

# 組込み分野のための UML モデルカタログ

2013 年 5 月版



組込み分野でモデリングする全ての人に捧げるモデル集

UMTP 組込み  
モデリング部会

2013.5.7 更新

本書は、組込み分野のさまざまなモデルを集めたカタログです。モデリングの初心者には教科書や参考書として、モデリングのベテランの方々にはモデルのヒントとして、ぜひともお手元に置いて活用してください。

UMTP は特定非営利活動法人 UML モデリング推進協議会の登録商標です。その他、本書に記載されている会社名、商品名などは、一般に各社の商標または登録商標です。

# 目次

<b>UML モデルカタログの歩き方</b> .....	<b>5</b>
UML モデルカタログとは .....	6
カタログの構成 .....	7
カタログの使い方 .....	9
ラインナップ .....	11
<b>モデルカタログ：製品編</b> .....	<b>12</b>
P001 孔版印刷機 .....	13
要求仕様 .....	13
モデル一覧 .....	16
エンティティに着目したモデル .....	17
P002 電子オルゴール .....	19
はじめに .....	19
要求仕様 .....	19
モデル一覧 .....	22
エンティティに着目したモデル .....	23
P003 インライン装置 .....	25
要求仕様 .....	25
モデル一覧 .....	29
ドメイン構造に着目したモデル .....	30
<b>モデルカタログ：機能編</b> .....	<b>32</b>
F001 認証 .....	33
要求仕様 .....	33
モデル一覧 .....	35
機能に着目したモデル .....	36
エンティティに着目したモデル .....	38
状態に着目したモデル .....	40
F002 自己診断 .....	42
要求仕様 .....	42
モデル一覧 .....	44
エンティティに着目したモデル .....	45
メタファを使ったモデル .....	47

<b>モデルカタログ : 部品編</b> .....	<b>49</b>
C001 目標制御 .....	50
はじめに .....	50
要求仕様 .....	50
モデル一覧.....	53
エンティティに着目したモデル.....	54
役割に着目したモデル.....	56
おわりに .....	58

## UML モデルカタログの歩き方

ここでは、みなさんがこのカタログを有効活用できるように、カタログの全体構成とお勧めの使い方を具体的にご紹介します。

## UML モデルカタログとは

昨今の組込み分野における開発規模の増大・複雑化・短納期化に対応する手法として、UML によるモデリングが成果を上げています。UML モデルを利用することで、大規模システムであっても、全体を俯瞰し、アーキテクチャレベルでシステムを理解したうえで各種の問題に対する対策を打てるので、生産性・品質を向上させることができます。

しかし、実情は UML モデリングを開発に取り入れてみたものの、モデリングの有識者がいないがために、その効果を十分に得ることができず、従来の方法に戻ってしまう現場が多々あります。

そのような状況を打開するために、初心者には UML モデリングに取り組む第一歩として、ベテランの方にはより良いモデリングのヒントとして、「カタログ」(本書)を提供することにしました。

本書が開発の現場で UML モデリングを利用する際の参考になれば幸いです。

本書の適用範囲は、主に、製品開発におけるソフトウェア設計工程です。「組込みソフトウェア向け 開発プロセスガイド<sup>1</sup>」(以下 ESPR) では、右図で示す範囲です。

ある部分をハードウェアで実現するかソフトウェアで実現するかは状況によって変わるため、明確にせず、システム要求定義やシステム・アーキテクチャ設計までをモデルに含めている場合もあります。

特に本書に関係の深い工程についての定義を以下に示します。

工程名	定義
ソフトウェア要求定義	当該製品を実現するためにソフトウェアとして実現が必要となる要求を明確にします。
ソフトウェア・アーキテクチャ設計	開発する組込みソフトウェアのアーキテクチャ (= 動作[振る舞い]と構造) を決定します。 ※後述する「PIM」設計モデルにあたります。

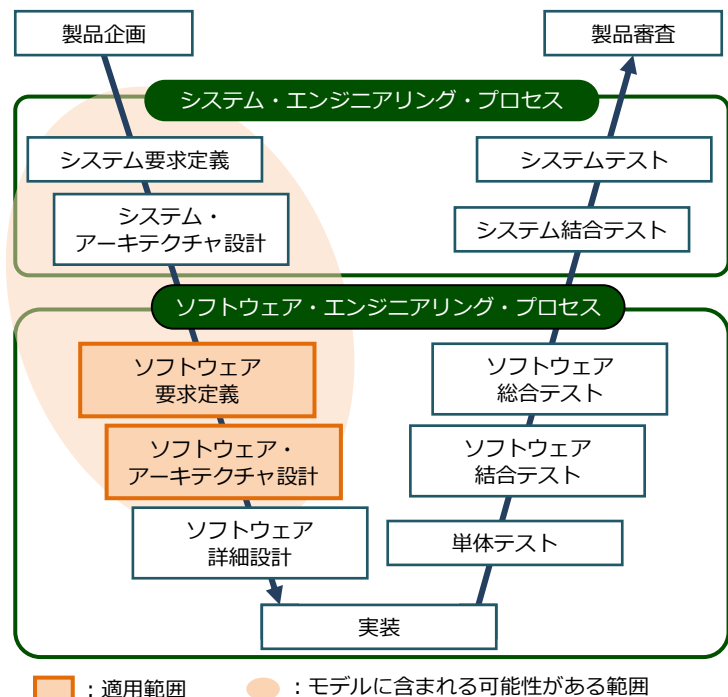


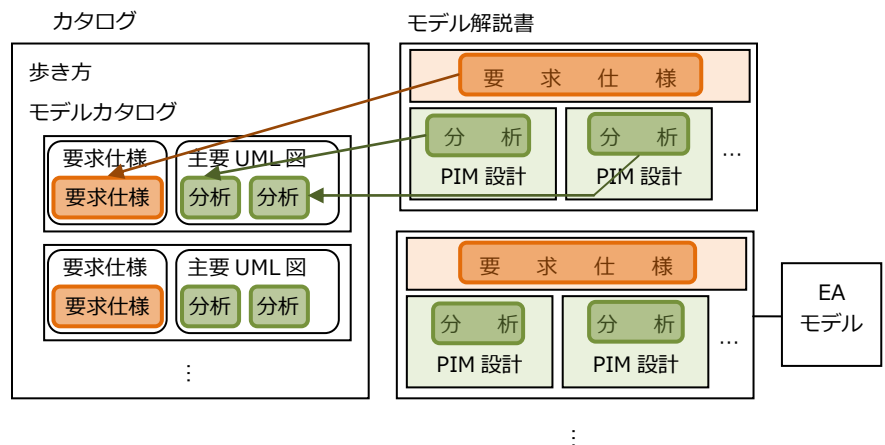
図 1 組込みソフトウェア開発プロセスと本書の適用範囲

<sup>1</sup> 『組込みソフトウェア向け開発プロセスガイド』独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編 著、翔泳社、2007

URL : <http://sec.ipa.go.jp/publish/index.html>

## カタログの構成

UMLモデルカタログは、全体を網羅した「カタログ」ファイル(本書)と、個々のモデルについて詳しく記述した「モデル解説書」ファイル(名前は"<カタログ番号>\_<対象名>.pdf"。例："F001\_Authentication.pdf")に分かれています。カタログファイルには、モデリング対象の概略と、モデル全体が見渡せる少数のモデル図を抜粋したものが含まれています。モデル解説書には、モデルについての全情報が含まれています。また、場合によっては Enterprise Architect のモデルファイルが付属していますので、併せて参考にしてください。



### 【Enterprise Architect について】

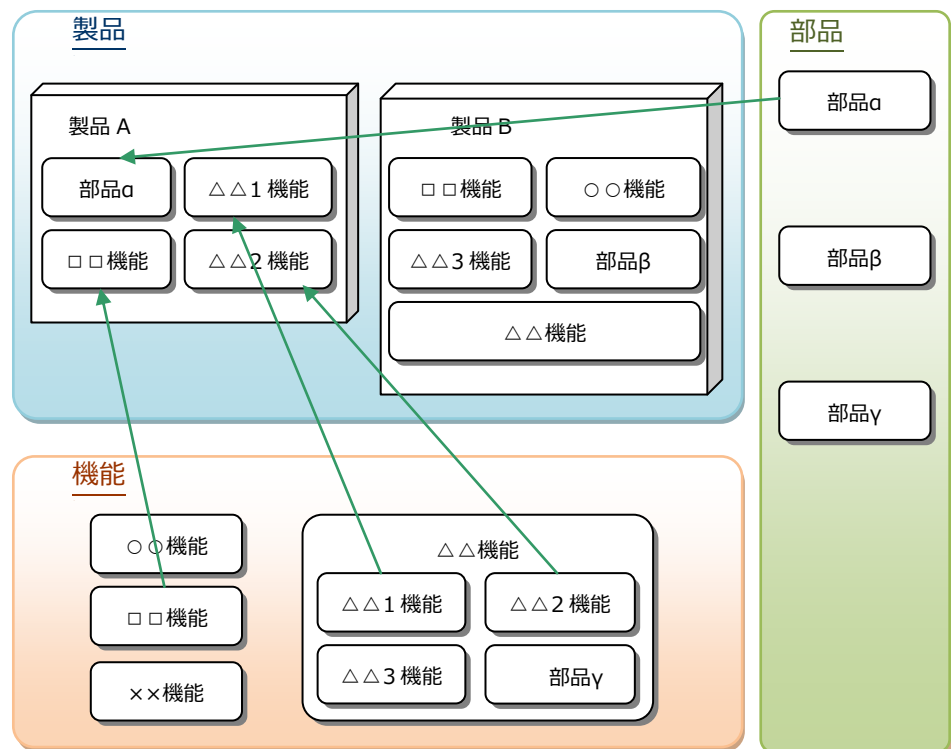
Enterprise Architect は、Windows で動作する UML モデリングツールです。組込みモデリング部会では、スパークスシステムズ ジャパン株式会社様のご厚意により、モデリングに Enterprise Architect を使用しています。モデルファイルを公開するにあたり、UMTP 専用の機能限定版 Viewer を提供頂きましたので、必要な方は UMTP サイトよりダウンロードしてお使いください。Enterprise Architect について詳しくは、<http://www.sparxsystems.jp/> を参照してください。

また、各モデルは、「製品編」、「機能編」、「部品編」に分類されています。

「製品編」は、印刷機など製品全体をモデリングして掲載しています。

「機能編」は、多くの製品に共通して搭載されている機能からいくつかをピックアップして掲載しています。「機能編」に掲載している機能をいくつか組み合わせることで、「製品編」に掲載している製品ができあがる場合もあります。

「部品編」は、より汎用的な、ライブラリとして使用できる部品を掲載しています。この部品は、「機能編」の機能でも「製品編」の製品でも、使用することができます。

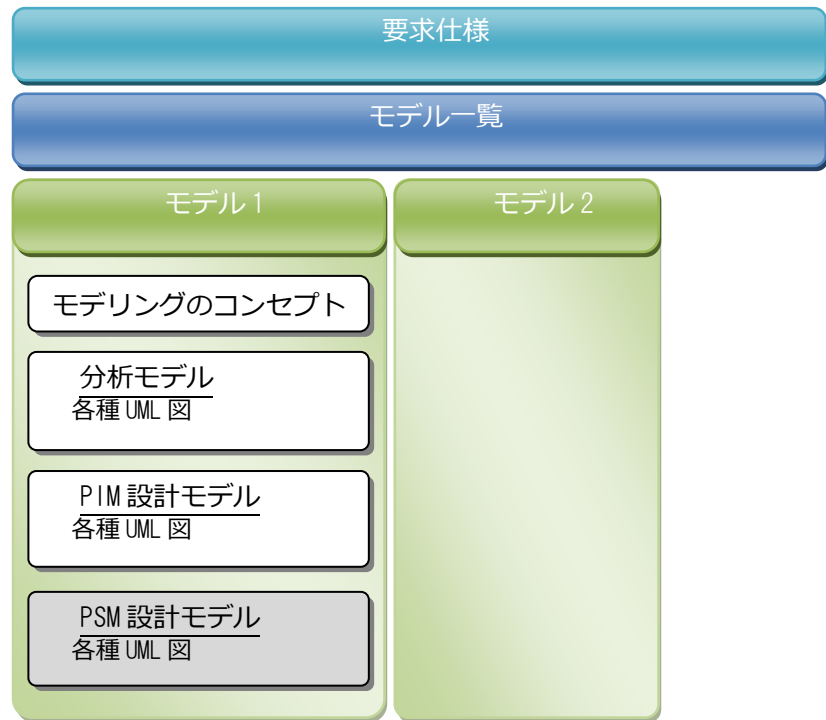


本カタログでは、まず、モデリングの対象となる製品や機能の要求仕様が定義されています。この要求仕様をもとにモデリングを行います。

本カタログでは、1つの要求仕様に対して、複数のモデルを掲載しています。どのような状況でも正解であるモデルというものはありません。さまざまな状況で、さまざまな見方があり、複数のモデルが考えられます。

要求仕様の後に、これら複数のモデルのインデックスとなるモデル一覧を用意しています。

各モデルでは、分析モデル、PIM 設計モデル、PSM 設計モデルといった具合に、段階的に詳細化を行っており、それぞれクラス図、オブジェクト図、コミュニケーション図、ステートマシン図などの UML モデルを掲載して、モデルの解説を行っています。



※PSM 設計モデルは、本書の対象外。将来の改版において含める予定。

セクション	記載内容	ESPR におけるフェーズ
要求仕様	ユースケース図・ユースケース記述・シナリオなどを使用し、製品または機能の要求仕様を定義します。	ソフトウェア要求定義
分析モデル	モデリングのコンセプトをもとに作成したモデルです。	
PIM 設計モデル	OS やミドルウェアや言語などのプラットフォームに依存しないモデルです(Platform-Independent Model)。	ソフトウェア・アーキテクチャ設計
PSM 設計モデル	プラットフォームに特化したモデルです(Platform-Specific Model)。 ただし、今回はモデルカタログの対象外です。	

【複数モデルについて】

モデリングの対象が同じでも、要求仕様の整理整頓の仕方や重点を置くポイントが異なれば、できあがるモデルは異なるものになります。たとえば、「機能」「エンティティ」「状態」などに着目してモデリングすることもできれば、「メタファ」に重点を置いてモデリングすることもできます。



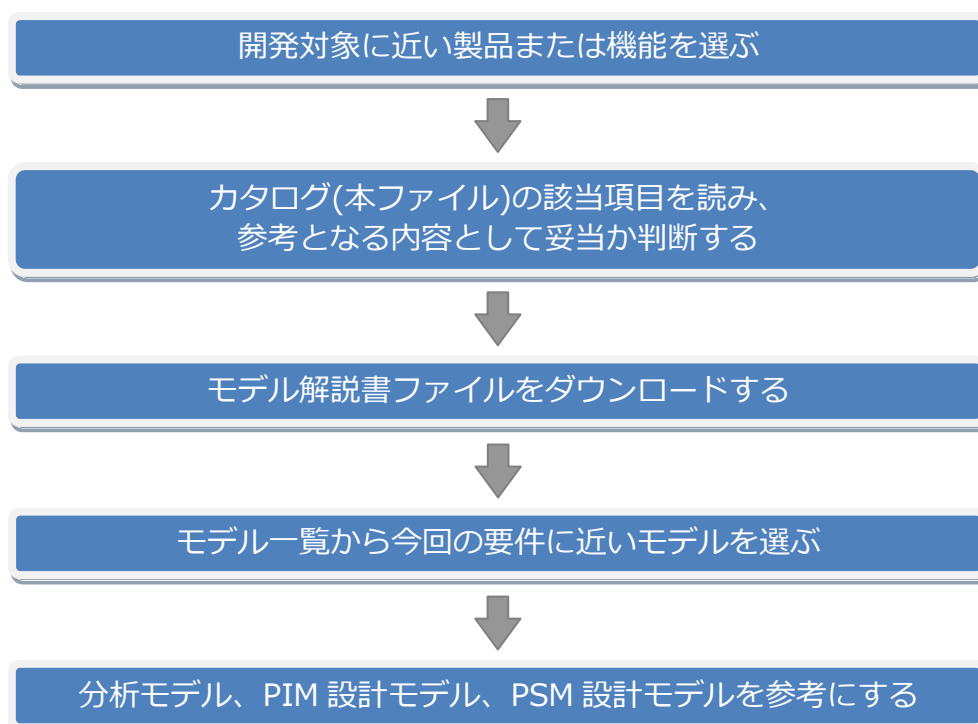
## カタログの使い方

本書では、次のような場面で利用されることを想定しています。

- ◆ 新しい製品を開発することになり、製品全体のモデリングの参考として・・・
- ◆ 既存製品に新しい機能を追加することになり、その機能のサンプルとして・・・
- ◆ 既存製品の既存機能を改善するために、比較サンプルとして・・・
- ◆ 組込みシステムにおけるモデリング教育の参考資料として・・・

