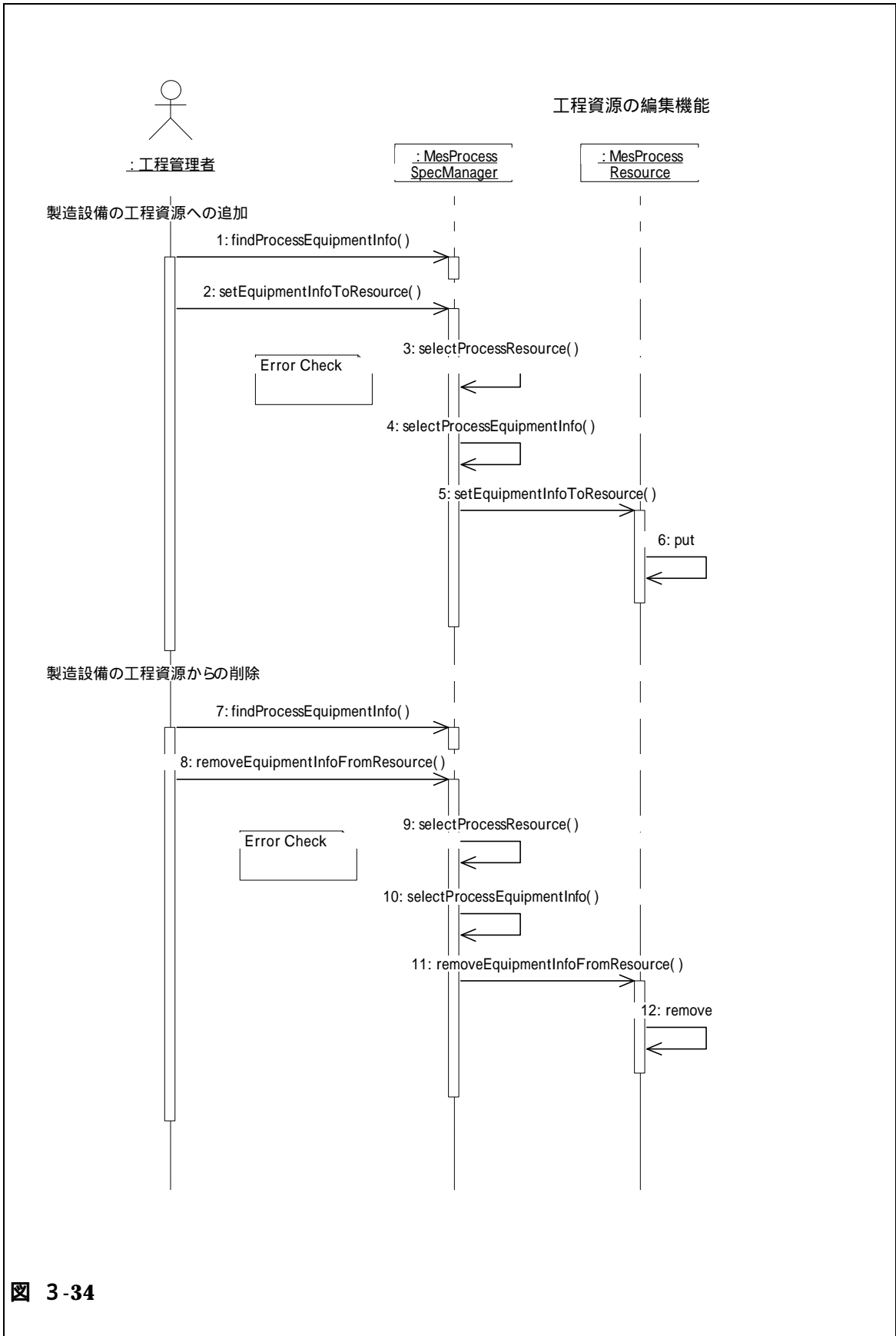


図 3-32





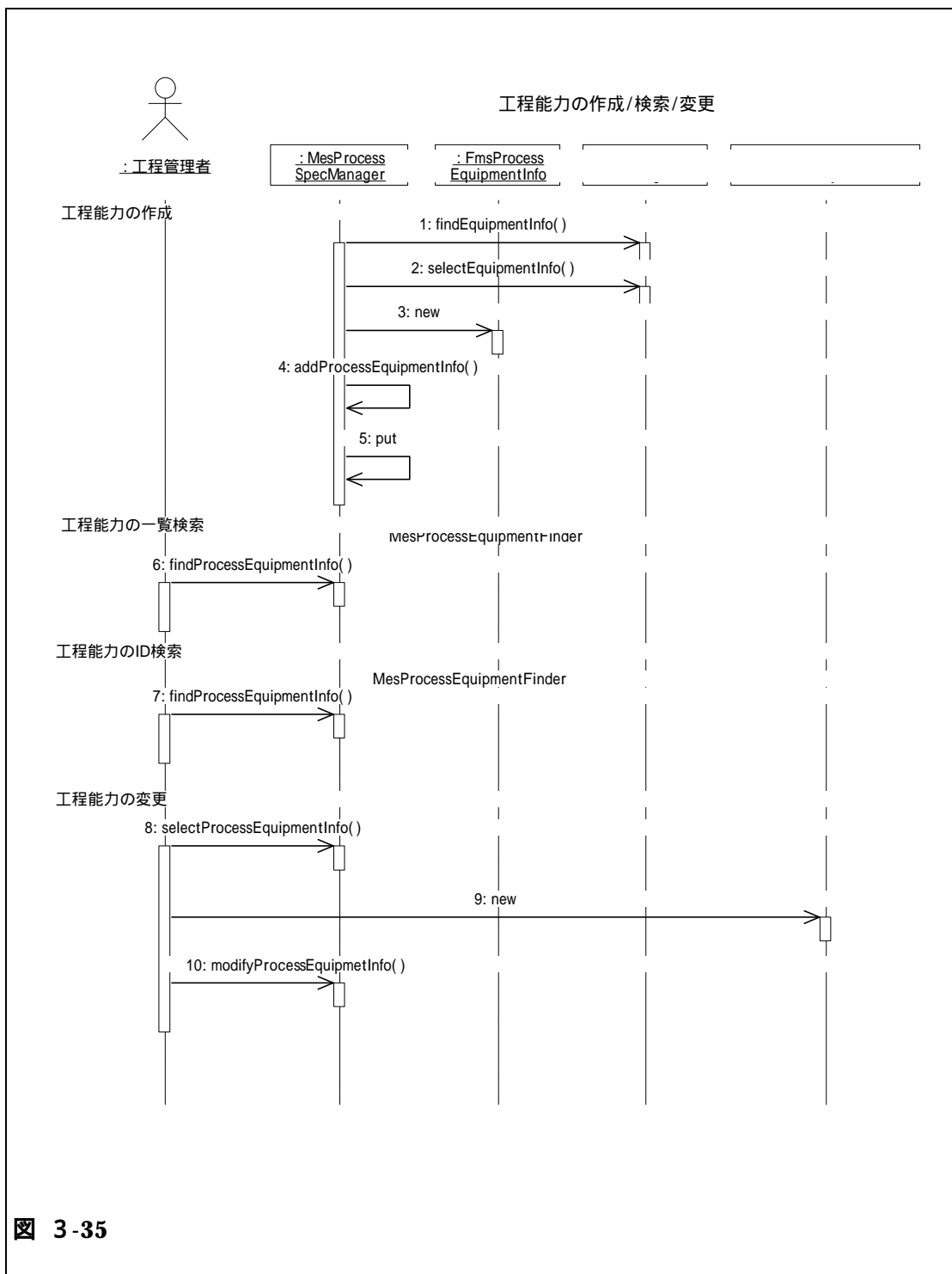
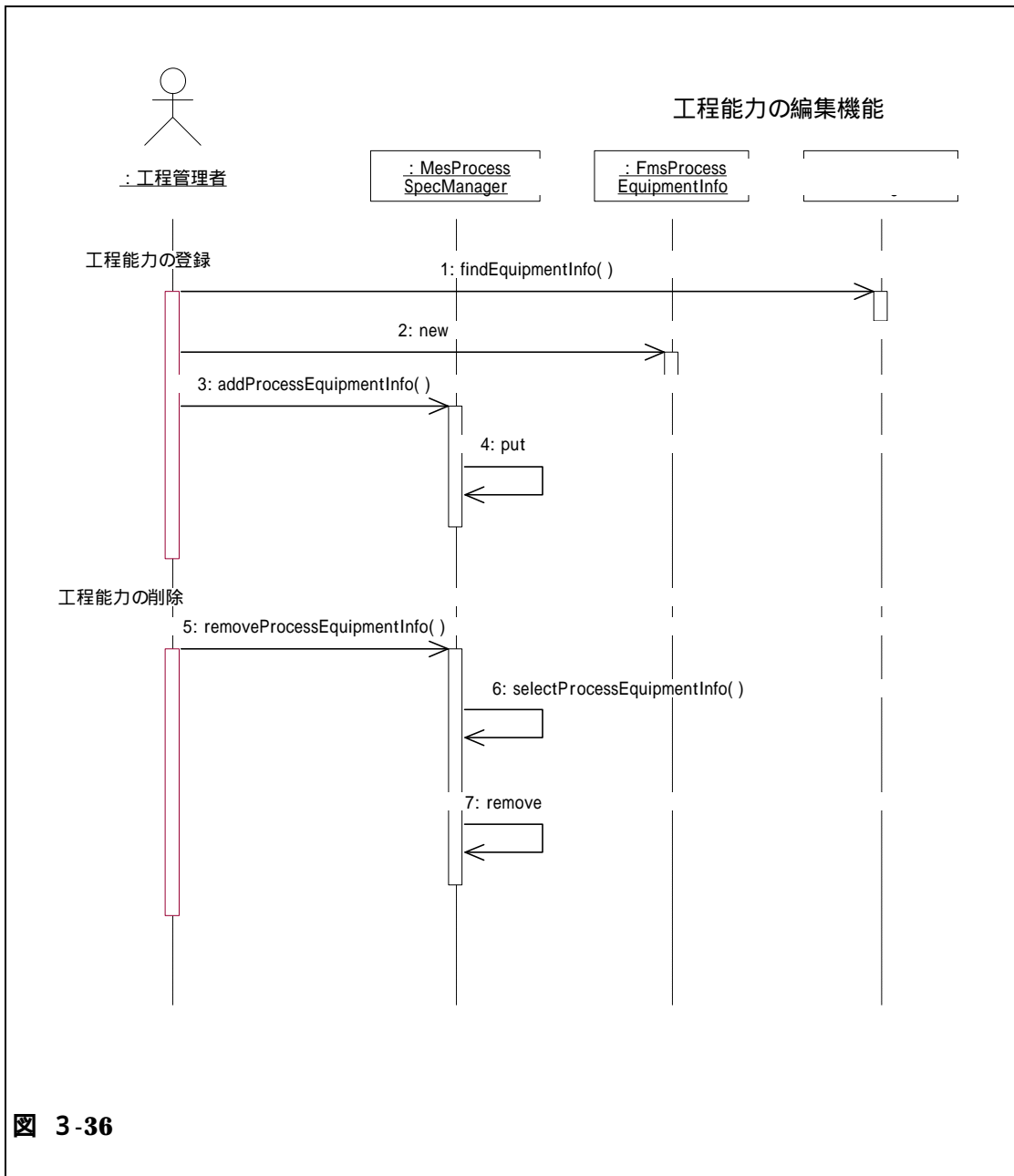