### 利用例 1

新しい製品の開発や既存製品への機能追加のサンプルとして利用する場合、下記のような流れで利用します。



例えば、「コピー機」を開発する場合、製品編から「コピー機」のモデルを探しますが、見当たりません。このような場合、製品概要やキーワードからコピー機と「似た」製品を探します。その結果、「孔版印刷機」を選ぶことにします。

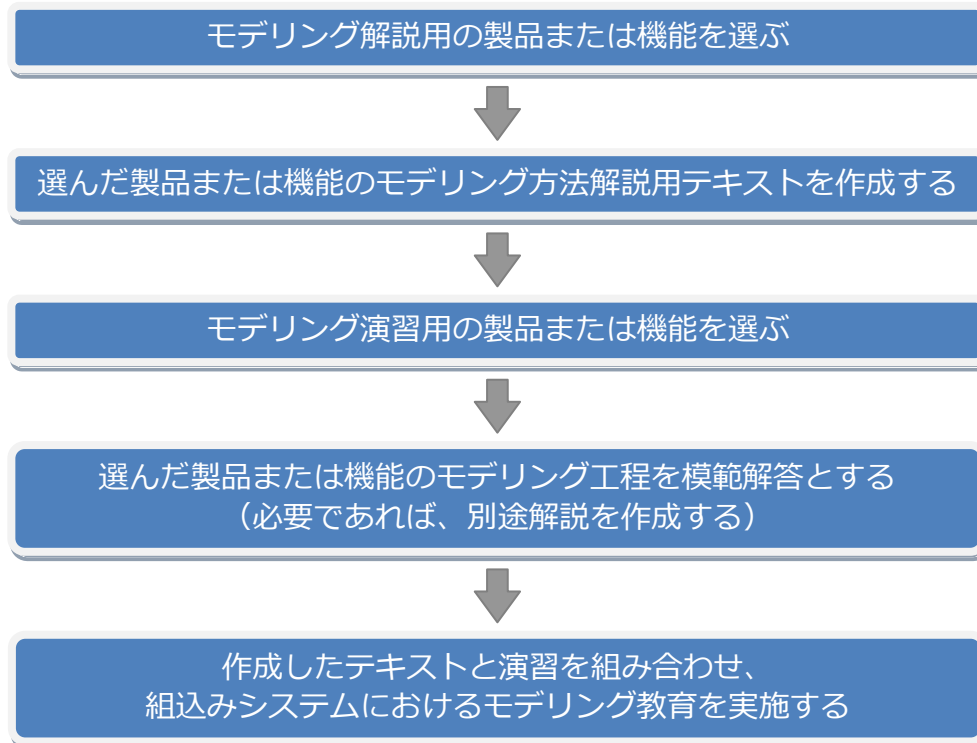
「孔版印刷機」と「コピー機」には、「読み取り」、「書き込み」、「自己診断」等の共通した機能がおり、これらがどのようにモデリングされるかを参考に開発ができます。

さらに、「自己診断」については、機能編にその詳細が記述されていますので、機能編の「自己診断」で掲載されているモデルも参考にするとよいでしょう。また、「コピー機」に必要な機能が、参考とする「孔版印刷機」にない場合、機能編を探してみると、その機能が掲載されているかもしれません。製品を構成する機能の多くは、同様の製品に搭載されていることが多く、本書においても機能編でより詳細なモデリングを行っている場合がありますので、機能編も合わせて参考にしてください。

各モデルは、全てを取り入れる必要はありません。必要な部分を取り入れて、開発しようとする製品に合わせて、うまくカスタマイズしてください。

**利用例 2**

組込みシステムにおけるモデリング教育の参考資料としては、モデリング技法の解説や、演習問題の題材として、以下のような利用ができます。



例えば、組込みソフトウェアのモデリング技法を演習付きで習得できる講座を作るとすると、以下のような流れになります。

まずモデリング解説のためのテキストを作成します。

その際に活用するモデルとして、例えば「自己診断」のモデルのうち「メタファを使って分析したモデル」を選びます。ここでは、モデルの作成手順や考えが詳細に記述してあるものから選択すると良いでしょう。モデリングを解説するテキストでは「自己診断」の「メタファを使ったモデル」に記述された本書のモデルが、どのような考えに基づいてされたのか、行間を埋めていくイメージで作成します。例えば、本書のモデルでは、静的モデルとしてのクラス図に続いてシーケンス図が記述されています。テキストでは、静的モデルとしてのクラス図を作成した後は、クラス図には表現しきれない振る舞いを、シーケンス図で表現することや、クラス図のクラスとシーケンス図のクラスの対応から、具体的にどのようにクラス図からシーケンス図が作成されたのかを解説します。

次に、演習を作成します。演習の課題として、例えば「認証」を選びます。ここでは、モデリング例が多いものを選択すると良いでしょう。

必要であれば、モデリング解説用のテキスト作成時と同様に、「認証」に対するモデリング例から模範解答を選び、解説を作成します。以上を準備として、講義を実施します。

## ラインナップ

### 製品編

名前	概要	キーワード*
孔版印刷機 (P001)	低コストに大量の印刷物を作成することを実現する印刷機	孔版印刷 読み取り 書き込み 紙搬送 機器管理 ユーザインタフェース <u>自己診断</u> <u>認証</u>
電子オルゴール (P002)	機械式オルゴールの発音操作をソフトウェア制御することで複数の曲の再生に対応できるようにしたオルゴール	リスト管理 ユーザインタフェース
インライン装置 (P003)	工場の生産ラインなどで使用される、上流から下流に向かって、製造物や検査品を搬送/処理する装置。どのような機構で搬送するか、どのような加工/検査をするかに関係なく、再利用可能なモデルとして提供しています。	自動運転 並行処理の待ち合わせ

※下線は機能編に記載があります。

### 機能編

名前	概要	キーワード
認証 (F001)	ユーザを識別し、ユーザ毎に適切なサービスを提供したり、記録を取ったりする機能	ユーザ認証 サービス実行承認 サービス利用状況監視
自己診断 (F002)	組込みシステムにおいては、システムを構成するデバイスのチェックを行い、結果をレポートする機能。チェックする項目、チェック項目の実行順序、実行する手段(自動、手動)などをモデリングしています。	装置診断

### 部品編

名前	概要	キーワード
目標制御 (C001)	制御対象の計測値が目標値となるように制御する仕組み。エアコンの温度調節や自動車のABSなど数多くの製品に組み込まれています。	数式のブラックボックス化 入力値の反映タイミング



## モデルカタログ：製品編

製品編のモデルカタログです。

## P001 孔版印刷機

ご注意：

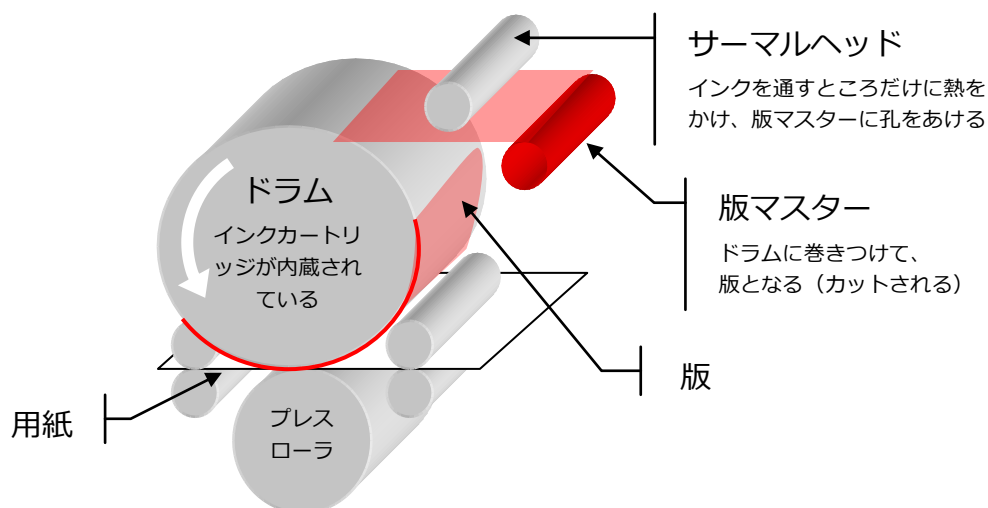
孔版印刷機の要求仕様は、理想科学様よりご提供いただきましたが、本カタログ内のモデルは、理想科学様内で使われている製品版のモデルとは一切関係ございません。また、理想科学様メンバは、本孔版印刷機のモデリング作業には、一切関わっておりません。

### 要求仕様

孔版印刷機とは、「孔版印刷」という技術を使った印刷機で、低コストに大量の印刷物を作成することを実現する機械です。

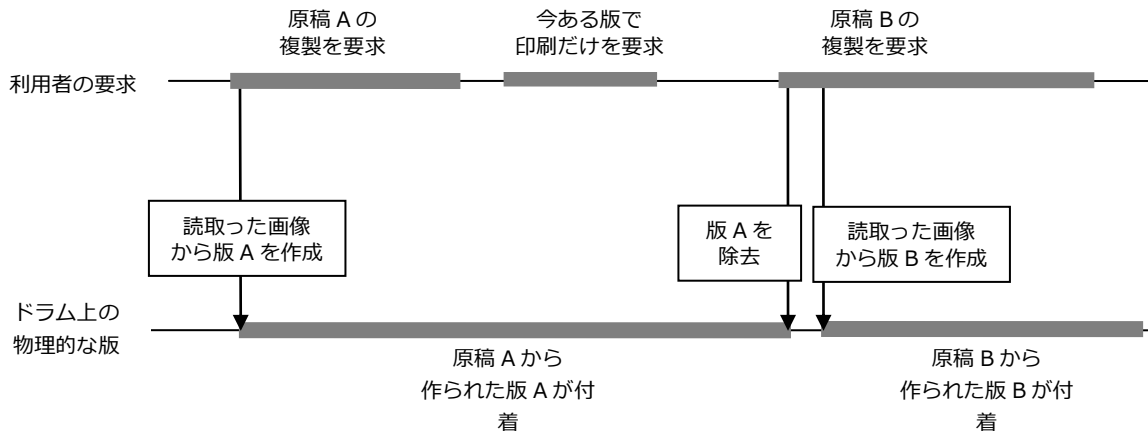
### 孔版印刷機での基本的な印刷方法

孔版印刷とは、『版』と言われるものに孔（あな）をあけ、孔があいた所にインクを通過させて、画像を用紙に転写する方法です。一度『版』を作ってしまうと、『版』が巻き付いたドラムに用紙を押し当てて通過させることで転写が完了することから、レーザープリンタ・インクジェットプリンタ・PPC方式の複写機などの比較にならないほどのスピードで印刷することができます。



### 利用者の要求と版のライフサイクル

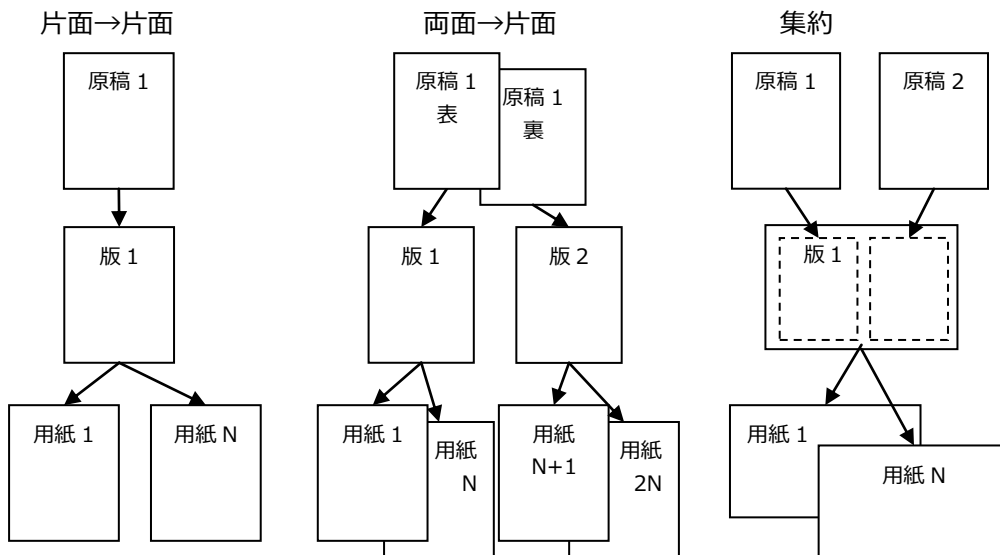
孔版印刷機では、利用者が要求する複写や印刷の単位と、その際に作成した版のライフサイクルは一致しません。例えば、一度複写するために版を作成すると、そのまま版を残しておき、別のタイミングでその版を使って印刷を行うことができます。版を除去するタイミングは、次に原稿から複製を作るとき、もしくは、利用者が版の除去を指定した時となります。



## さまざまな印刷設定

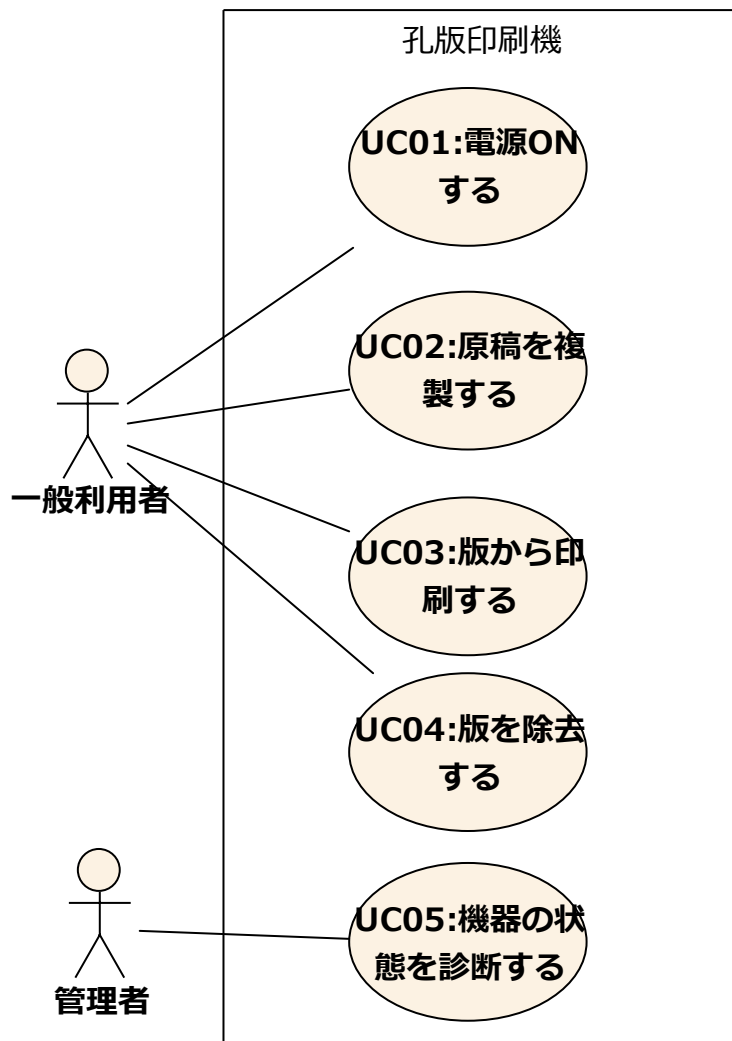
孔版印刷機の印刷機能としては、片面・両面・集約・分割・色分割（エリア指定・自動認識）などがあります。それらの機能は、ドラムが 1 つで実現できるものもあれば、ドラムを複数必要とする機能もあります。

本カタログモデルでは、ドラムが 1 つで実現できる機能にフォーカスしてモデルを作成することにします。



## ユースケース

本カタログモデルが実現する範囲のユースケースを示します。



## モデル一覧

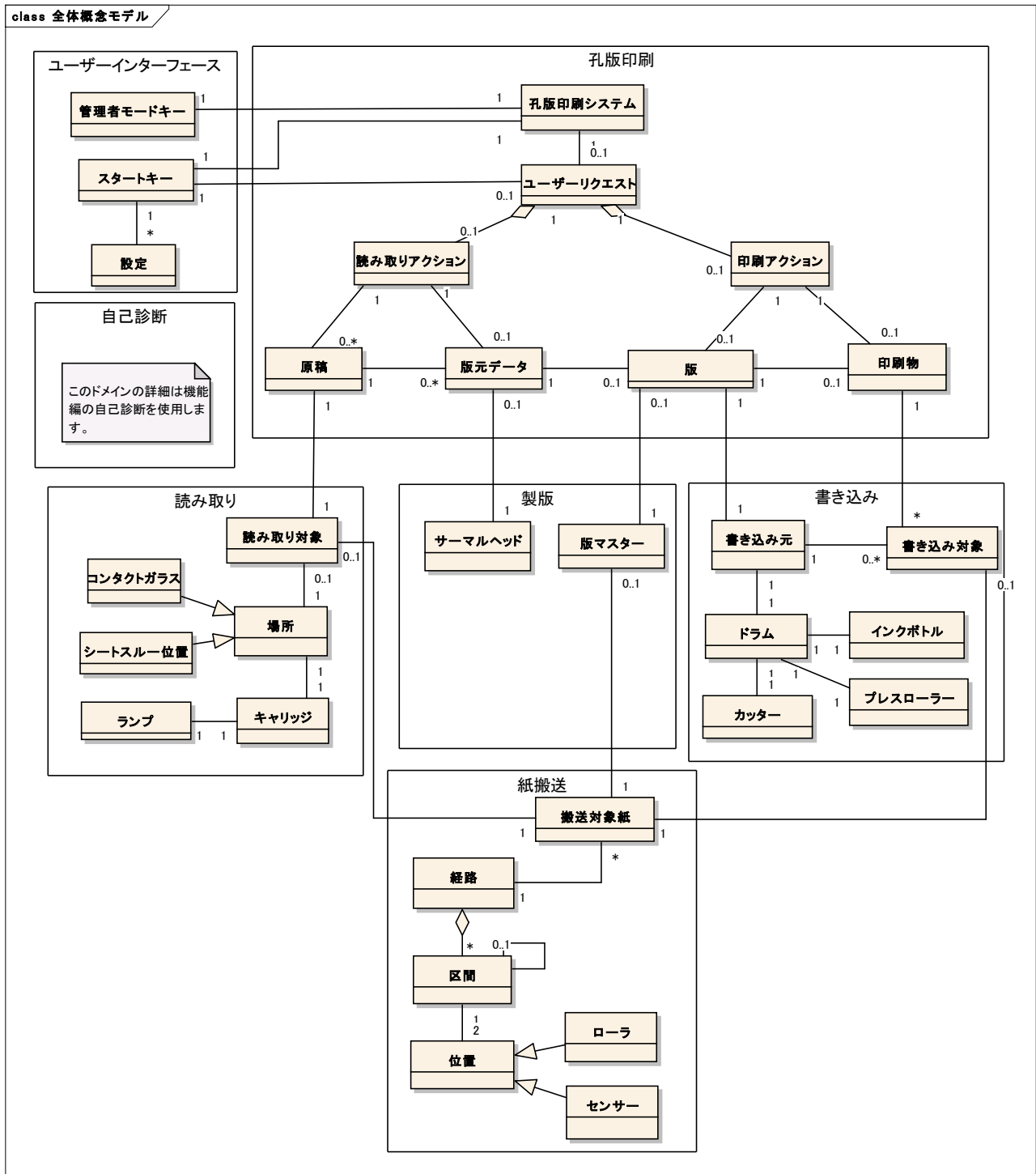
着目点	コンセプト	ポイント
機能に着目したモデル	なし	
エンティティに着目したモデル	孔版印刷機の特徴は、原稿から用紙に複写されるまでの特徴として、「版」が存在することです。それらの対応関係の構造を中心に、エンティティに着目して分析したモデルです。	いろいろな印刷設定が増えた場合に対応しやすいモデルです。
状態に着目したモデル	なし	
メタファを使ったモデル	なし	



## エンティティに着目したモデル

### モデリングのコンセプト

要求仕様で整理された原稿、版、用紙の静的な構造に着目し、これらを孔版印刷機を中心と位置づけます。その構造を中心として孔版印刷機をモデル化した全体概念モデルと、それをドメインに分類したクラス図を次に示します。孔版印刷ドメインは、主にシステム全体の状態や、利用者からの要求を取り扱うことが使命です。それ以外のドメインは、各種デバイスを制御し、孔版印刷を実現するための手段を提供する、という役割分担となっています。



## 分析モデル

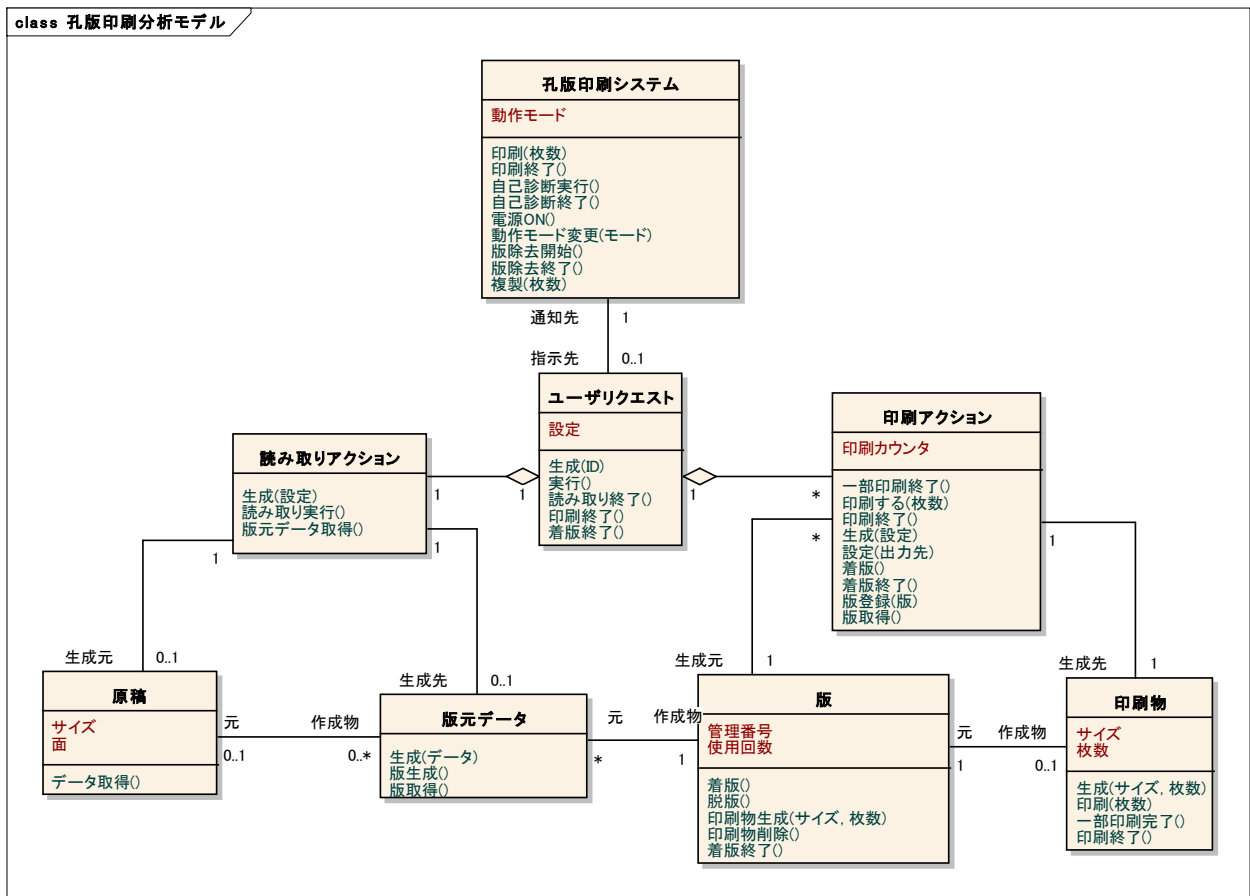
### 孔版印刷ドメイン

孔版印刷ドメインは、システム全体の最上位のドメインであり、利用者からの要求を実現することが使命です。

利用者からの要求の特徴は、要求仕様でまとめたように、原稿から複製を作成する場合と、すでにある版を使って印刷する場合の2種類があります。原稿から複製を作成する場合には、原稿を読み取って画像化し、その画像を使って版を作成し、版から印刷物を作成する、といった流れになります。すでにある版を使って印刷する場合には、版から印刷物を作成するだけとなります。

また、このドメインでは、パフォーマンスを出すために、一枚ずつ印刷結果を確認しません。印刷自体のながれは、書き込みドメインに指定した枚数分の印刷の管理をゆだねます。

### 静的モデル



## P002 電子オルゴール

### はじめに

電子オルゴールは、携帯音楽プレーヤーのような音楽再生装置で、曲やプレイリスト<sup>2</sup>を再生できます。オルゴールや音楽再生装置という身近なものを考えていますので、分かりやすいモデルです。

本モデルでは、曲やプレイリスト、再生パラメータをリストにまとめて管理する方式を表現しています。これは、データをグルーピングして管理する場合の参考になります。また、ユーザーインターフェースに関しては、設計モデルにデザインパターンを適用したモデルを提示しています。こちらは、小型の画面を持つ機器のユーザーインターフェースの設計に用いることができます。

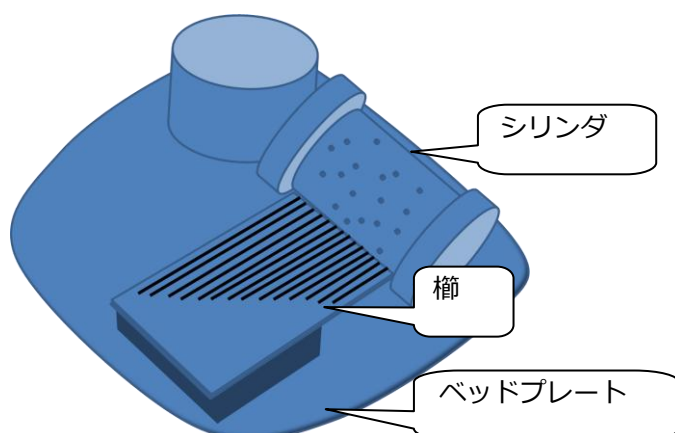
### 要求仕様

機械式オルゴールは、シリンダやディスク上のピンが、オルゴールの櫛の歯を弾くことによって音楽を奏でます。本カタログで扱う電子オルゴールは、機械式オルゴールの発音操作をソフトウェア制御します。これにより、(通常の機械式オルゴールは1曲の演奏を行います)、複数の曲の再生に対応できるオルゴールとなります。

曲を再生する機器には、携帯音楽プレーヤーがあります。電子オルゴールは、携帯音楽プレーヤーと同様のプレイリストにより複数の曲を再生します。プレイリストについて、携帯音楽プレーヤーとほぼ同様です。反面、携帯音楽プレーヤーは、テンポを変更できません(再生速度を変更すると可能ですが、音程も変わってしまいます)、電子オルゴールは、音程を変えずに、テンポを変更できます。これは、オルゴールが楽譜としてシリンダを持っており、それを元に演奏することをソフトウェアで再現しているためです。



### 機械式オルゴールの解説

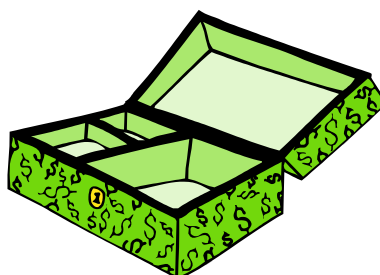


<sup>2</sup> プレイリストは、曲や動画を再生する順番に並べたリスト。携帯音楽プレーヤーやPCの音楽再生ソフトウェアでは、プレイリストを使用して、好みの順番で曲や動画を再生することができる。

シリンダ	オルゴールにとっての「楽譜」。櫛を弾くためのピンが打ち込まれている。シリンダが回転することにより櫛を弾くことで発音する。
櫛(くし)	発音するための金属板(「歯」)が並んでいる。歯の長さにより音程が決まる。
ベッドプレート	鋳物でできた土台。櫛が弾かれた際の音をボックスに伝える。
ボックス	通常は木製の箱。ベッドプレートの振動を反響させる共鳴装置として機能する。

## ハードウェア仕様

1. 小型の木製の箱である（発音装置との組み合わせでオルゴールを再現できるもの）
2. 駆動
  - バッテリで駆動すること
  - AC アダプタ接続により駆動すること
3. インターフェース
  - 小型液晶ディスプレイ
  - 上/下/左/右/中央の 5 個のキー
  - SD-Card
4. 音源方式
  - ウェーブテーブル SW シンセ



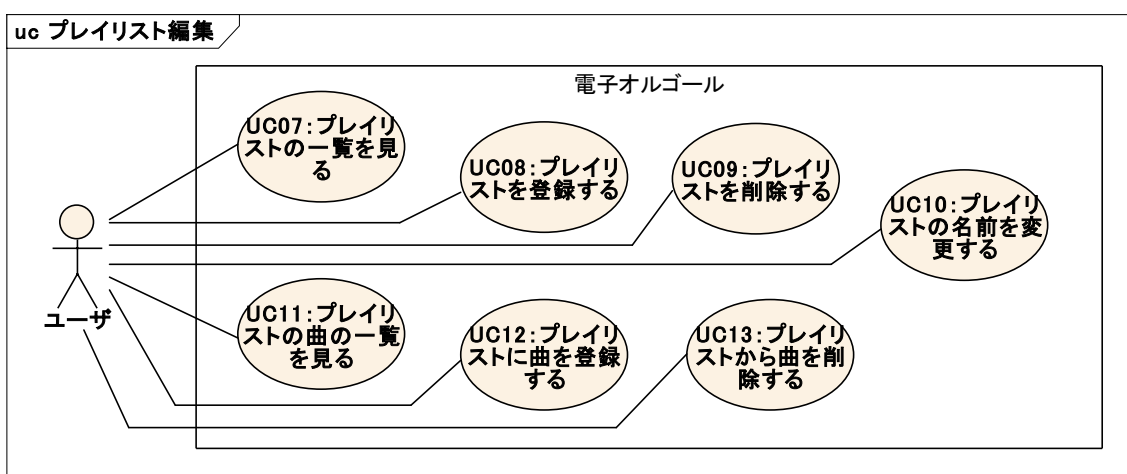
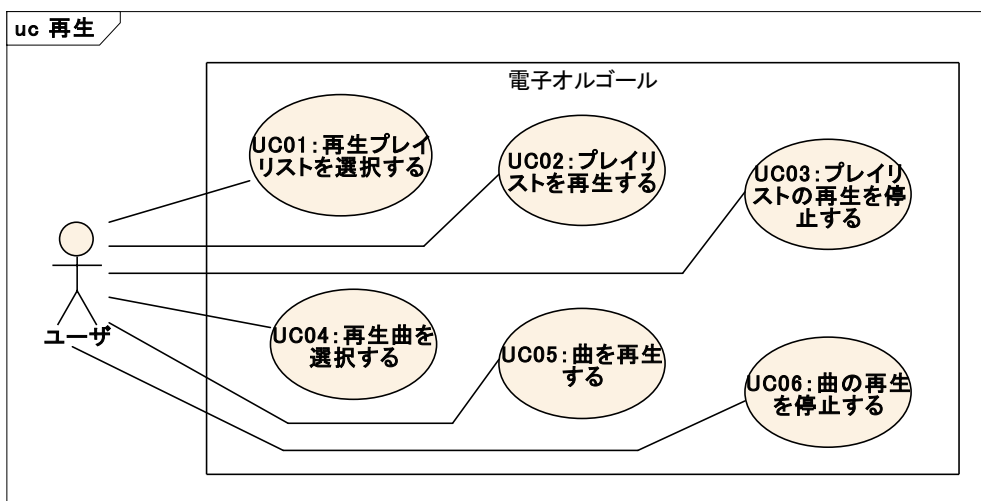
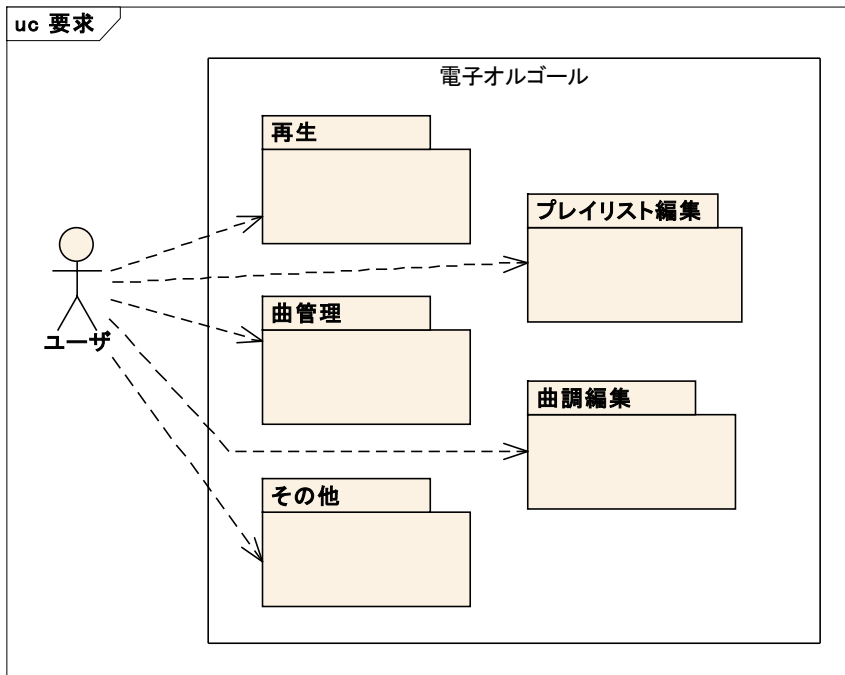
## 電子オルゴールの基本的な機能