図 3-35



### 3.8 設備管理機能グループ

設備管理機能グループは、工場内の製造設備と MES 間の作業指示、作業実績、アラーム等のインタフェースとなる機能を提供する。この機能グループは、個々の製造設備に対応するクラスと、それらの設備をまとめて管理するクラスから構成されている。

ここで扱う設備は、物理的な実体を持たない抽象的な設備である。抽象設備は、下記の三つの性質を持つものとしてモデル化されている。

- 稼働状態の管理単位としての設備 (MesEquipment)
- 工程の実行単位としての設備 (MesProcessEquipment)
- 制御装置の単位としての設備 (MesProcessEquipmentAdaptor)

#### 3.8.1 設備管理コンポーネント

工程管理機能グループからは MesProcessEquipment に対し、MesWorkOrder が差し立てられ、MesProcessEquipment が MesEquipmentAdapter 経由で制御装置の API を呼び出す。作業が終了すると、制御装置からは MesProcessEquipmentCallback が呼び出され、作業実績 (MesWorkResult) が MesWorkOrder と関連づけられる。

MesProcessEquipment が工程管理機能グループとのインタフェース機能を持ち、MesProcessEquipmentAdapter が制御装置とのインタフェース機能を持つため、設備との接続に関しては CORBA の知識を必要としない。



## 加工設備 (工作機械)

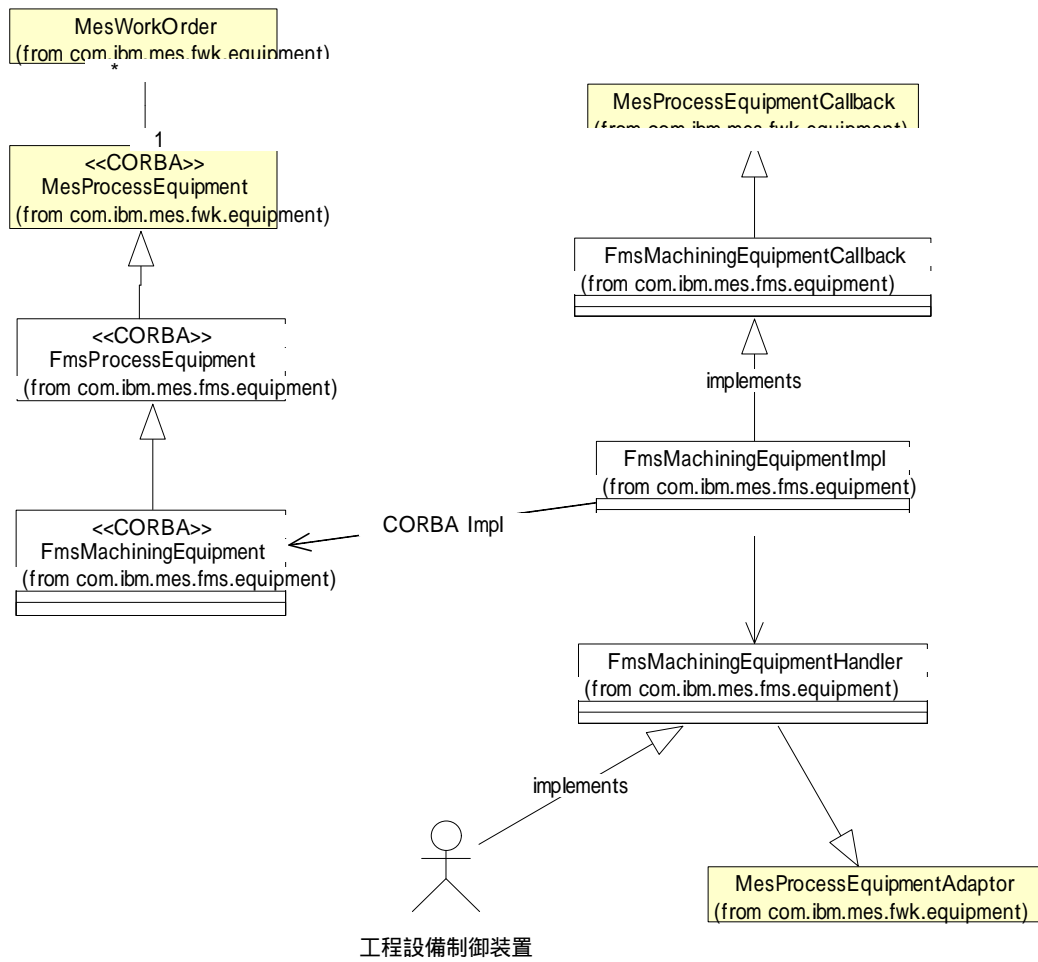


図 3-38

## 搬送設備

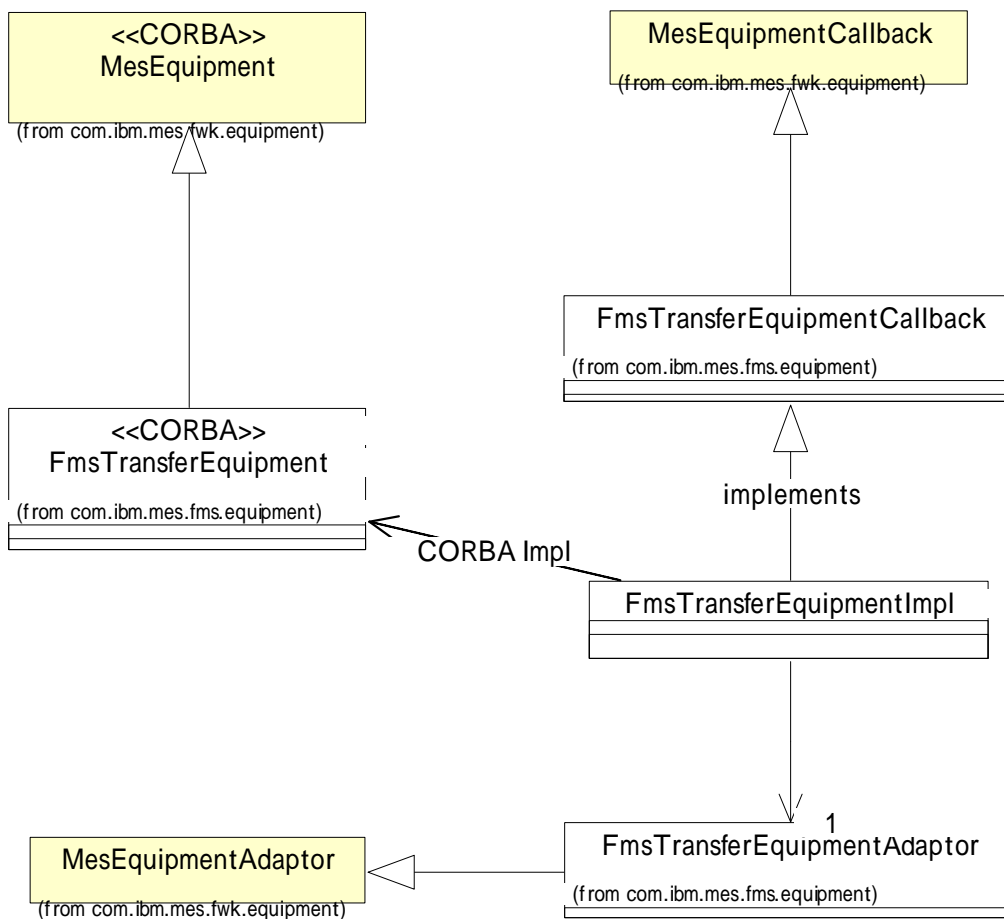


図 3-39

### ( 1 ) MesEquipment

このインタフェースは、抽象設備のための以下の機能を定義する。

- 設備の設置場所、型番、種別を設定する。
- 設備の設置場所、型番、種別を取得する。
- 設備の始動、遮断を行う。
- アラームの発生を通知する。
- 現在発生中のアラームがあるかどうかをチェックする。
- 発生中のアラームを取得する。
- 全てのアラームを解除する。
- 作業者による介入を要求する。
- 作業者による介入要求が発生しているかどうかをチェックする。
- 作業者による介入要求を解除する。
- 設備がオンライン状態かどうかをチェックする。
- 設備をオフラインにし、設備を製造工程内で利用不可とする。
- 設備をオンラインにし、設備を製造工程内で利用可能とする。
- 設備に搬送装置に対するポート(MesPort)を登録する。
- 設備に登録されている全てのポートを検索する。
- 設備からポートを削除する。
- 設備に登録されているポートを取得する。

### ( 2 ) MesEquipmentAdapter

このインタフェースは、抽象設備からの要求に従い、その制御装置を呼び出すための機能を定義する。

- 設備の始動を行う。
- 設備の遮断を行う。
- 発生しているアラームの解除を行う。
- 設備をオフラインにする。
- 設備をオンラインにする。

### ( 3 ) MesEquipmentCallback

このインタフェースは、抽象設備がその制御装置から呼び出されるメソッドを定義する。

- 設備の始動が開始された通知を受け取る。
- 設備の始動が完了した通知を受け取る。
- 設備の始動が取り消された通知を受け取る。
- 設備の遮断が開始された通知を受け取る。
- 設備の遮断が完了した通知を受け取る。
- 設備の遮断が取り消された通知を受け取る。

- アラームが解除された通知を受け取る。
- 全てのアラームが解除された通知を受け取る。
- 設備がオンラインになった通知を受け取る。
- 設備がオフラインになった通知を受け取る。
- 設備の緊急停止状態が解除された通知を受け取る。
- 設備の緊急停止が発生した通知を受け取る。

#### (4) MesEquipmentManager

このインターフェースは、MesEquipment を管理（登録、取得）する。

#### (5) MesPort

このインタフェースは、ポート(搬送装置との接続部分。パレットチェンジャなど)の持つ機能を定義する。

#### (6) MesProcessEquipment

このインタフェースは、工程の割付対象となる抽象工程設備の機能を定義する。

- 工程内作業指示を登録する。
- 全ての工程内作業指示を取得する。
- 次に着手すべき工程内作業指示を取得する。

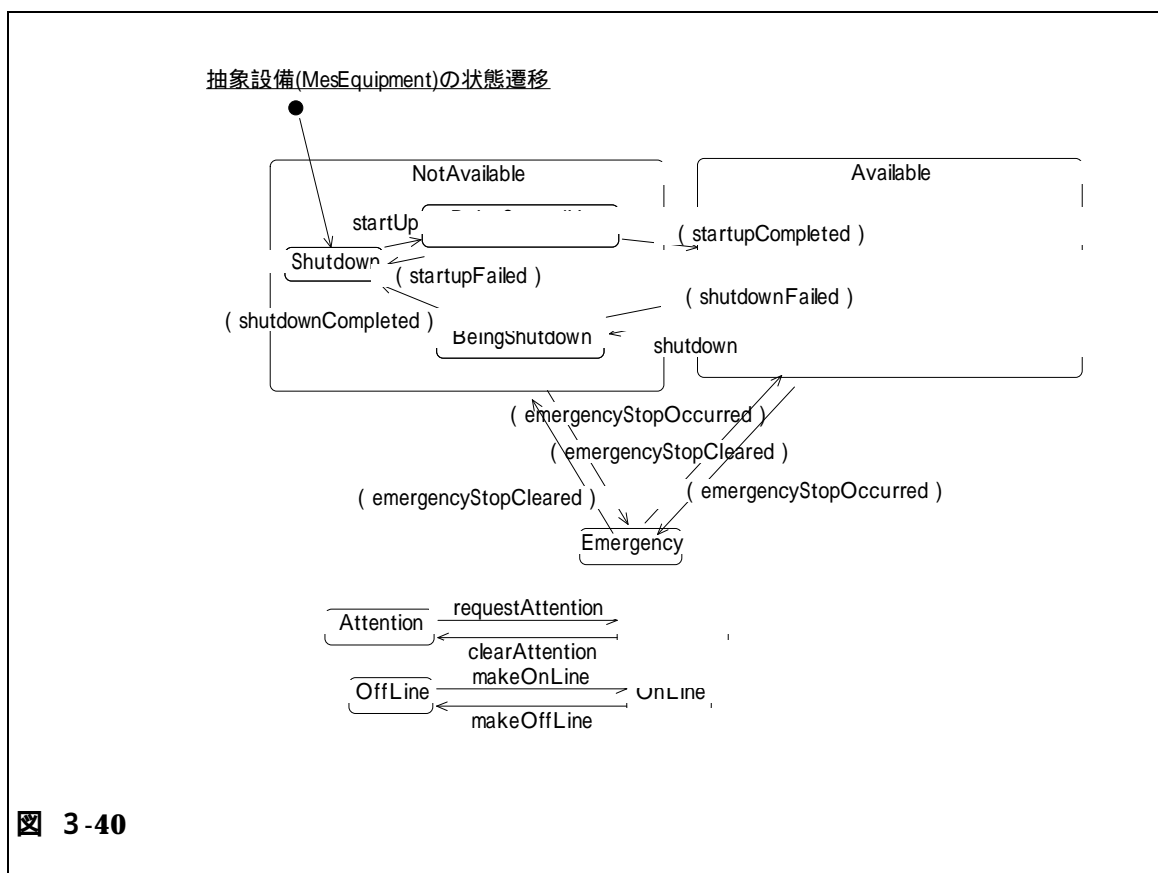


図 3-40

- 状態を指定して工程内作業指示を取得する。

- 工程内作業指示に対応した作業指図を取得する。
- 工程内作業を一時停止状態にする。
- 工程内作業の一時停止状態を解除する。
- 工程内作業指示を取得する。
- 工程内作業を取り消す。
- 工程内作業を中断する。
- 工程内作業の開始を要求する。
- 工程内作業を完了する。
- 工程内作業履歴を追加する。
- 工程設備に現在設定されている運転モード（自動、半自動、手動）を取得する。
- 運転モードを自動運転モードに設定する。
- 運転モードを手動運転モードに設定する。
- 運転モードを半自動運転モードに設定する。
- 工程設備に固有の運転モードを設定、取得する。固有の運転モードとは自動、半自動、手動のいずれかである。
- 中断された工程内作業指示を削除する。
- 完了している工程内作業指示を削除する。
- 指定された工程内作業指示を削除する。
- 工程内作業指示の情報を MesWorkOrderUpdateParam の値に従って更新する。

#### ( 7 ) MesProcessEquipmentAdapter

このインタフェースは、抽象工程設備からその制御装置を呼び出すための機能を定義する。

- 設備の運転モードを自動運転モードに変更する。
- 設備の運転モードを手動運転モードに変更する。
- 設備の運転モードを半自動運転モードに変更する。
- 工程内作業を開始する。
- 工程内作業指示が追加された通知を受け取る。