1. プレイリストに曲を登録し、プレイリスト内の曲を再生していくことができます。
2. 調律<sup>3</sup>、リヴァーブ<sup>4</sup>、音量の設定を「曲調」として登録しておくことができます。

<sup>3</sup> 調律は、音階(ドレミファソラシド)の音の高さ(周波数)の決め方。現在、1 オクターブを 12 等分した 12 平均律が一般的であるが、純正律、ミントーンなど、色々な方式が存在する。

<sup>4</sup> リヴァーブは、音の響き方。英語 reverberation(残響)より。同じ楽器の音でも、コンサートホールの響きや小さな部屋の響きなど、環境により音が異なる。この場合は、演算処理によりコンサートホールや小さな部屋の残響を再現することを示す。

## ユースケース



## モデル一覧

モデル名	概要	ポイント
機能に着目したモデル	なし	
エンティティに着目したモデル	要求仕様の内容をエンティティ中心に分析し、オルゴールの部分と曲を順番に再生する音楽再生装置の2面から考えました。	オルゴールや音楽再生装置という身近なものを考えていますので、分かりやすいモデルです。 また、応用例も考えやすいと思います。
状態に着目したモデル	なし	
メタファを使ったモデル	なし	

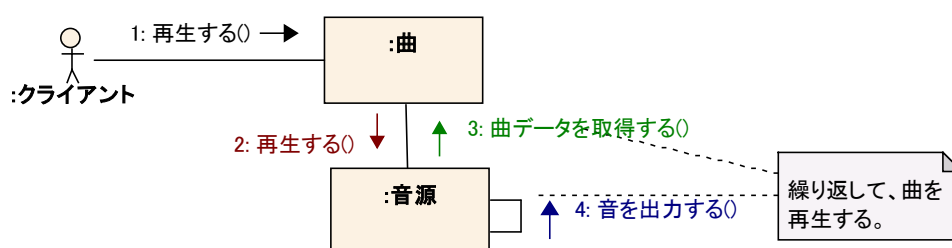
## エンティティに着目したモデル

### モデリングのコンセプト

(1) 電子オルゴールをモデリングするにあたり、機械式オルゴールを以下のようにとらえなおしています。

機械式オルゴール	電子オルゴール
シリンダ :	曲 : 機械式オルゴールのシリンダを抽象化している。
櫛 ベッドプレート ボックス	音源 : 機械式オルゴールの櫛、ベッドプレート、ボックスに対応している。

シリンダを曲、櫛などを音源ととらえ、曲と音源が協調して、曲を再生します。



(2) 「電子オルゴール」を一般的な「曲を順番に再生する音楽再生装置」として捉えます。

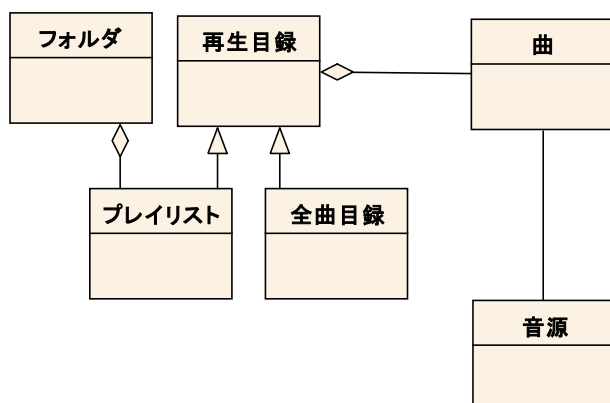
(2a) プレイリストは曲を順番に再生するので、プレイリストは曲を集約します。

(2b) 一般的に音楽再生装置のメニューはフォルダのような階層があり、プレイリストを選択します(フォルダはフォルダまたはプレイリストを集約します)。電子オルゴールには階層はなく、フォルダはプレイリストを集約します。

(2c) 電子オルゴールのすべての曲を管理するものが必要で、全曲目録とします。

(2d) プレイリストと全曲目録は、曲を集約し、再生する責務を持つことが共通しています。共通部分を再生目録として汎化させます。

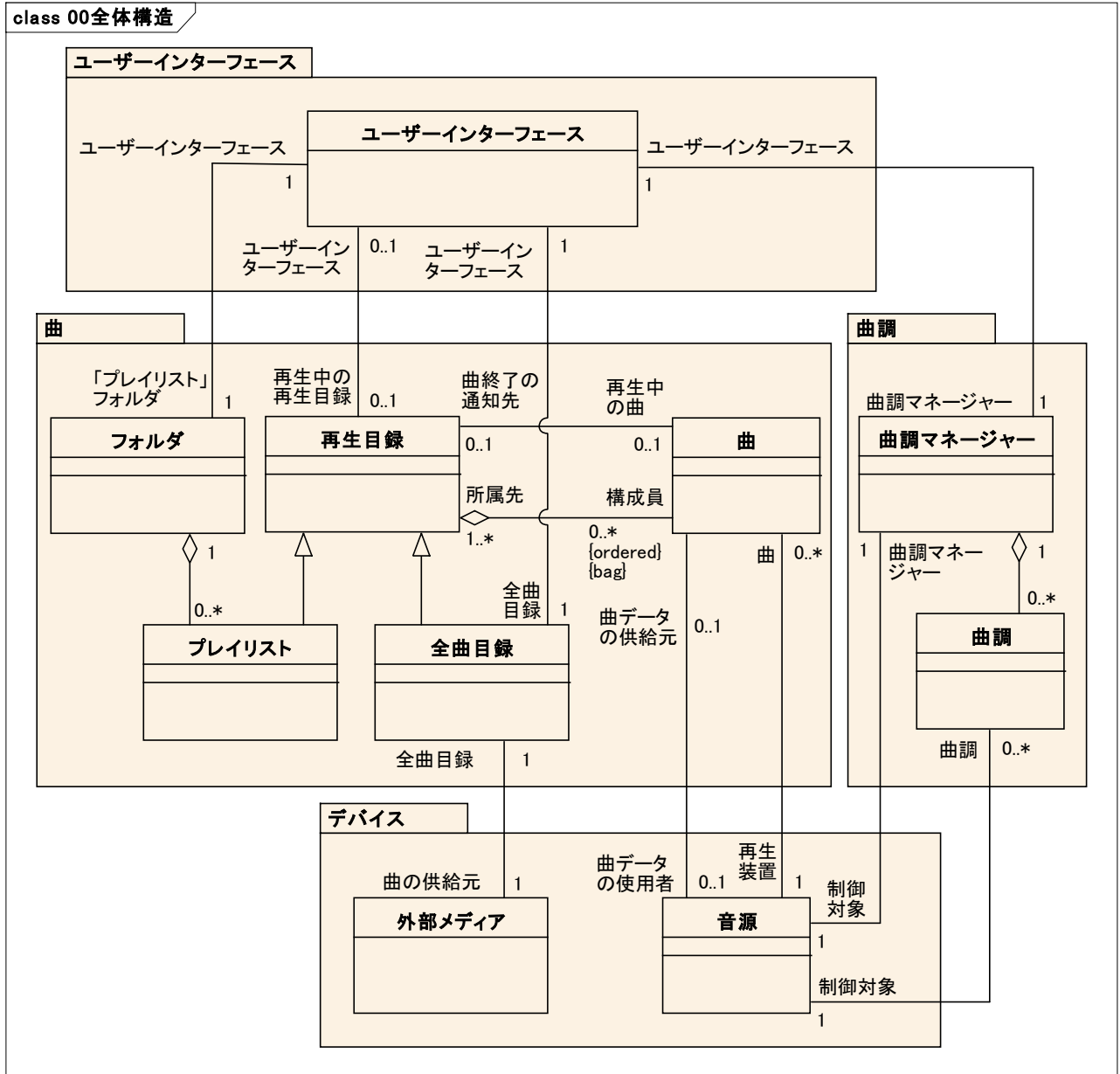
以上をまとめた、電子オルゴールの主要なクラスを次に示します。



## 分析モデル

前述の電子オルゴールの主要なクラスに、ユースケースの実行に必要なユーザーインターフェース、曲調、曲調マネージャー、外部メディアを追加し、分析モデルを作成しました。また、クラス数が増えたため、ユーザーインターフェース、曲、曲調、デバイスのパッケージに分割しました。

### 電子オルゴールの全体構造



曲パッケージは、曲やプレイリストなど、曲を連続して再生するという音楽再生装置として主要な部分になります。Ordered と bag については後述します。

ユーザーインターフェースパッケージはユーザとのインターフェースを司ります。

曲調パッケージは、本カタログの要求仕様の曲調を実現する部分です。要求仕様により内容が大きく変動する部分です。

デバイスパッケージは、電子オルゴールのデバイスをまとめたものです。



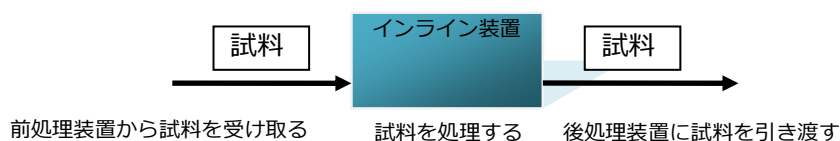
## P003 インライン装置

「インライン装置」は、工場の生産ラインなどで使用される上流から下流に向かって、製造物や検査品を搬送/処理する装置のことです。どのような機構で搬送するか、どのような加工/検査をするかに関係なく、再利用可能なモデルとして提供しています。また、自動運転や並行処理の待ち合わせといった装置開発でよくみられる課題を扱っています。

このモデル解説書では「インライン装置」と呼ばれる装置を、

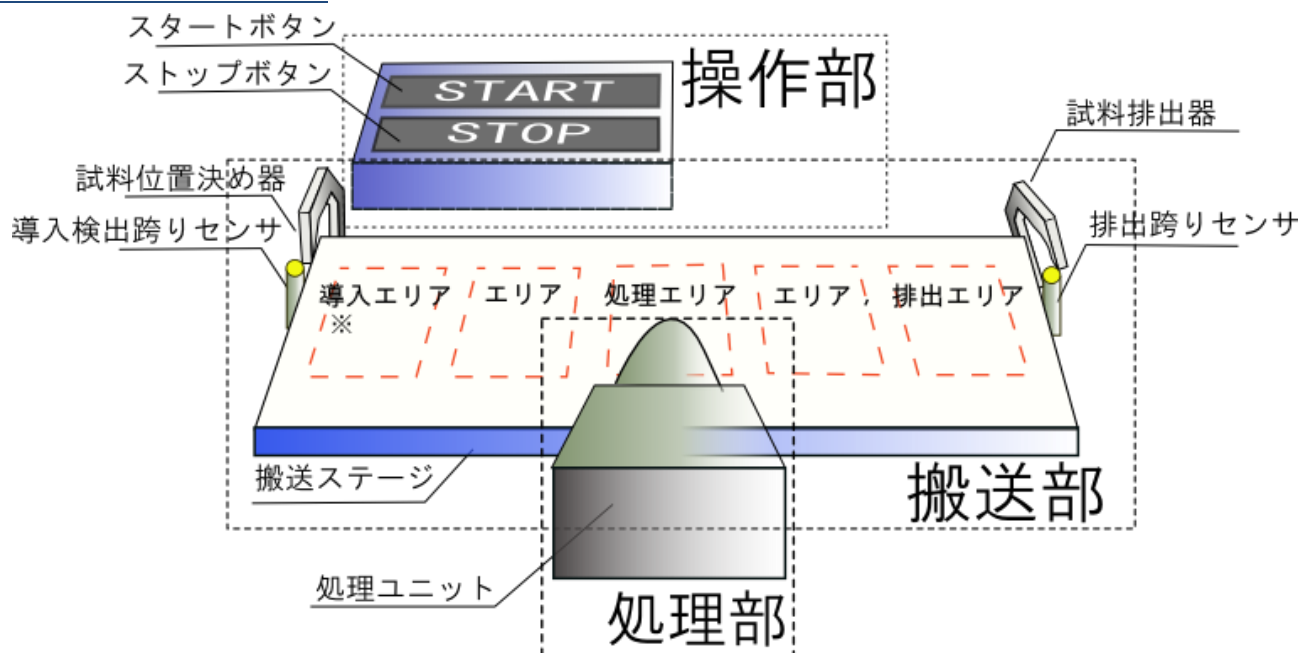
1. 前の装置から引き渡された「何か（試料）」を、
2. なんらか処理して、
3. 後ろの装置に引き渡す。

のステップでとらえ、汎用的な要求仕様として、提示しています。



## 要求仕様

### 論理ハードウェア仕様



※ 図中の「XXエリア」は、試料が搬送されるときに、一回一回止まる場所を示す概念。たとえば、搬送ステージ上で試料だけを移動する機構ならば試料が一回一回の搬送される位置のことであり、たとえば、搬送ステージごと試料を移動するコンベアのような機能ならば、コンベアが一回一回の搬送で止まる位置のこと。

論理ブロック	構成要素	説明
操作部	スタートボタン	処理部に対しては処理の開始、搬送部に対しては前処理装置からの試料の受付を開始する。
	ストップボタン	処理部に対しては処理の停止、搬送部に対しては前処理装置からの試料の受付を停止する。

処理部	処理ユニット	試料に対する処理を行う。たとえば、加工や検査など。
搬送部	搬送ステージ	試料を搬送する領域全体の部分。
	導入検出跨りセンサ	「前処理装置から受け取った試料が存在すること」・ 「前処理装置から受け取り終わった試料を取り損なっていないこと」を確認する。
	導入エリア	前処理装置から受け取った試料が一時的に置かれ、位置決めを行う場所。
	試料位置決め器	試料を位置決めする機構。
	エリア	試料が搬送される時に、一回一回止まる場所。搬送ステージ全体が長い場合は、この普通の「エリア」が複数連続して並ぶ。
	処理エリア	搬送ステージの内、試料に対し処理を行う場所のこと。
	排出エリア	試料を後処理装置へ排出する場所。
	試料排出器	試料を後処理装置へ排出する機構。
	排出跨りセンサ	後処理装置に排出し終わった試料が、排出し損なっていないかを確認する。

## 機能要求

### ◆ 処理開始機能

操作部のスタートボタンを押下すると、装置状態が運転モードになり、自動処理を開始します。

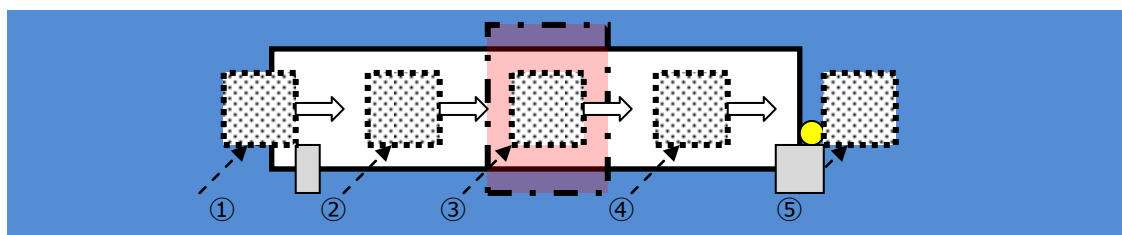
### ◆ 処理停止機能

操作部のストップボタンを押下すると、自動処理を停止し、装置状態が停止モードになります。

### ◆ 試料搬送機能

運転モードのときに、前処理装置から搬送部へ試料が導入されると、連続運転の実動作を開始します。

## 連続運転の動作仕様

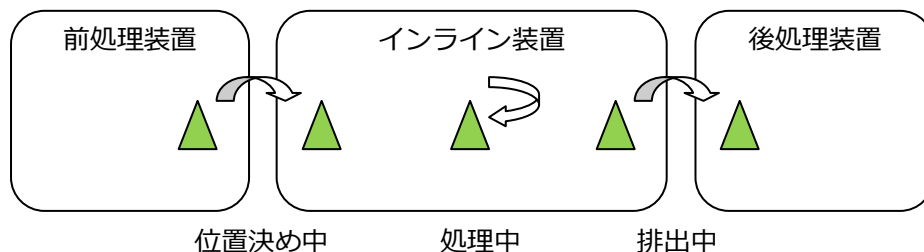


- ① 前処理装置から引き渡された試料は、試料位置決め器によって、処理エリアから決まった距離になるように位置決めをします。この時、導入検出跨りセンサによって、前処理装置から導入されているかを確認します。
- ② 位置決めをした所から、一定の間隔で搬送します。この時、導入検出跨りセンサによって、前処理装置から搬送ステージ内に確実に引きこんだかを確認します。
- ③ 処理エリアまで搬送します。処理エリアでは、処理部によって試料を処理します。
- ④ 次の試料を処理エリアに運ぶのと同時に、処理済みの試料を後処理装置に向かって搬送します。
- ⑤ 試料排出器によって試料を後処理装置へ排出後、排出跨りセンサで、引き渡しができているかを確認します。（払い出し部の一例：）ただし、後処理装置が排出を受け付けられない場合は、受け付けられるようになるまで待つてから排出をします。