#### ( 8 ) MesProcessEquipmentCallback

このインタフェースは、抽象工程設備がその制御装置から受け取るコールバックを定義する。

- 作業開始の許可を要求する。
- 作業の開始通知を受け取る。
- 作業の完了通知を受け取る。
- 作業の停止通知を受け取る。

#### ( 9 ) MesAlarm

このクラスは、設備内で発生するアラームを定義する。

( 1 0 ) **MesAlarmEvent**

このクラスは、アラームの発生状況の変化を通知（発生通知、解除通知）するイベントを定義する。

( 1 1 ) **MesEquipmentAttentionEvent**

このクラスは、アテンション（作業者が介入する状態）の発生状況の変化を通知するイベントを定義する。

( 1 2 ) **MesEquipmentLineModeEvent**

このクラスは、ライン・モード（オンライン、オフライン）の変化を通知するイベントを定義する。

( 1 3 ) **MesEquipmentManagerServer**

このクラスは、MesEquipment の起動サーバを定義する。

( 1 4 ) **MesEquipmentOpMode**

このクラスは、設備の運転モード（自動、半自動、手動）を定義する。運転モードとは、MesProcessEquipment が工程作業を実行する際の実行方法を定義するものである。MesProcessEquipment の運転モードを問い合わせた結果として戻される。

( 1 5 ) **MesEquipmentOpModeEvent**

このクラスは、運転モード（MesEquipmentOpMode）の変更を通知するイベントを定義する。

( 1 6 ) **MesEquipmentProcessingEvent**

このクラスは、設備内工程処理状態（開始、完了、停止）の変更を通知するイベントを定義する。

( 1 7 ) **MesEquipmentServer**

このクラスは、抽象設備の起動サーバを定義する。

( 1 8 ) **MesEquipmentState**

このクラスは、設備の稼働状態を定義する。

- 利用可能状態
- 利用不可状態
- 始動中状態
- 遮断中状態
- 遮断状態
- 緊急停止状態
- 不明な状態

( 1 9 ) **MesEquipmentStateEvent**

このクラスは、設備の稼働状態（MesEquipmentState）の変更を通知するイベントを定義する。

( 2 0 ) **MesProcessEquipmentServer**

このクラスは、抽象工程設備の起動サーバを定義する。

( 2 1 ) **MesProcessEquipmentState**

このクラスは、設備の稼働状態（待機中、実行中）を定義する。

( 2 2 ) **MesProcessEquipmentStateEvent**

このクラスは、設備の稼働状態の変化を通知するイベントを定義する。

( 2 3 ) **MesWorkHistoryElement**

このクラスは、工程内作業と共に記録される履歴項目（タイムスタンプ、ロット ID、工程 ID、工程内作業指示 ID、コメント）を定義する。

( 2 4 ) **MesWorkInstruction**

このクラスは、工程設備に渡される工程内作業指図を定義する。

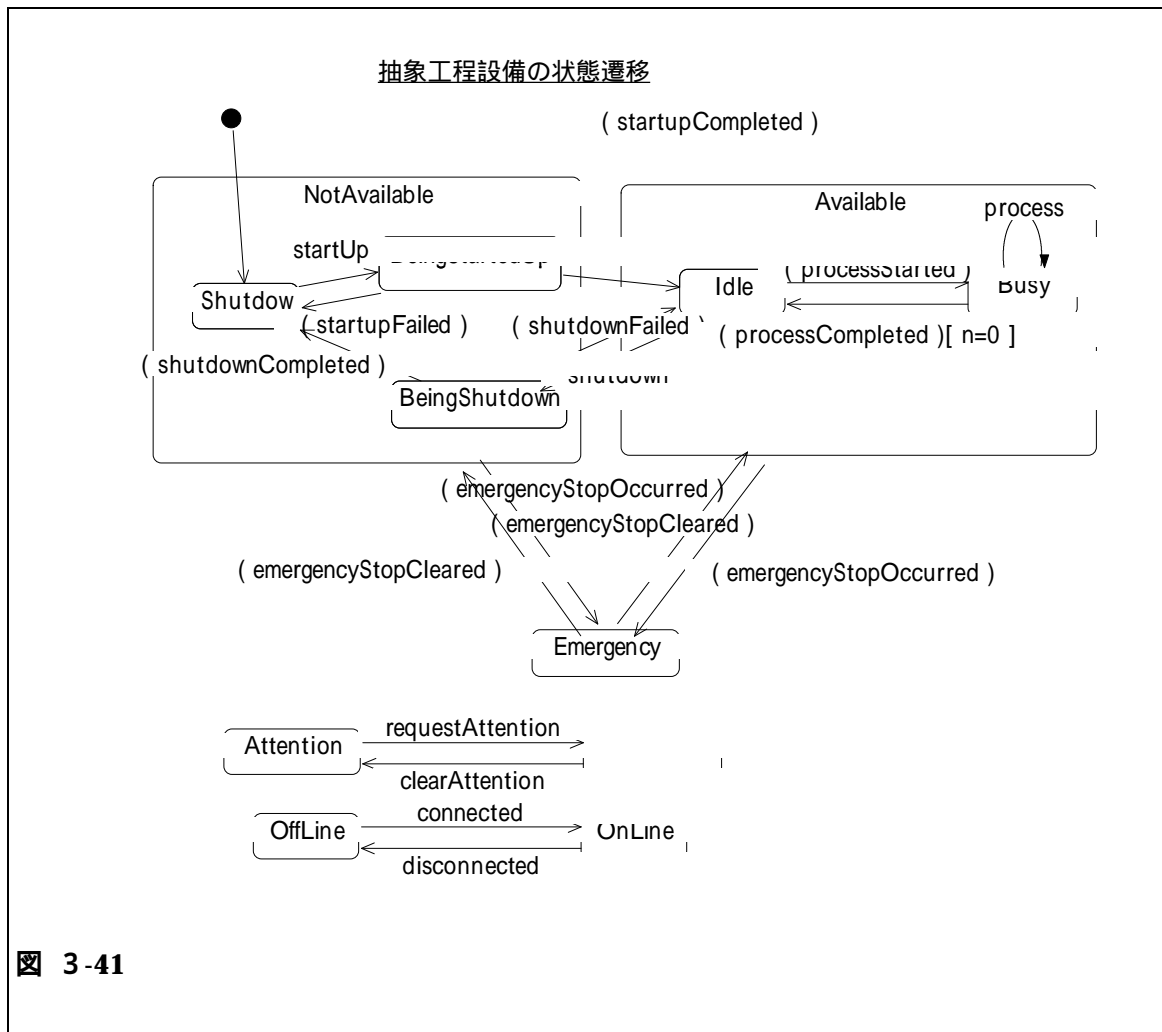


図 3-41

( 2 5 ) **MesWorkOrder**

このクラスは、工程設備の受け取る工程内作業指示を定義する。

- 工程内作業指示の ID を設定、取得する。
- 工程内作業指示の工程 ID を設定、取得する。
- 工程内作業指示を含むロットの ID を設定、取得する。
- 作業予定数量を設定、取得する。
- 作業開始予定日時を設定、取得する。
- 作業完了予定日時を設定、取得する。
- 工程内作業指示の優先順位を設定、取得する。
- MesLotJob を設定、取得する。
- 工程内作業指示が割り付けられた設備の ID を設定する。
- 差し立て内での工程内作業指示の順位を取得する。
- 工程内作業指示の状態 ( MesWorkOrderState ) を取得する。
- 段取り替えの必要性を設定、取得する。
- 工程内作業を着手不可にする。
- 工程内作業を着手可能にする。
- 工程処理の開始、完了、停止を通知する。
- 工程内作業の開始許可を要求する。
- 工程内作業を中断する。
- 工程内作業を取り消す。
- 工程内作業を完了する。
- 工程内作業指示を一時停止状態とする。
- 工程内作業指示の一時停止状態を取得する。
- 工程内作業指示の一時停止状態を解除する。
- 工程内作業指示のグローバルな一時停止状態を取得する。
- 工程内作業指示のグローバル ( 代替設備全部 ) な一時停止を設定・解除する。
- 工程内作業指示のローカル ( 特定の設備 ) な一時停止を設定・解除する。
- 差し立て内での工程内作業指示の順位を設定する。
- 作業結果を登録する。
- 工程内作業履歴を登録する。

#### ( 2 6 ) MesWorkOrderAddParam

このクラスは、工程設備に対して工程内作業指示を割り付けるためのパラメータ ( 工程内作業を要求したオブジェクト、工程内作業指図、工程内作業指示 ) を定義する。

#### ( 2 7 ) MesWorkOrderEvent

このクラスは、設備の持つ工程内作業指示の状態 ( MesWorkOrderState ) の変更を通知するイベントを定義する。

#### ( 2 8 ) MesWorkOrderState

このクラスは、工程内作業指示の状態を定義する。

- 作業中断
- 作業完了
- 作業中
- 着手不可
- 作業中でかつ工程処理が完了している状態
- 作業中でかつ工程処理が開始不可能な状態
- 作業中でかつ工程処理が開始可能な状態
- 作業中でかつ工程処理が開始されている状態
- 作業中でかつ工程処理が停止中の状態
- 着手可能
- 前工程が完了して着手可能であるが、グローバルまたはローカルにサスペンドしている状態

#### ( 2 9 ) **MesWorkOrderTransferState**

このクラスは、工程内作業指示で指示されたロットの搬送状態を定義する。

- 搬入指示により搬入を実行中である状態
- 搬出中の状態
- 搬入済みの状態
- 搬出済みの状態
- 搬入未着手の状態

#### ( 3 0 ) **MesWorkOrderUpdateParam**

このクラスは、工程設備に登録された工程内作業指示の情報を更新するためのパラメータを定義する。

- 作業の完了予定日時
- 作業の着手予定日時
- 優先順位
- グローバル・サスペンド・フラグ
- 差し立て内での順位
- 工程内作業指図
- 工程内作業指示 ID

#### ( 3 1 ) **MesWorkResult**

このクラスは、工程設備の返す工程内作業結果を定義する。工程内作業結果は、工程内作業指示に対して、その作業結果の更新を要求することができる。

- 工程内作業を行った設備の ID を設定、取得する。
- 工程内作業で発生した不良品数を設定、取得する。
- 工程内作業の完了日時を設定、取得する。
- 工程内作業で処理が完了した数量を設定、取得する。