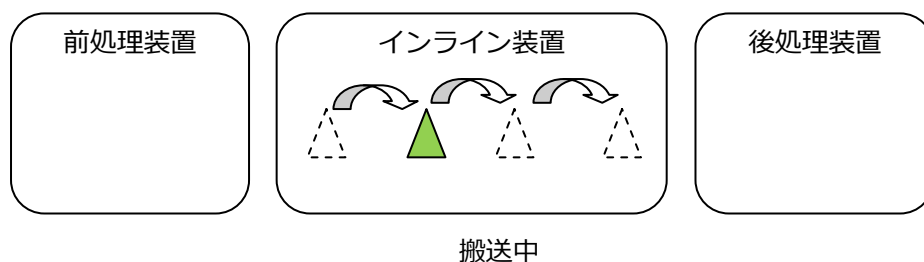
※ 以上の①～③と③～⑤の間が何回で搬送されるかは、搬送ステージの長さや試料の長さによって、さまざまなパターンが考えられます。上記の例は、最小限を考えた一例です。

## 搬送部の各動作と処理部の動作の並行タイミングの要求

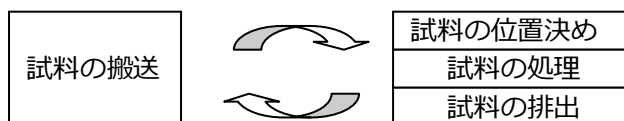
インライン装置は、試料の位置決め、試料の処理、試料の排出を並行して行う事が出来ます。



試料の搬送は、試料の位置決め、試料の処理、試料の搬出を行っていない場合に行う事が出来ます。



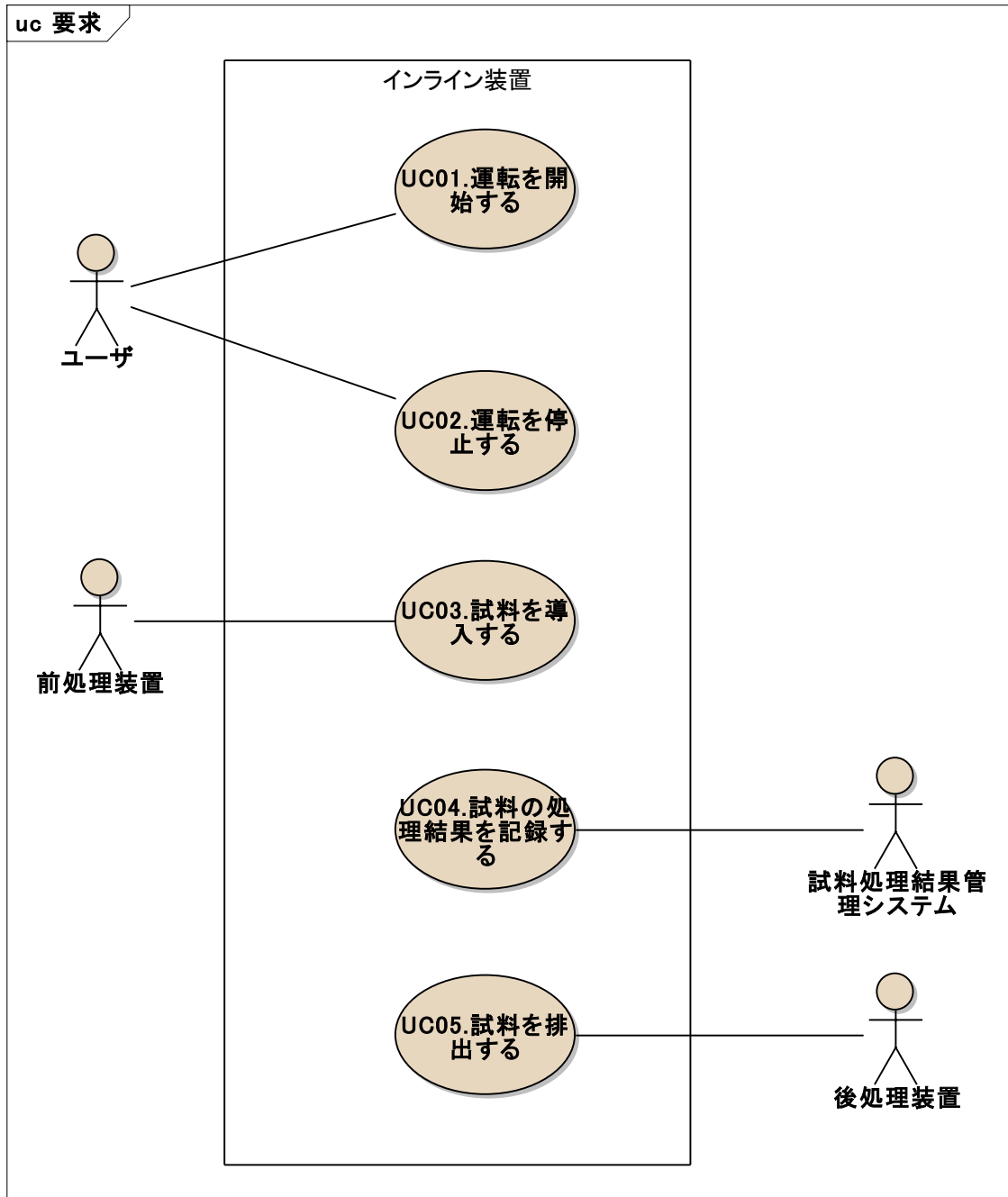
上記二つの状態を交互に繰り返すことによってインライン装置の動作を実現します。



このため、試料の搬送が完了を待って試料の位置決め、試料の処理、試料の搬出を実行する必要があります。また、反対に、試料の位置決め、試料の処理、試料の搬出の全ての完了を待って試料の搬送を実行する必要があります。

## ユースケース

本カタログモデルが実現する範囲のユースケースを示します。



## モデル一覧

モデル名	概要	ポイント
ドメイン構造に着目したモデル	インライン装置の構造を中心に分析し、機能もその構造の要素自身にすべて配置しました。搬送部や処理部を（派生などで）拡張することにより、さまざまなインライン装置に再利用できるようにしています。	自動運転／並行処理の待ち合わせといった一般的な課題を扱っています。

## ドメイン構造に着目したモデル

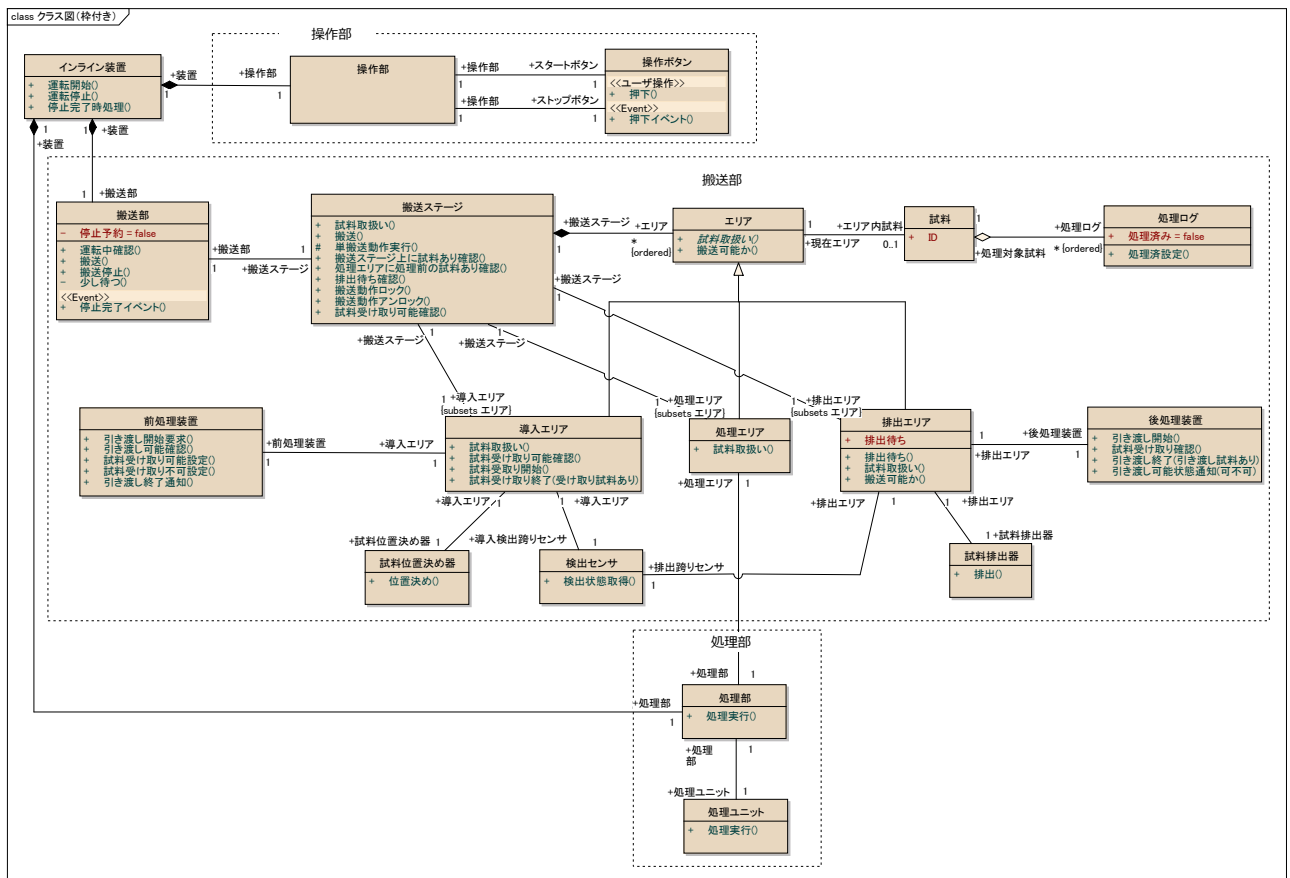
### モデリングのコンセプト

このモデルは、管理クラスや制御クラスといったクラスを出さずに、要求仕様からのモノの構造のみに着目してモデル化しています。

またモデルを利用する際に、オブジェクト同士の協調に集中してもらうため、（オブジェクト同士の境界線を引いてしまうような）パッケージ分けをしていません。

### 分析モデル

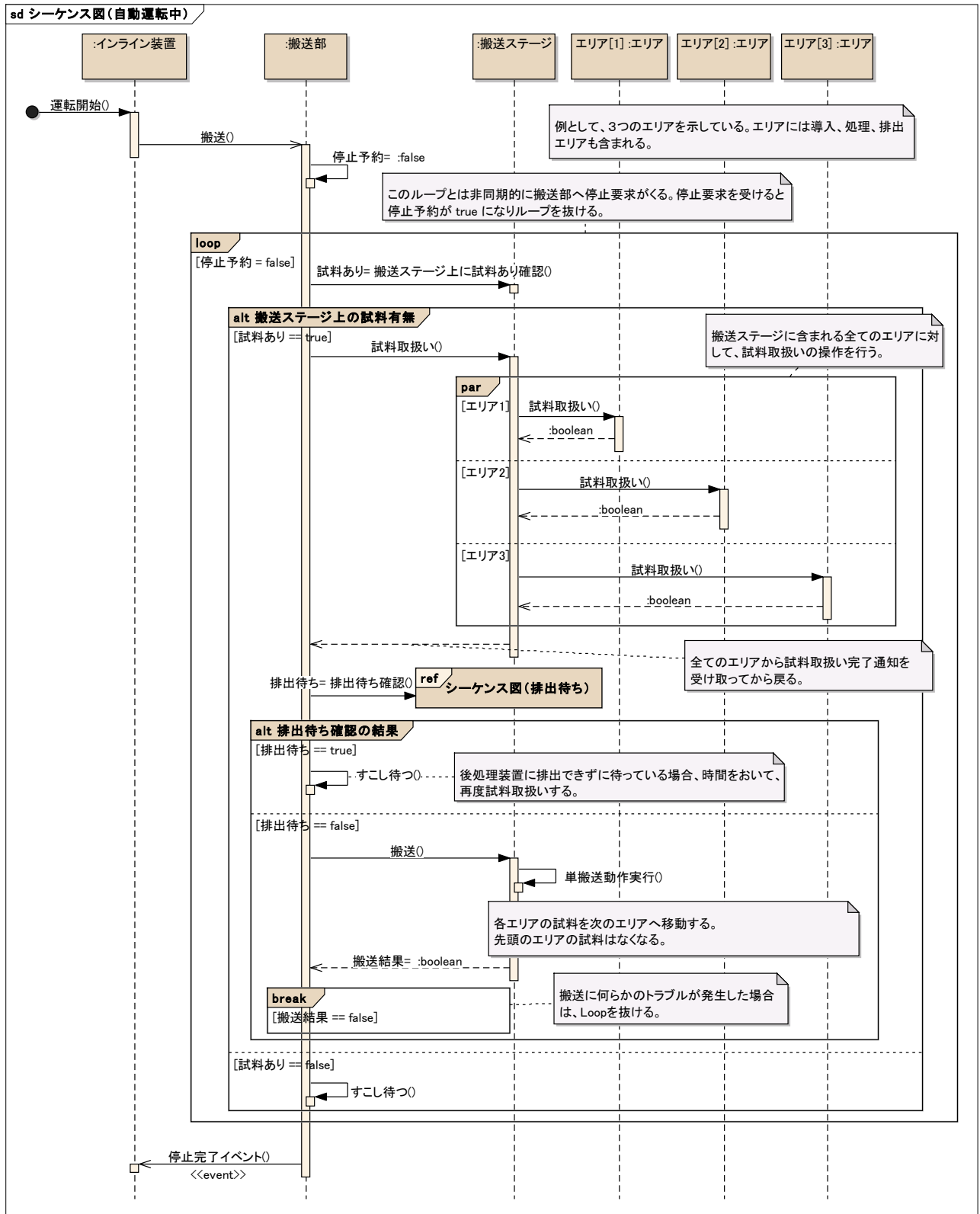
#### インライン装置システム全体のクラス構成



点線の枠は、論理ハードウェア仕様の論理ブロックの範囲を示しています。

搬送ステージはエリアへの関連を持たせました。導入エリア、処理エリア、排出エリアはエリアから派生しているため、搬送ステージはこれら3種類のエリアを統一的に扱うことができます。しかし、搬送ステージからそれぞれ個別の特性も扱う必要があります。そこで、搬送ステージには、導入エリア、処理エリア、排出エリアへの関連も持たせました。また、これらの関連に{subsets エリア}を指定することによって、関連先の要素がエリアへの関連先要素に含まれていることを明確にしています。

連続運転の相互作用





## モデルカタログ：機能編

機能編のモデルカタログです。

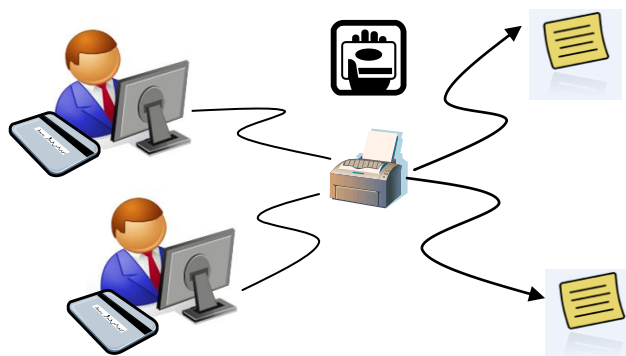


## F001 認証

### 要求仕様

#### 認証例

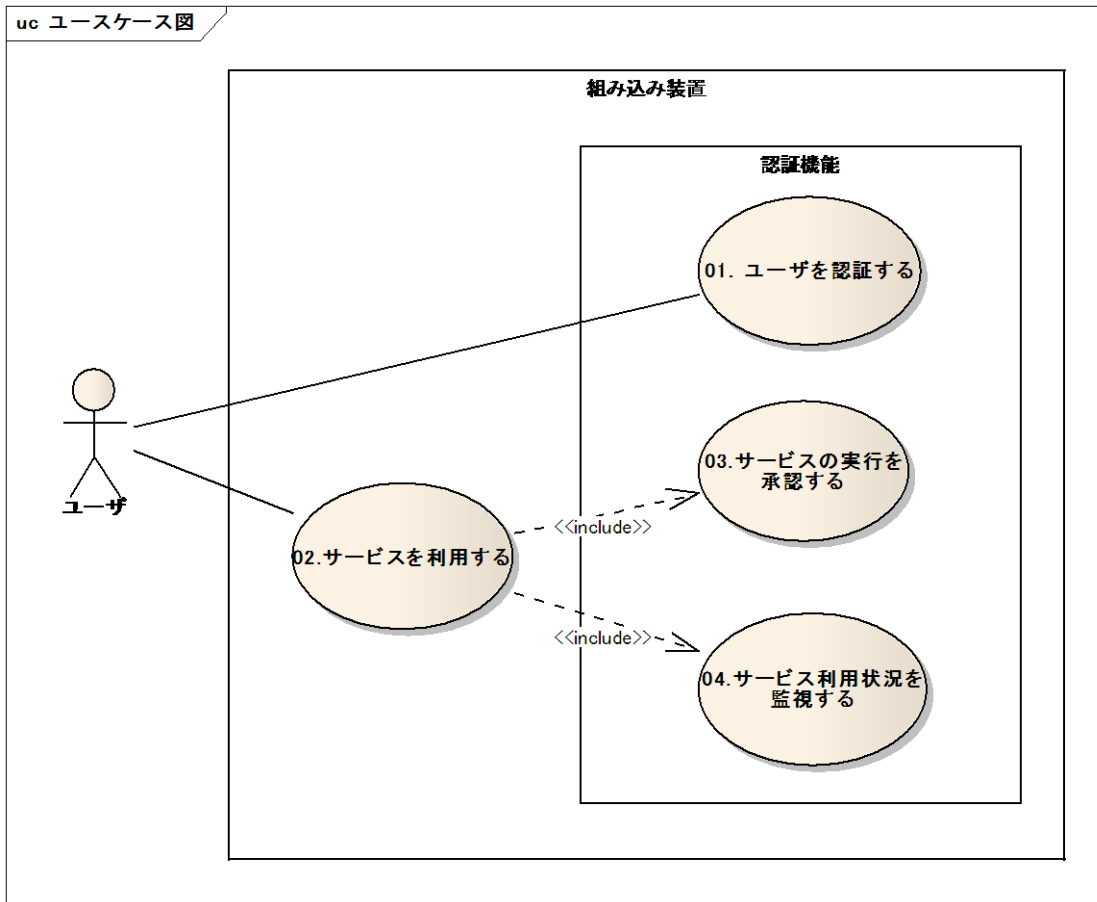
ユーザ A とユーザ B がいます。二人はそれぞれ固有の ID カードを持っています。この 2 人が 2 台の PC からそれぞれの電子データを 1 台のプリンタに印刷指示を出します。ユーザ A がプリンタに ID カードをかざすと、ユーザ A が印刷指示した電子データが呼び出され、プリントが始まります。ユーザ A の電子データの印刷処理終了後、ユーザ B が同様に ID カードをプリンタにかざすと、ユーザ B の電子データが呼び出されプリントが始まります。ユーザ A,B の使用日時や電子データのタイトル、プリント枚数など使用記録がプリンタ保存されます。



このようなユーザごとに適切なサービスを提供できる仕組みを【認証】と呼称します。

この【認証】では、ユーザを識別し、ユーザごとに適切なサービスを提供し、機器の使用記録を取りま  
す。この一連の機能の設計モデルを提供します。

## ユースケース



## モデル一覧

着目点	コンセプト	ポイント
機能に着目したモデル	認証機能の設計として、広く知られている AAA モデルの 3 機能に着目したモデルです。	AAA モデルの 3 要素を基本クラスとしたモデルです。なんらかの指針を使った設計例として利用してください。
エンティティに着目したモデル	認証がユーザのサービス使用に対して制限を掛けるという観点から、制限を掛けるエンティティに着目したモデルです。	エンティティを中心に理解しやすいシンプルな作りにしたので、モデリング初心者の方にお勧めです。
状態に着目したモデル	ユーザが認証された・認証されていない、といった状態に着目したモデルです。	状態毎の振舞いの違いを、ステートパターンで実現します。
メタファを使ったモデル	なし	

### AAA モデルとは

本章には、AAA モデルの用語が 3 つの設計モデルで共通概念として使用されています。

ここで、AAA モデルの概要を説明します。

AAA モデルとは、認証機能を実現する上で、主だった機能をつぎの 3 要素に分割したモデルを指します。

ユーザを認証する、適切なサービスを提供する、サービス使用の記録を作成する、これらを認証 (Authentication)、承認 (Authorization)、アカウントリング (Accounting) と分類し、3 つの頭文字 A をつなげて AAA モデルと呼ばれます。

また AAA モデルは、AAA プロトコルとも言われ、認証、承認、アカウントリングの 3 要素自体と、またその要素間の通信とにおいて秘匿性が設計上考慮されていることを指します。

本書に於いて、AAA のアカウントリングの呼称を、監視と改名しています。これはアカウントリングがユーザ情報 (アカウント) と誤解しやすいためです。

## 機能に着目したモデル

### モデリングのコンセプト

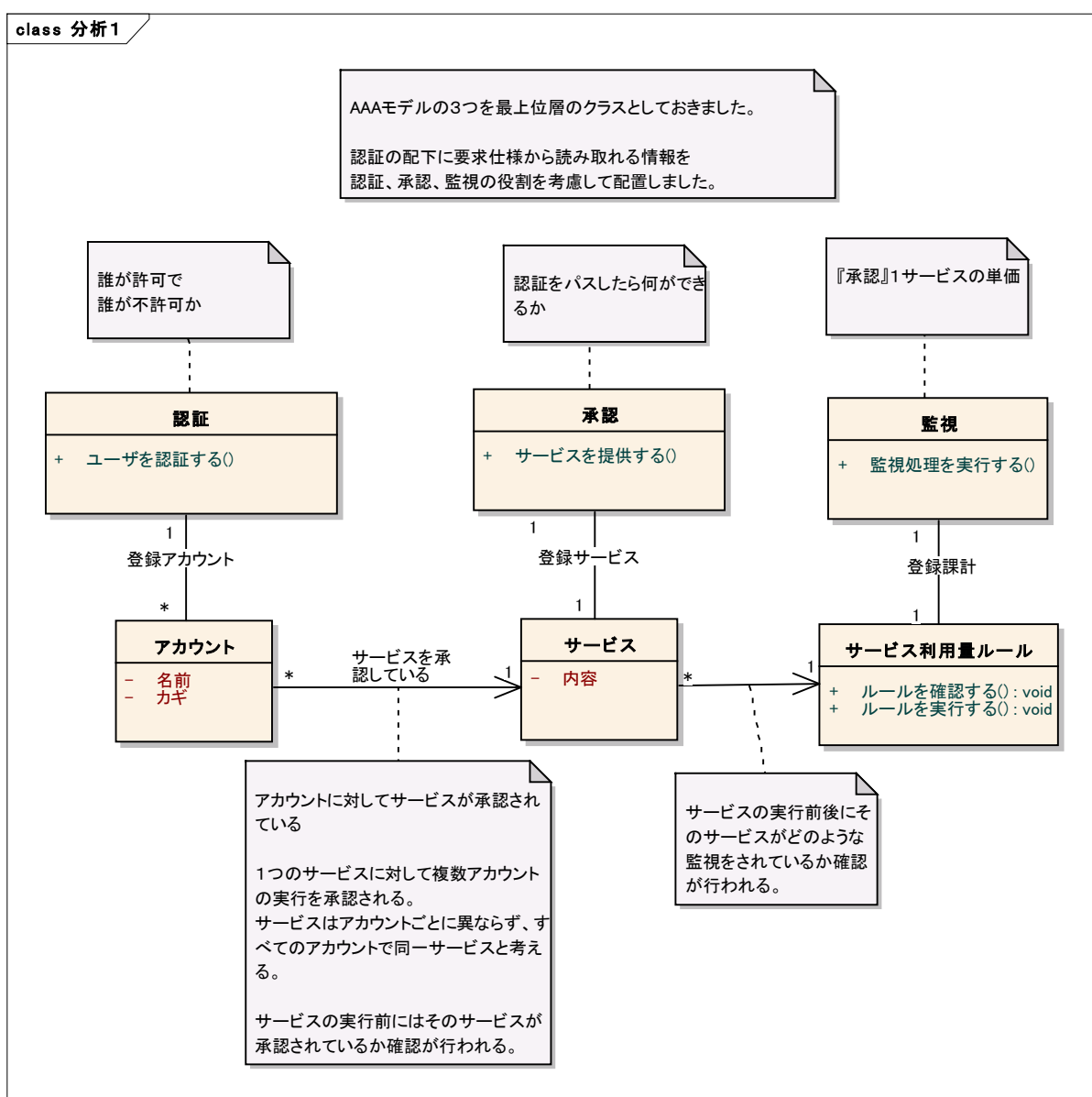
本モデルは、広く知られているAAAモデルの構造を模す事を出発点としました。AAAに登場する認証、承認、アカウントの3要素が取り扱う情報の責務に着目することを本モデルのコンセプトとして設計を行いました。

3要素のほかに要求仕様から読み取れる、ユーザのIDカード、ユーザに提供するサービス、ユーザの使用履歴といった情報を抽出し、それを元に分析作業を行いました。

### 分析モデル

#### 静的モデル

##### クラス構成その1



最上段にはAAAの3要素、認証、承認、監視を配置しました。

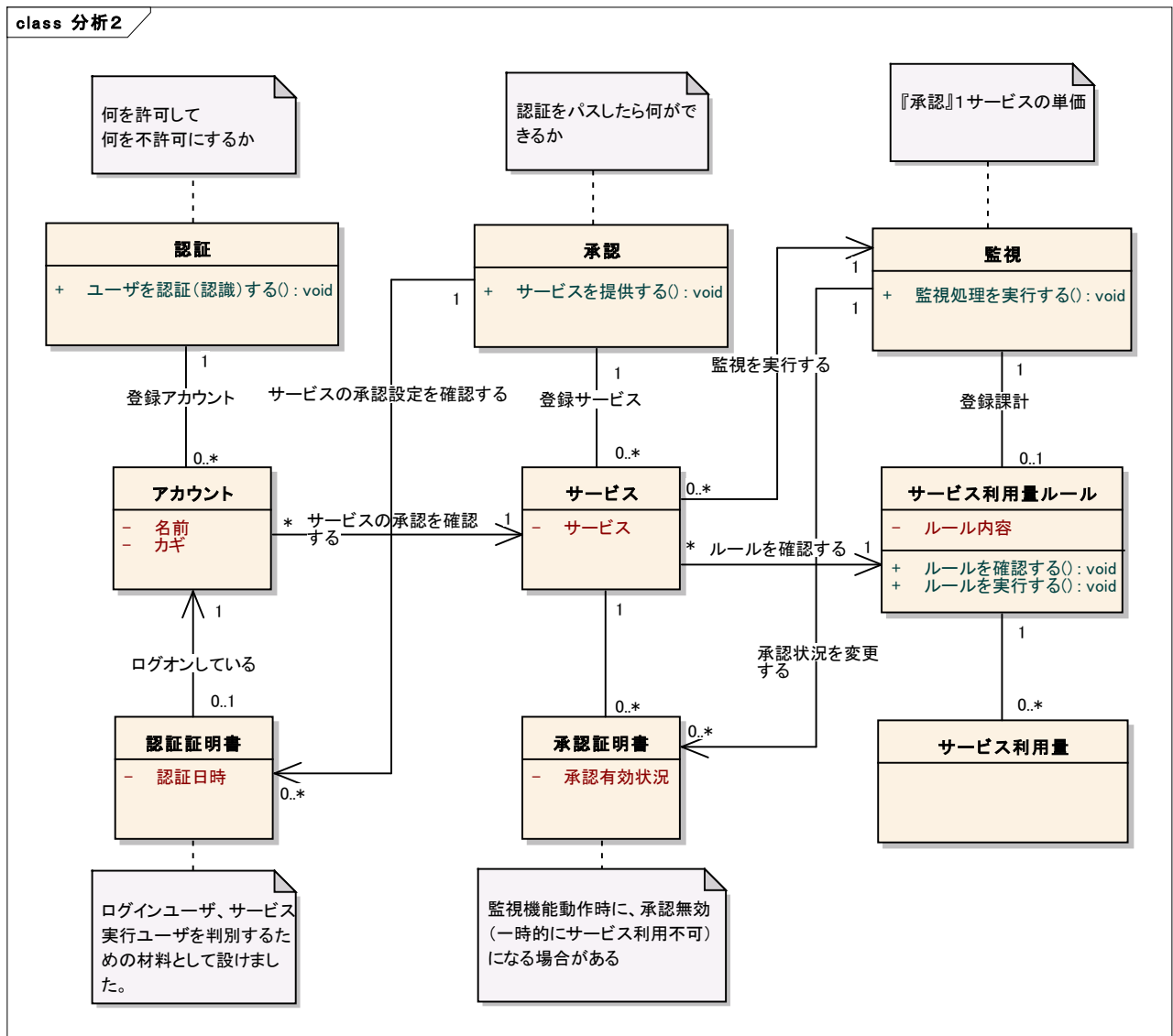
認証が扱う情報は、ユーザを識別するための情報です。認証が扱うユーザを識別する情報をアカウントとし、その下に配置しました。

承認が扱う情報は、サービスです。承認が扱うサービスを、その下に配置しました。

監視が扱う情報は、サービス利用（サービス利用後情報）です。

## クラス構成その2

クラス構成その1をもとに、さらにクラスを抽出します。



シーケンスの検討や PIM モデルの検討を通してクラス図は上記のように変化しました。以下は追加された最下段のクラスの責務と説明となります。

認証証明書：ログインアカウントと非ログインアカウントを区別するためにログイン時に生成されます。

承認証明書：サービス実行前にサービスの実行権が監視からも認められているか確認するクラスです。例えば監視条件を満たさない場合（課金不足など）、サービスの実行権を失う状況を実現します。

サービス利用量：監視の使用記録などの実データとなります。

## エンティティに着目したモデル

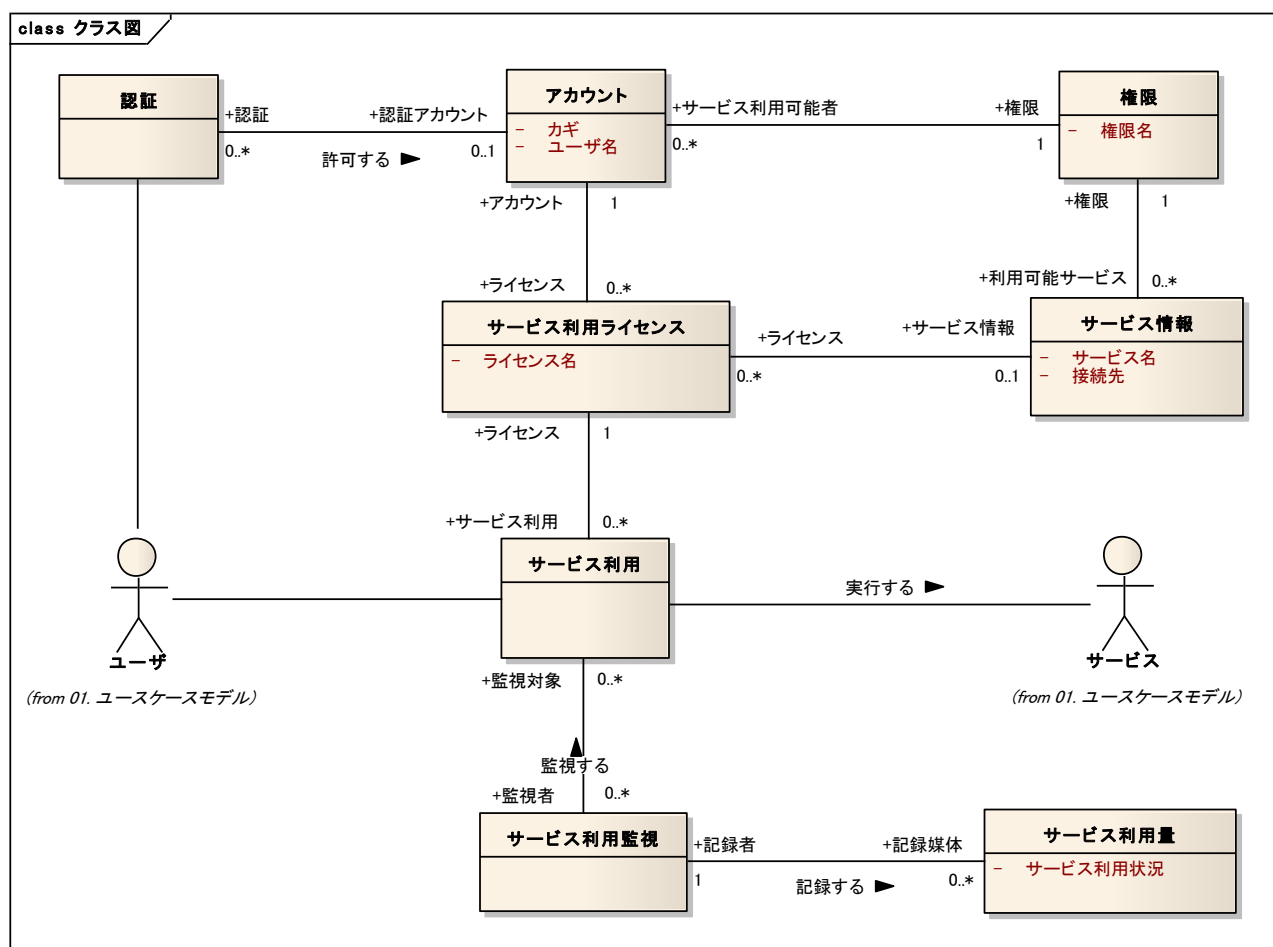
### モデリングのコンセプト

認証機能は、ユーザがサービスを使用したいという要求に対して、正当なユーザか？正当なサービスか？正しく使用されているか？といった観点からサービス使用に対して制限を掛けるという機能になります。この事からユーザがサービスを利用するまでに、ユーザにサービス利用の制限をかけるエンティティが幾つか出現すると考えられます。