- 工程内作業を担当した担当者の ID を設定、取得する。
- 工程内作業の開始日時を設定、取得する。

### 抽象工程設備-自動実行

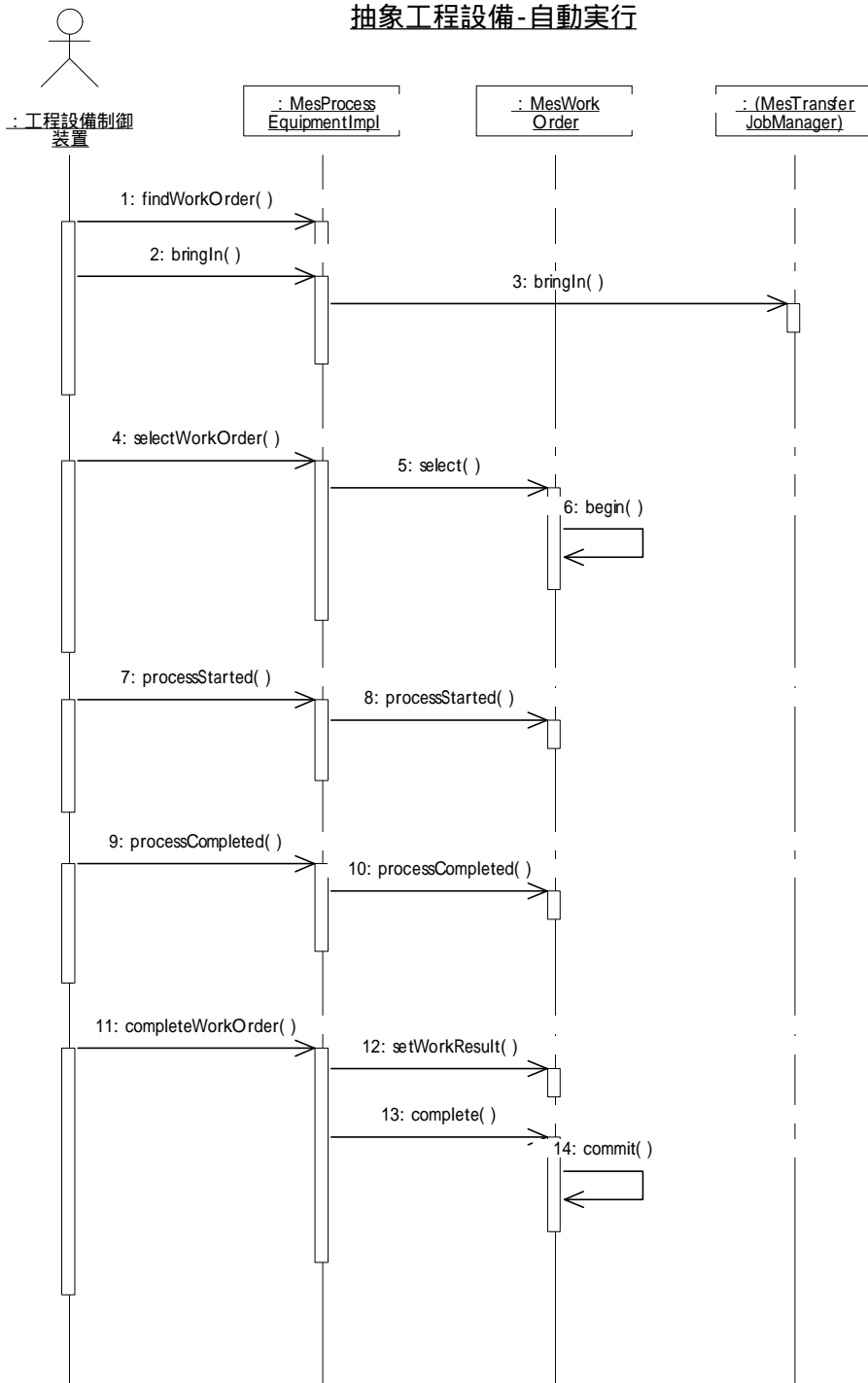


図 3-42

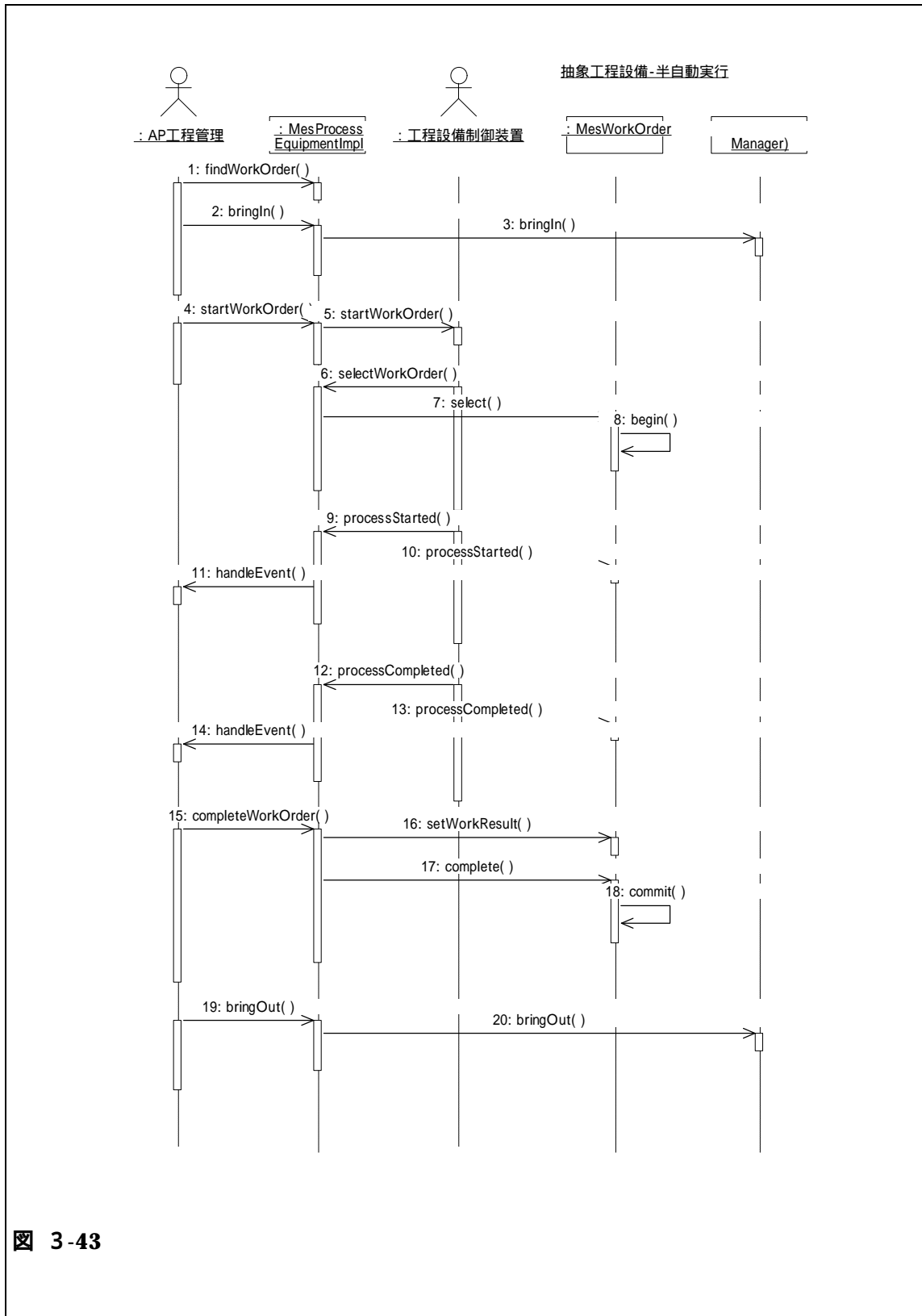
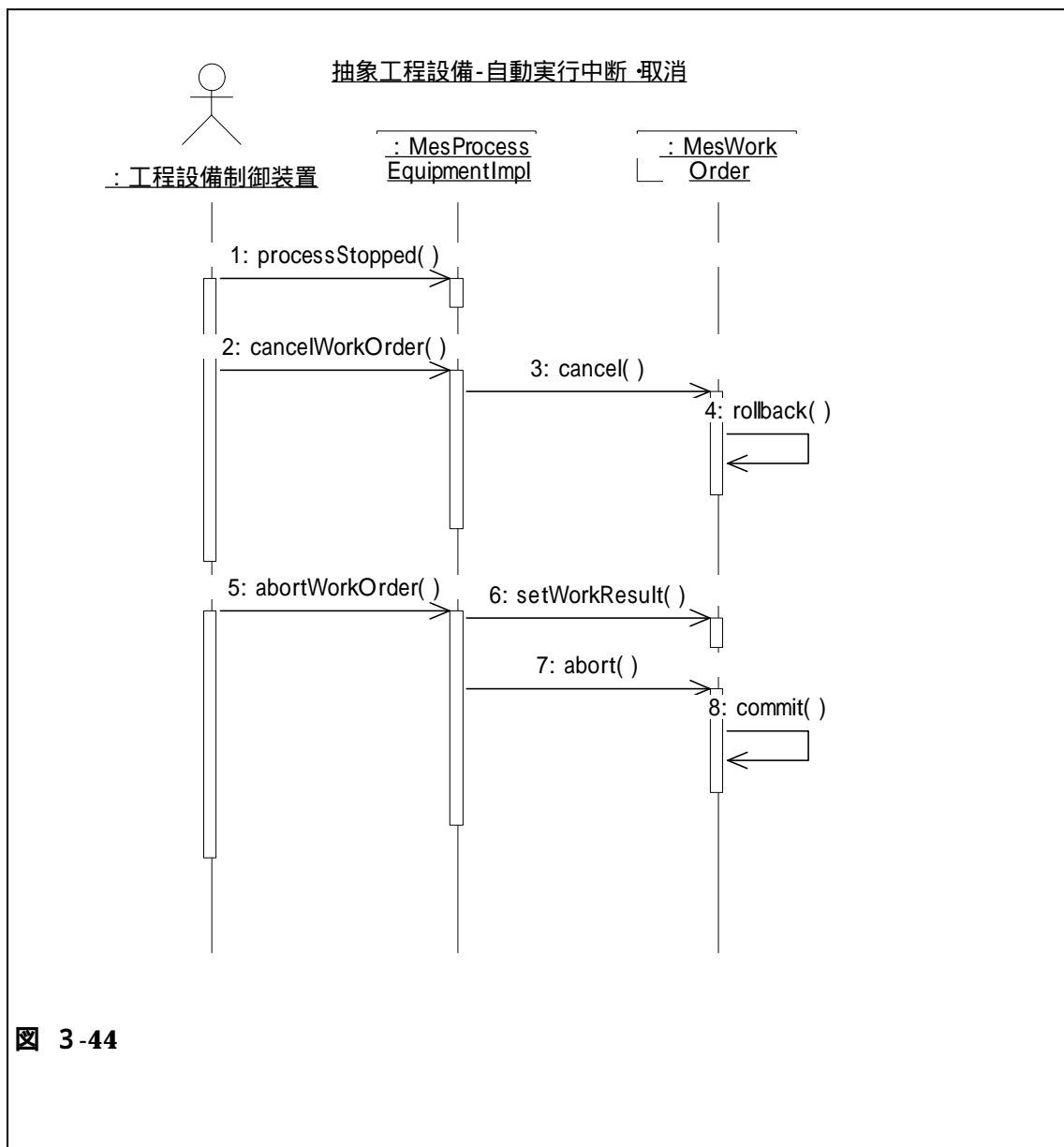