これらのエンティティを明確化することをモデリングの出発点として分析作業を行いました。

### 分析モデル

#### 静的モデル



このクラス図において、「認証」「サービス利用」「サービス利用監視」がそれぞれ認証、承認、監視の役割に当たります。これら以外の情報がサービス利用に制限をかけるためのエンティティとなります。制限をかけるためにどのような情報が必要かという点を検討するために、サービスの利用時に制限するために行う確認事項を洗い出してみます。

- ・ 正当なユーザか？
- ・ 正当なサービスか？
- ・ 正当な組み合わせか？
- ・ 正当に使用されているか？（監視記録）

このうち、正当な組み合わせに関しては、サービスの実行時刻や実行場所など様々な条件が考えられますが、モデルを簡素化するために、一旦、アカウントとサービスの組み合わせが適切か？というシンプルな確認とします。

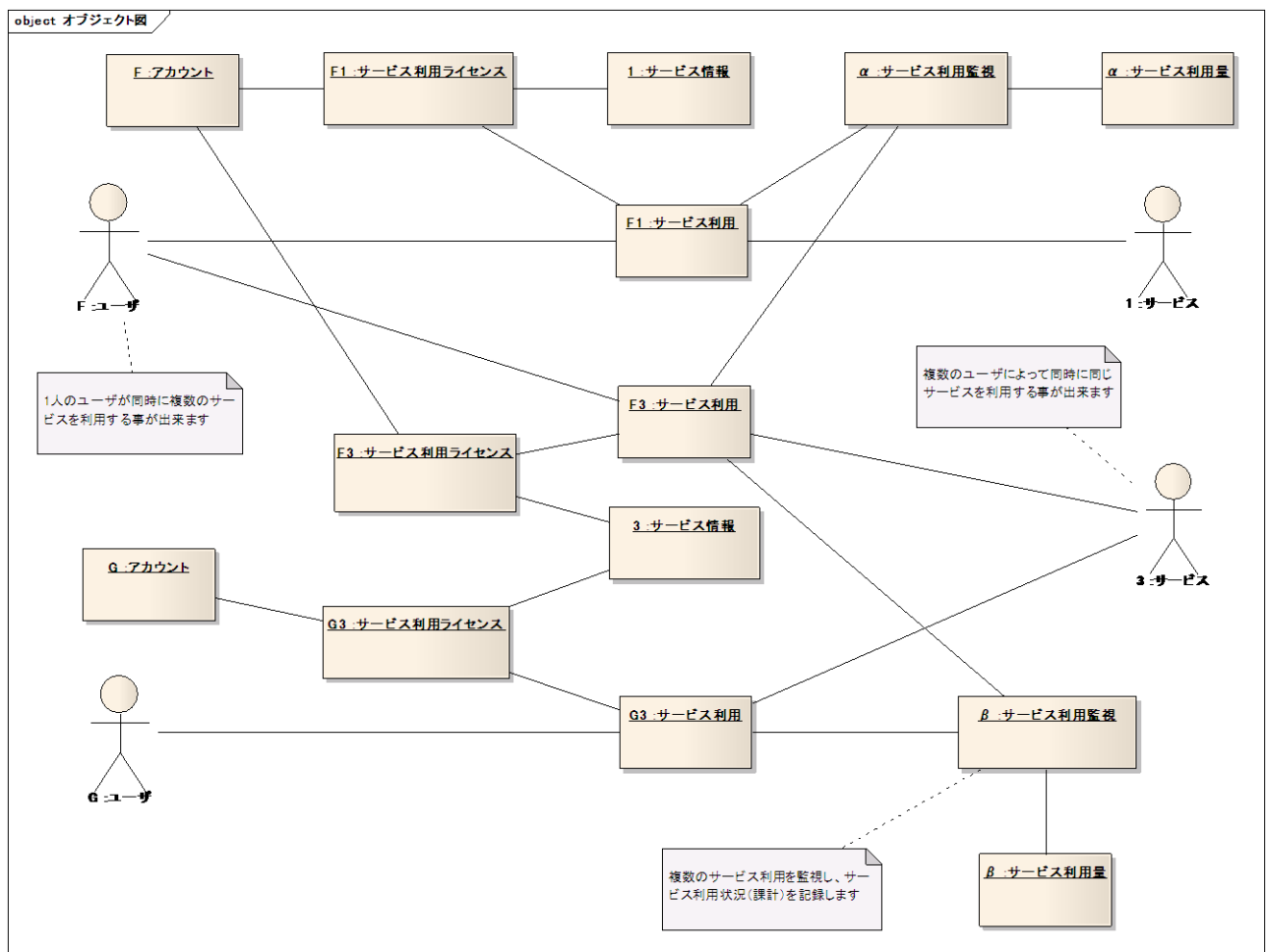
上述した確認事項から残りのエンティティを見た場合、

- ・「アカウント」は、サービス利用者が妥当かどうかを判断するための情報
- ・「サービス情報」は、正当なサービスを実行するための情報
- ・「権限」は、どのユーザがどのサービスを実行できるかを示した情報
- ・「サービス利用量」は、サービスの利用状況の記録

という位置づけになります。これで先ほどの確認に伴うエンティティは網羅できました。

クラス図の中で残された一つの「サービス利用ライセンス」は、ユーザが特定のサービスの実行を許可された事を示す情報として作成しました。

## サービス利用中のオブジェクト構成



## 状態に着目したモデル

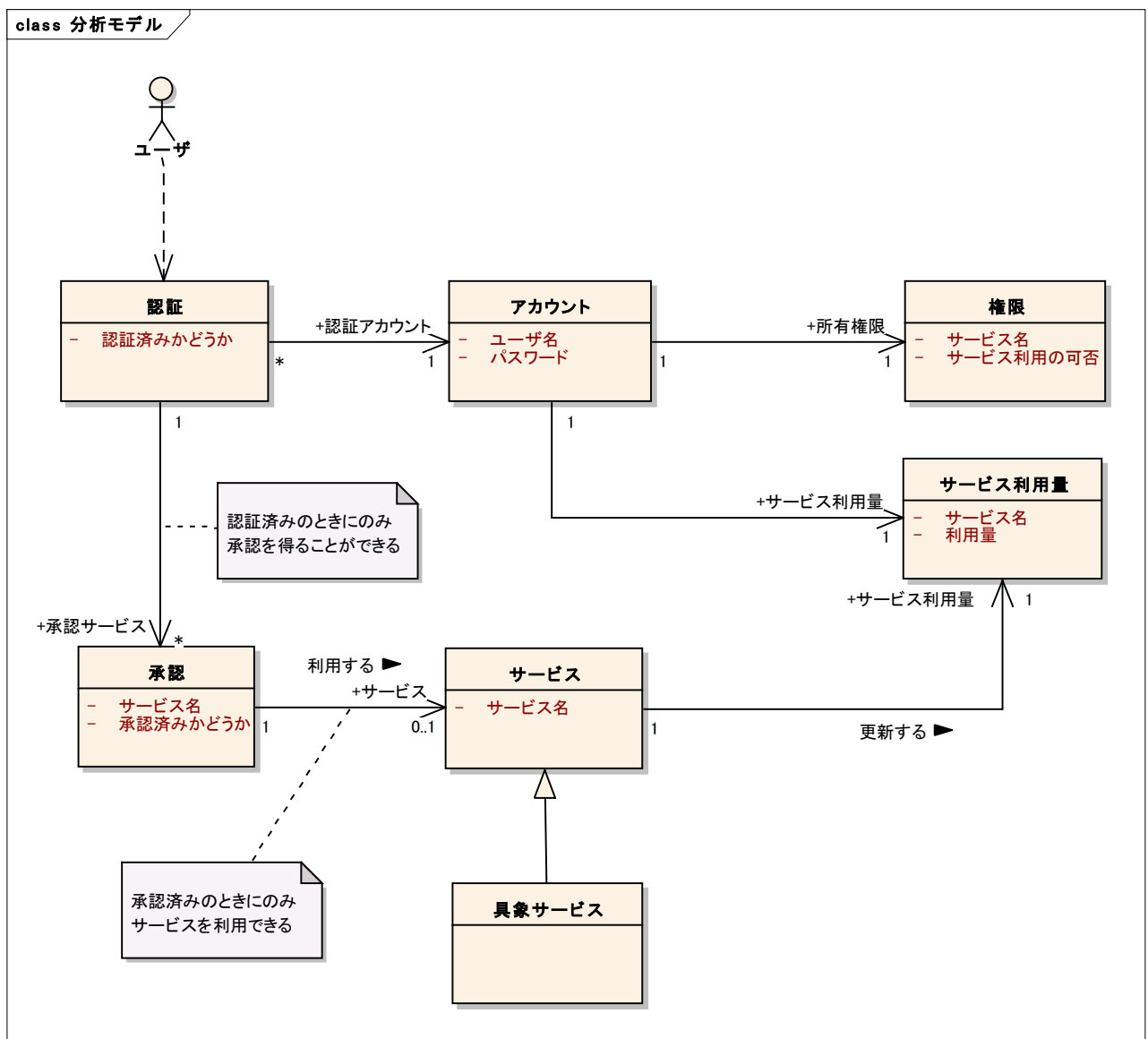
### モデリングのコンセプト

ユーザが認証された・認証されていない、といった状態に着目したモデルです。

認証モデルは、ユーザが認証されているか・認証されていないか、によってユーザからの処理依頼に対する振舞いを適切に切替えることが重要です。振舞いの切替えに、ヌケ・モレが無いことを、コーディング時ではなくて、構造設計レベルで抑えようというのがこのモデルのコンセプトです。

### 分析モデル

#### 静的モデル



ユーザとの認証に絡む振舞いを担うクラスとして「認証」を設けます。

ユーザ毎の情報を保持するクラスとして「アカウント」を設けます。「アカウント」には、認証に必要な情報「権限」と、そのアカウントが過去にサービスを利用した記録である「サービス利用量」を関連付けます。

ユーザが認証されたときに利用出来るサービスは複数あるため、サービス毎に承認が必要と考え、サービス毎の承認済みかどうかを表すために「承認」を設けます。

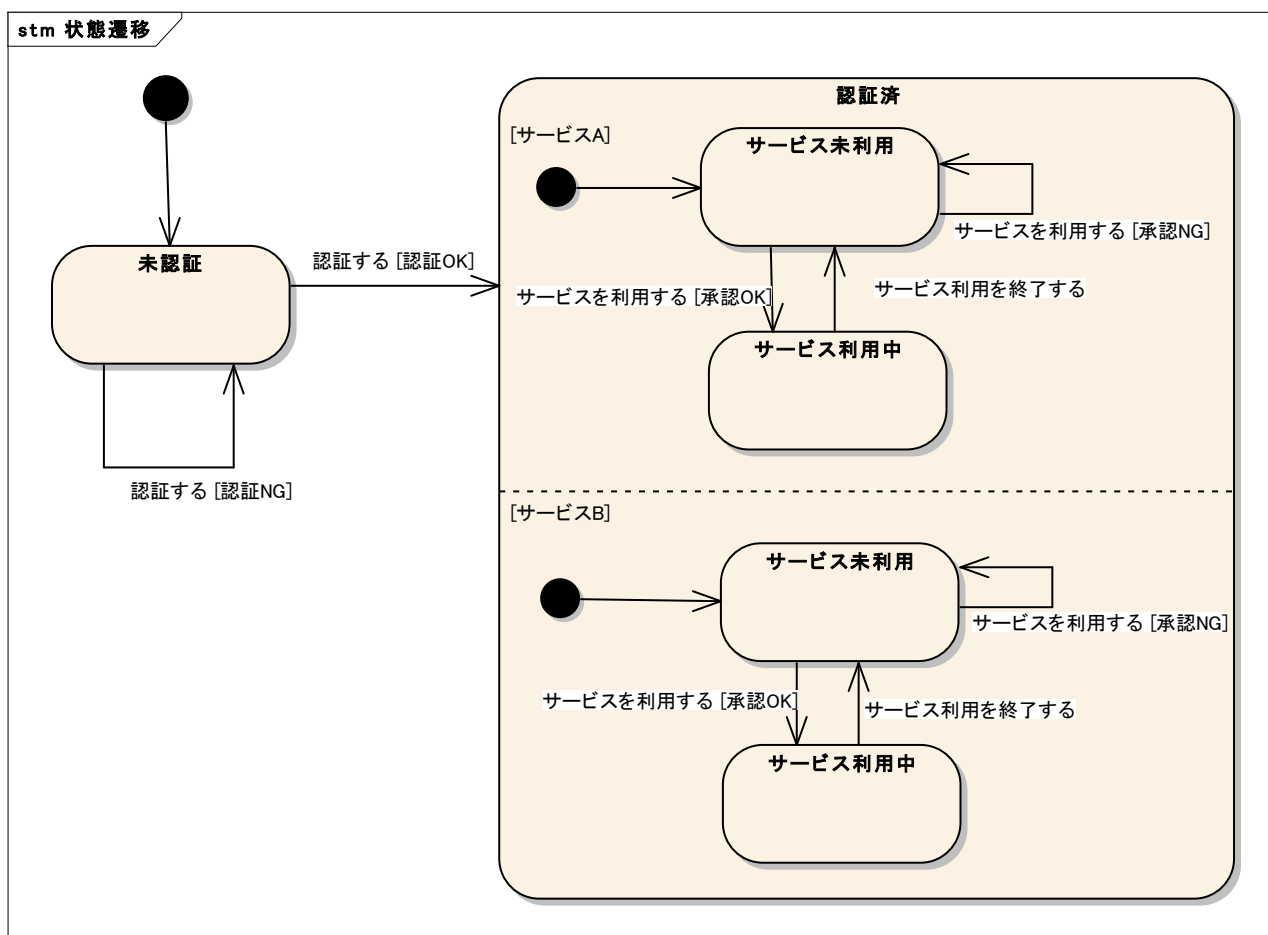


ユーザのサービス利用量を監視・記録するために「サービス」を設けます。各々のサービス固有処理は、「サービス」から派生した「具象サービス」が担うものとします。

ユーザによるサービス利用を監視した結果をユーザ毎に分けて記録するために、「サービス」からも、「アカウント」に関連づけた「サービス利用量」に関連を持たせます。

## 動的モデル

### 状態遷移



認証モデル全体が、ユーザ認証、サービス利用承認によりどのように状態が変化するかを示します。

ユーザが認証されていない状態を「未認証」、ユーザが認証された状態を「認証済」と呼ぶことにします。最初の状態は「未認証」です。ユーザが認証を試みて、システムに正当なユーザであると判断されると「認証済」へ遷移します。

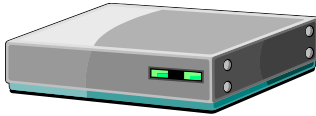
「認証済」状態にネストする形で、サービスを利用していない「サービス未利用」と、サービスを利用中の「サービス利用中」の2つの状態があります。「認証済」に遷移した直後は、「サービス未利用」です。これらの状態は、サービス毎にあります。ステートマシン図では、サービスAとサービスBの2つのサービスについて、それぞれ状態を持てることを示しています。

ユーザがサービスを利用しようとする時、システムが認可するかどうか判断します。システムが承認した場合は「サービス利用中」に遷移します。システムが承認しない場合は、「サービス未利用」に留まります。