図 3-43



抽象工程設備-半自動実行中断・取消

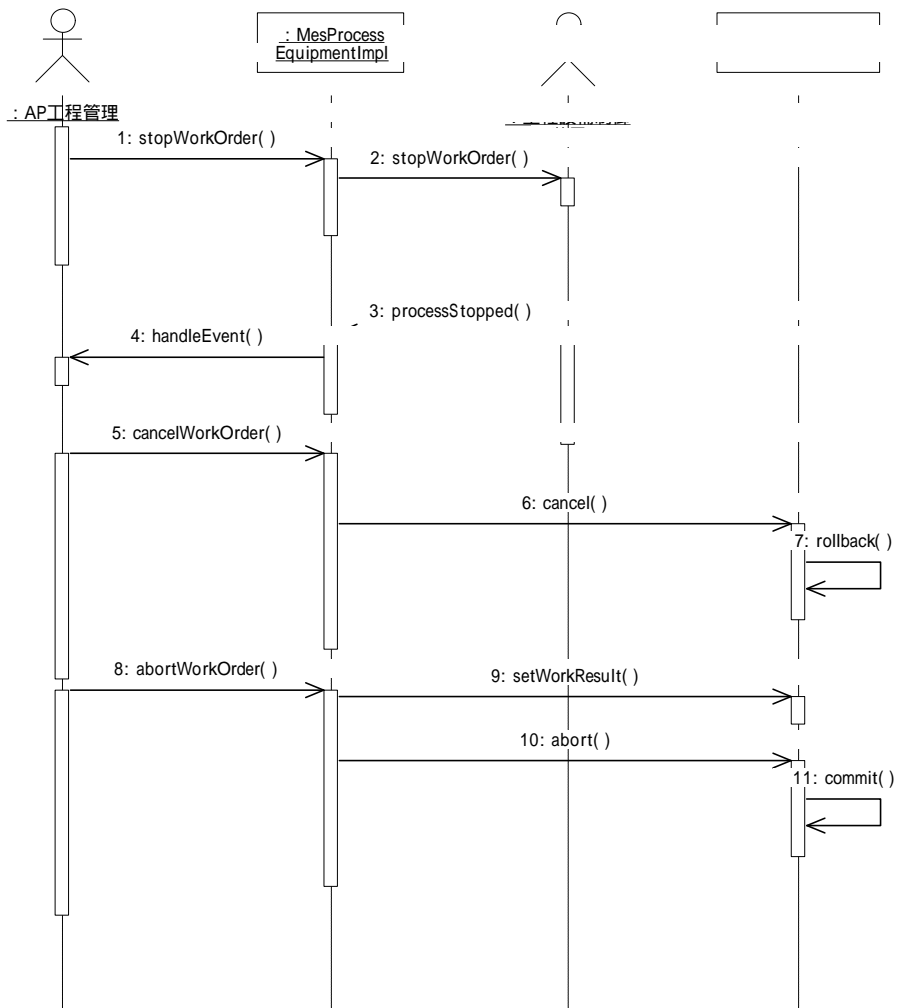


図 3-45

### 3.9 スケジュール管理機能グループ

#### 3.9.1 スケジューラインタフェースコンポーネント

##### (1) MesScheduler

このインタフェースはスケジューラへのインタフェース機能を定義する。既存のスケジューラと接続するため、抽象的なメソッドしか定義されていない。

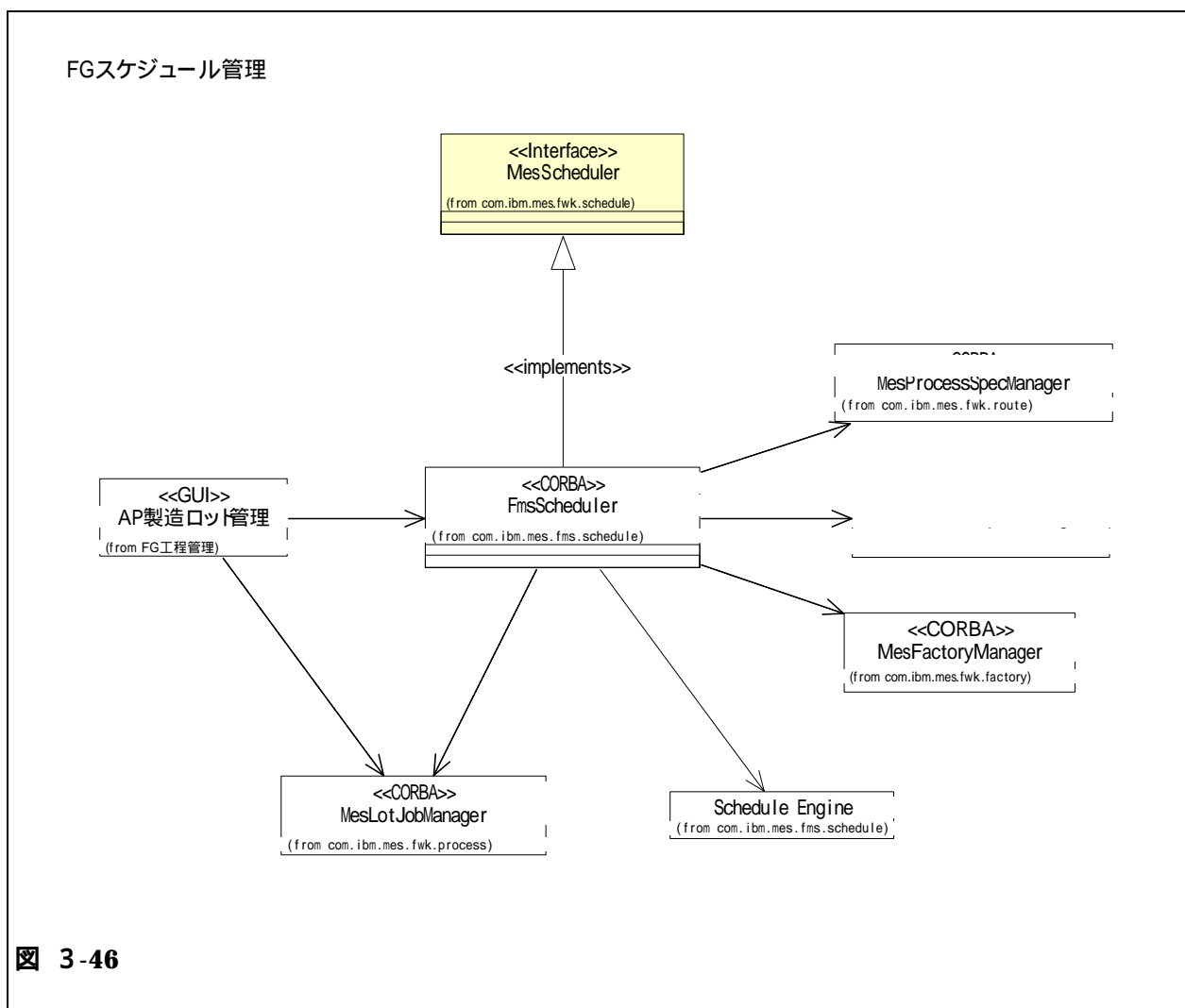
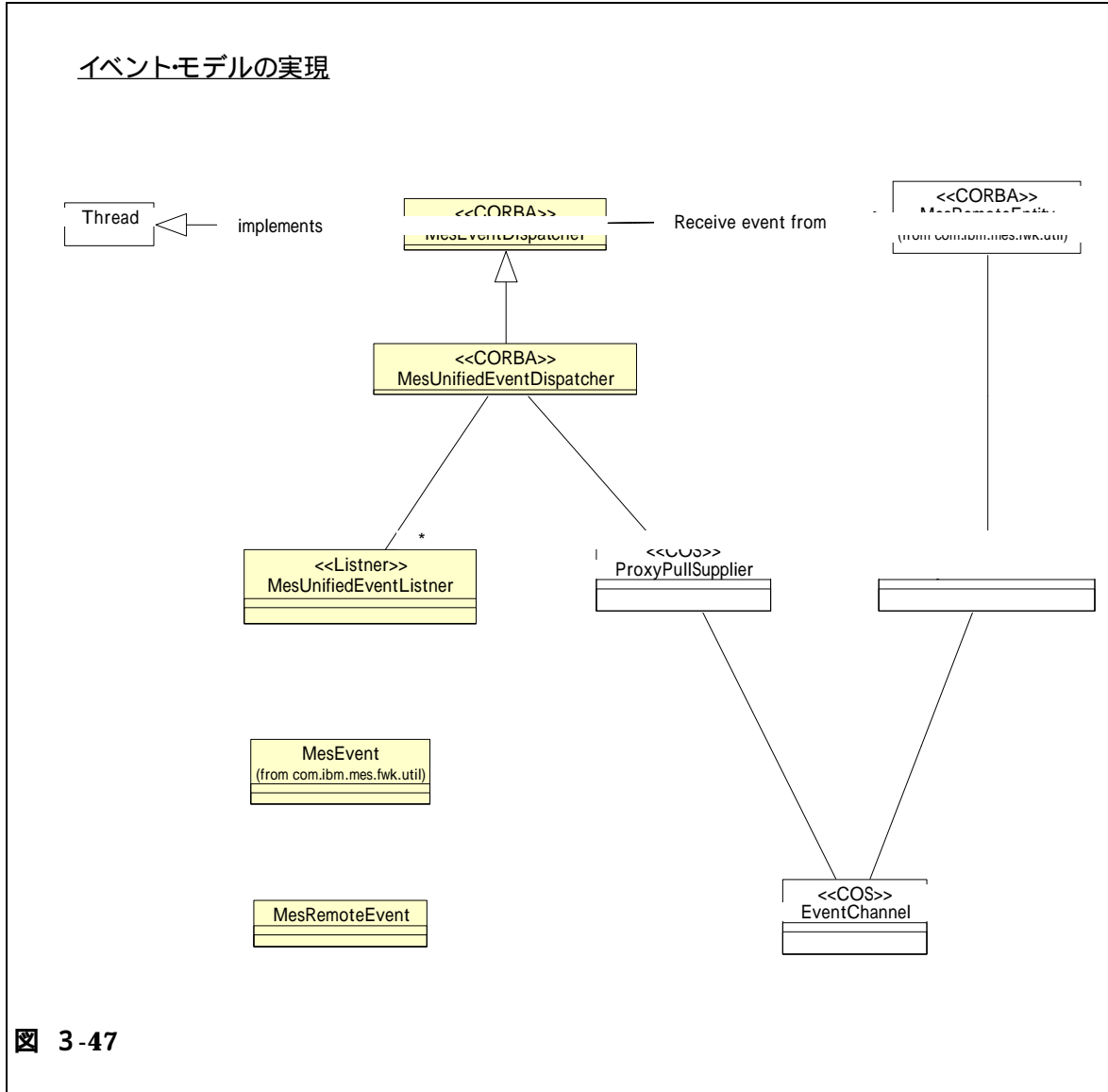


図 3-46

### 3.10 共通機能グループ

#### 3.10.1 イベント通知管理コンポーネント

CORBA の Event Service を使用している。イベントモデルについては、CORBA の仕様書を参照のこと。



#### (1) MesEvent

このクラスは、コンポーネント内に発生するイベントを定義する。

#### (2) MesEventDispatcher

このクラスは、CORBA サーバからのイベントを受け取り、ローカル空間内の Java オブジェクトに配信するディスパッチャを定義する。

#### (3) MesRemoteEvent

このクラスは、CORBA の Event Channel を通してイベントを通知するためのパケットを

定義する。このクラスは、IDL のマッピング上は struct に変換される。Event Channel が struct マッピングしか受け付けないため、MesEvent からこのクラスに変換して送る必要がある。

( 4 ) **MesUnifiedEventDipatchcer**

このクラスは、共通のインターフェースを通して全てのイベントの配信を行う

- イベントのリスナを登録する。
- イベント ( MesRemoteEvent ) の配信を行う。
- イベントのリスナの登録を削除する。

( 5 ) **MesUnifiedEventListener**

このインタフェースは、CORBA サーバからのイベントを受け取るリスナを定義する。

3.10.2 リモートエンティティコンポーネント

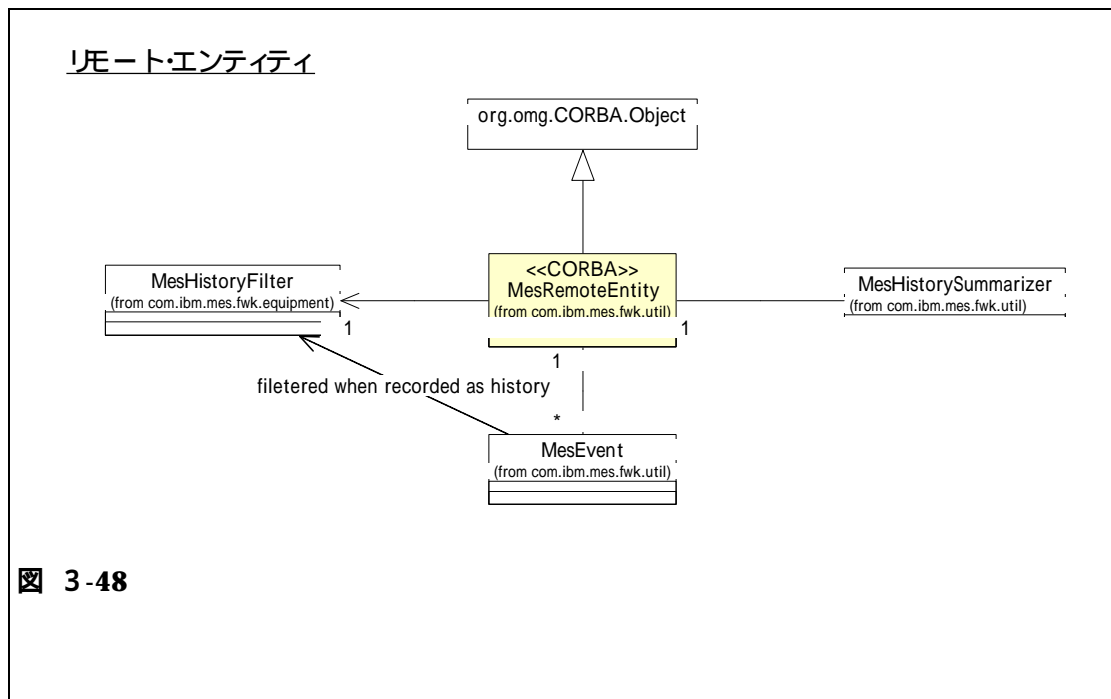


図 3-48

( 1 ) **MesRemoteEntity**

このインターフェースは、フレームワークを構成するリモート・オブジェクトの持つ共通機能を定義する。このインターフェースから取得された各種の属性はサーバ内部の属性のコピーであり、取得された属性に対する変更操作はサーバ内部の属性には反映されない。

( 2 ) **MesHistoryElement**

このクラスは、コンポーネント内で記録される履歴項目を定義する。

### ( 3 ) MesHistoryFilter

このクラスは、履歴を記録するかどうかのチェックを行うためのフィルタを定義する。

### ( 4 ) MesHistorySummarizer

このクラスは、MesRemoteEntity のオブジェクトの履歴からサマリを作成するた

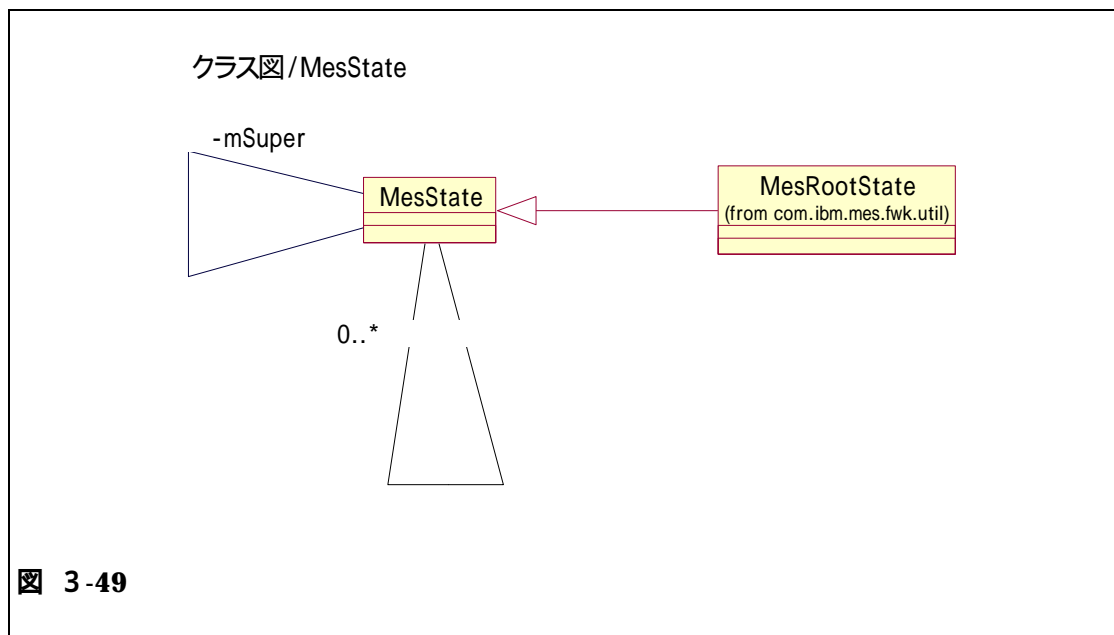


図 3-49

めの抽象クラスである。

## 3.10.3 状態管理コンポーネント

### ( 1 ) MesState

階層化された状態集合を木構造によって保持する。以下の機能を提供する。

- 状態 ID を取得する。
- 状態レベルを取得する。
- 状態名を設定、取得する。
- 上位状態を設定、取得する。
- 下位状態を設定、取得する。
- 与えられた状態がこの状態と同じかどうか判定する。
- 下位状態が登録されているかどうかを取得する。
- 指定された下位状態をその全ての下位状態とともに削除する。
- 与えられた状態 ID がこの状態あるいは下位状態として登録されている場合 その状態を返す。
- 与えられた状態 ID がこの状態あるいは上位状態として登録されている場合 その状態を返す。

## 状態図/状態表現

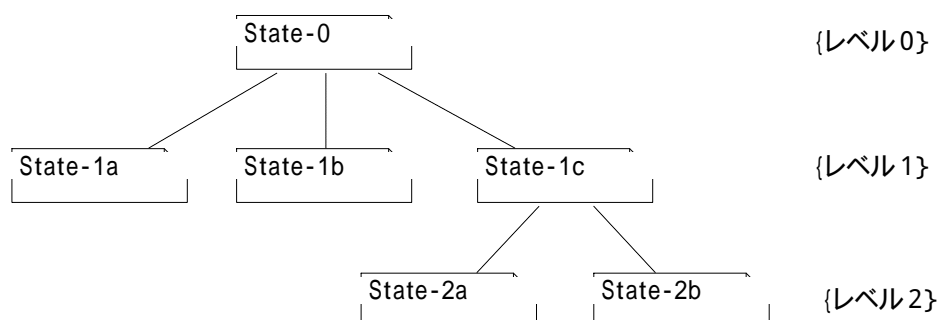
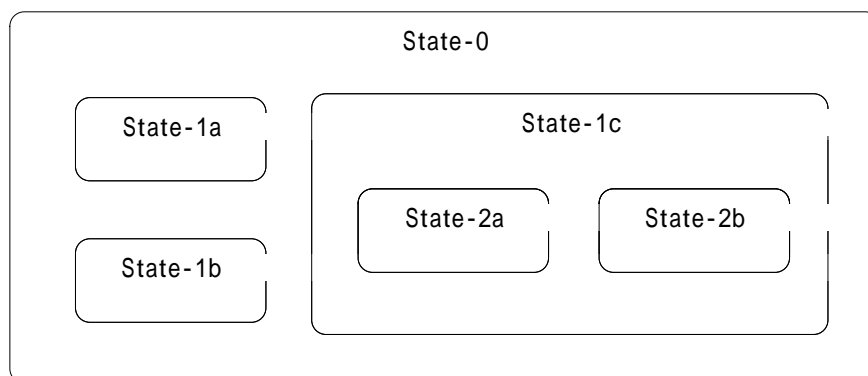


図 3-50

### ( 2 ) MesRootState

最上位状態。木構造によって階層化された全状態集合のルート(根)となる。このクラスは個別の状態に相当する MesState から派生したものであるが、特にその時点での状態を保持することができる。さらに、この状態集合に属する状態は addState (最上位状態の直下に属する場合)、もしくは addStateTo(既に登録されている状態の下位状態を登録する場合)によって登録される。これらのメソッドによって登録された状態についてはこの状態集合内での一意性が保証される。様々な状態を取り得る実体クラス(例えば、製造指示、製造ロット)には、該当する状態集合が関連付けられる。そのような実体クラスには、public な

メソッドとして 例えば `isInState()`、`getCurrentStateId()`、`getRootState()`を提供する。`isInState` はそのオブジェクトが与えられた状態もしくはその上位状態にあるかどうかを返答 する。`getCurrentStateId` はその時点でのオブジェクトの状態 ID を返す。さらに、`getRootState` は状態集合の全要素が関連付けられた木構造の根となる `MesRootState` もしくはその派生クラス(例えば `MesLotJobState`)を返す。`getRootState` によって 得られた木構造を辿ることによって状態集合に伴われる全ての状態 ID と対応する 状態名(例えば画面表示に使用)を参照することができる。

文書の終わり