ただし、「サービス未利用」と「サービス利用中」の状態は、システムに1つではなく、サービス毎に状態をもつ必要があります。

## F002 自己診断

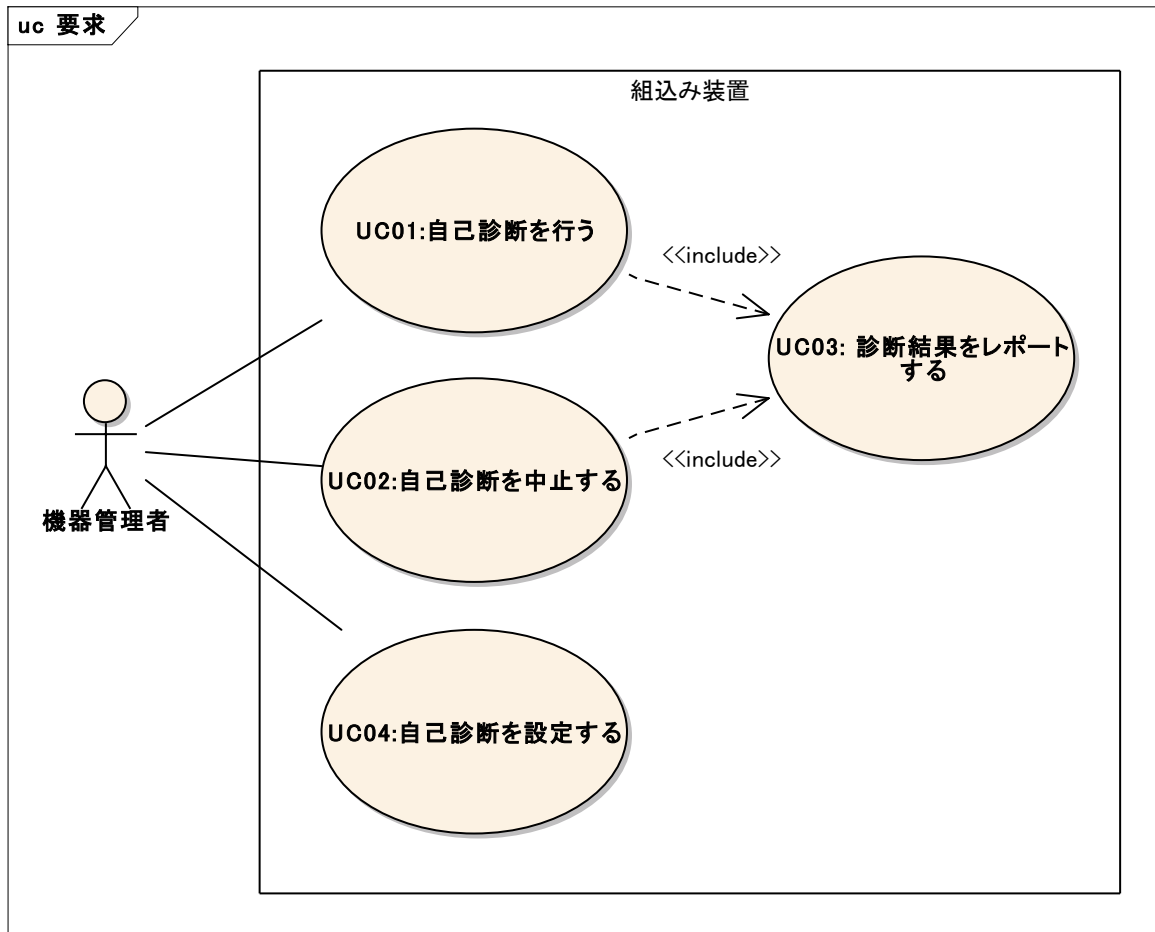
自動車や通信機器などには警告灯を備えるものや故障を示す表示ができるものがあります。これらには、装置が正しく動いているかどうかを調べる自己診断機能があり、異常を検知すると、(対処や)表示を行います。また、トラブル発生時に原因究明や対応を行うため、手動で自己診断ができる機器もあります。



### 要求仕様

1. 組込み装置には、装置が正しく動作しているかどうかを調べる自己診断機能があります。
2. 自己診断は、機器管理者の要求により起動される場合と、予め設定された条件に合わせて自動的に起動される場合があります。いずれも、機器管理者の指示により中止することが可能です。
3. 診断の対象には、デバイス単体や、複数デバイスから構成される機能ユニット、そして装置上で動作するアプリケーションが含まれます。  
機能ユニットは階層構造を持つことが可能で、上位の機能ユニットは複数の下位機能ユニットを含むことができます。
4. 一つの装置内では複数の自己診断を実施することが可能です。各自己診断は、1つの診断対象に対する診断でも、それらを任意に組み合わせたものでも構いません。  
(例：起動時診断や月次診断では全機能ユニット、週次診断では重要機能ユニット、日次診断では最重要ユニットなど。)  
また、自己診断に含まれる各診断は、並行・逐次、どちらの形でも実施可能です。
5. 診断対象ごとに、診断項目が決まっており、各診断ではどの診断項目を診断するかを設定することが可能です。  
(例：起動時診断や月次診断では診断項目 1~10 まで、週次診断では 1~5 と 6~10 を交互に、日次診断では 1 と 2 のみなど。)
6. 診断結果は、診断項目ごとに OK/NG の結果と、NG の場合には、エラーコードをレポートします。
7. 診断結果の履歴は保存する必要がありません。
8. 診断途中で中止されたときには、それまでの診断内容をレポートして終了します。

## ユースケース



## モデル一覧

モデル名	概要	ポイント
機能に着目したモデル	なし	
エンティティに着目したモデル	要求仕様の内容をエンティティを中心に分析し、診断と診断項目の静的構造に着目したモデルです。	要求仕様の内容を忠実に分析しています。
状態に着目したモデル	なし	
メタファを使ったモデル	装置の自己診断機能を、日常の診断(健康診断など)にたとえてモデリングしています。	身近な話題から入るので、モデリングの導入編としてお勧めです。

## エンティティに着目したモデル

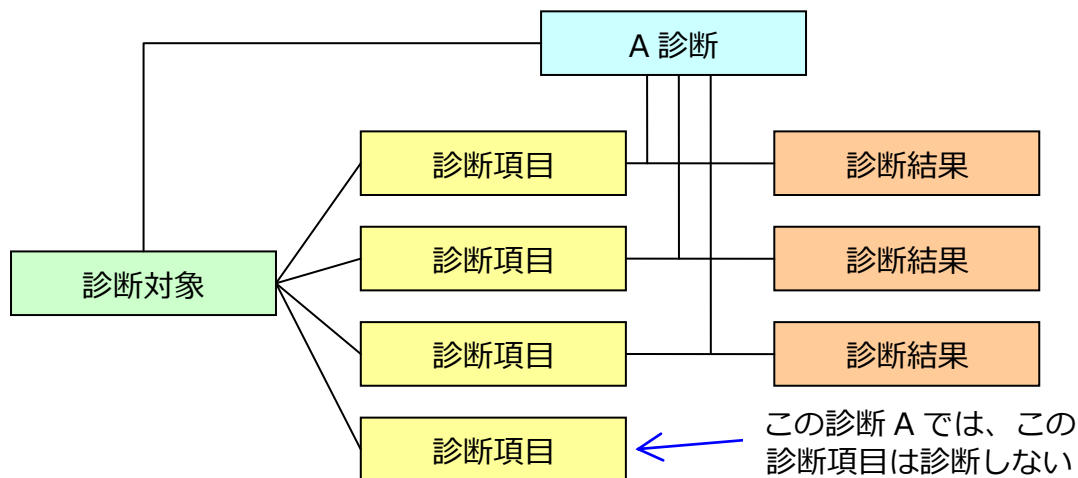
要求仕様をエンティティに着目し、診断を診断対象の診断項目と対応する結果と捉え、このような診断を組み合わせることでさまざまな診断をモデリングしています。

### モデリングのコンセプト

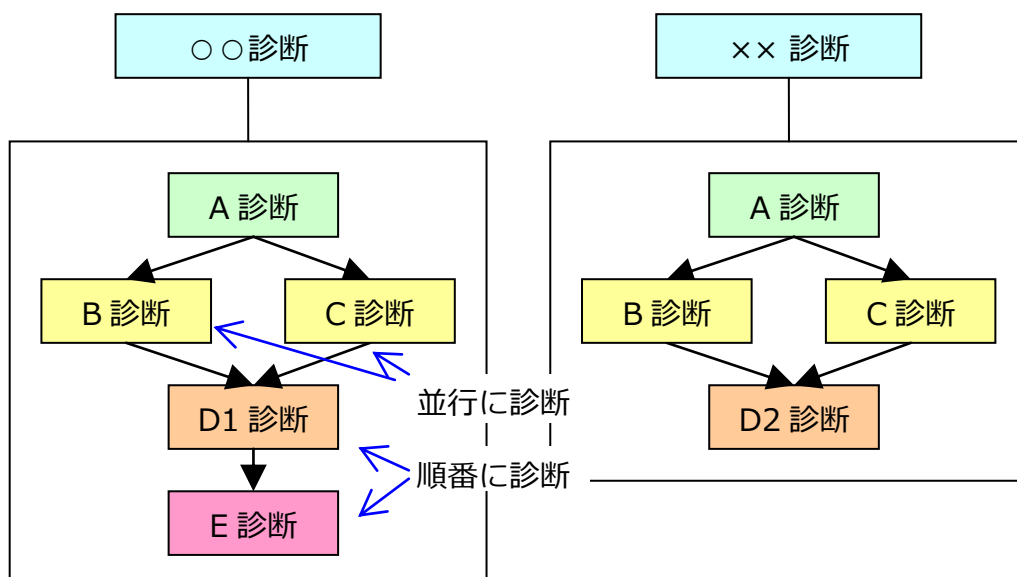
要求仕様の内容を忠実に分析し、モデリングします。

要求仕様から以下の特徴が読み取れました。

- (i) (後述の組み合わせた診断に対して) 単一の対象に対する診断とは 1 個の診断対象に対して定められた診断項目を診断し、診断結果を得ることです。診断対象に対して診断項目は決まっていますが、診断毎にどの診断項目を診断するかを設定することが可能です。(UML 図ではなく)簡単に図示すると以下のようになります。



- (ii) 診断は複数の診断から組み合わせることもでき、以下のように 並行/順番に実行されます(組み合わせた診断)。(UML 図ではなく)簡単に図示すると以下のようになります。

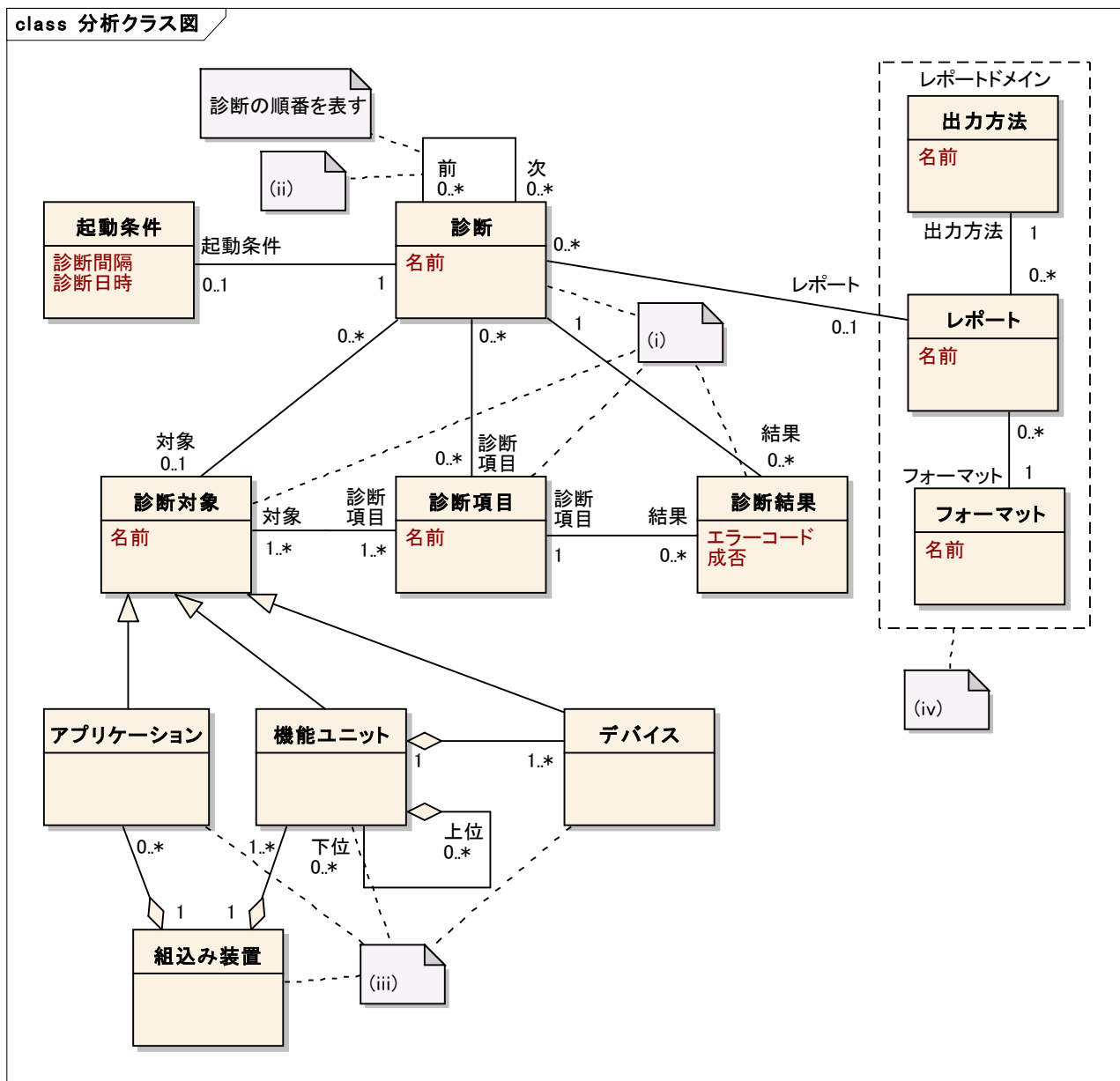


## 分析モデル

前述の(i)より、診断、診断対象、診断項目、診断結果、の各クラスとそれらの関係、(ii)より診断クラスの自己関連が導き出されます。さらに以下を加え、クラス図を作成しました。

(iii)要求仕様3より診断対象の構造(組込み装置、機能ユニット、デバイス、アプリケーション、の各クラス)

(iv)レポートを表すレポートドメイン(レポート、フォーマット、出力方法、の各クラス)



## メタファを使ったモデル

### モデリングのコンセプト

装置の自己診断機能をモデリングするにあたり、まずは診断の意味や目的を考え、日常生活の診断に適用できる概念モデルを作成してみます。診断についての理解を深め、概念を明確にすることで、分析モデルや設計モデルの作成に役立つと思います。

#### 1. 診断って何だろう？

診断とは病気や欠陥など悪いところがないか調べることです。装置の場合は故障箇所や不具合箇所がないか調べることです。悪いところに限らず、悪くなりかけているところや悪くなりそうなところも調べられると、よりよい診断といえるでしょう。

◆ 診断の類義語：検診、点検、検査

#### 2. 診断する目的は何だろう？

問題箇所や予防箇所がないかを診断することで、自己修復あるいは他者に治療・改善・修理依頼するきっかけを知ることができ、診断対象を長生き・長持ちさせるのが目的と考えられます。

#### 3. 身近な診断には何があるだろう？

身近にある診断、点検、検査を探してみました。

- ◆ 健康診断
- ◆ 車検
- ◆ 温水洗浄便座
- ◆ 分電盤
- ◆ 消防設備点検

#### 4. 診断に関する共通項目には何があるだろう？

前述の身近な診断(点検、検査を含む)において、共通する項目を挙げてみます。

対象物 共通項目	人	車	温水洗浄便座	分電盤	消防設備
診断名	健康診断 人間ドック	車検 定期点検	テスト	テスト	消防点検 防災点検
診断時期	年1回 望んだとき	初回3年 以降2年毎	ボタンを押した とき	ボタンを押 したとき	半年毎
診断対象	本人、子供、社員	車	電気設備	電気設備	消防設備
診断依頼者	本人、親、会社	所有者	家人	家人	本人、管理会 社、企業
診断場所	病院、健康管理室	車ディーラー、 車検場	設置場所	設置場所	設置場所
診断実施者	医者	メカニック	装置	装置	業者
診断項目	血液、血圧、視力、聴 力、胃、肺、心臓など	エンジン、ブレ ーキ、オイル、 タイヤなど	ショート	漏電、ショ ート	熱検知、煙検 知、防災扉、シ ャッター等
診断結果	健康ランク、問題部	問題なし、	ブザー？	ブザー？	異常なし

	位、状態など	要修理箇所、 予防修理箇所		異常箇所
--	--------	------------------	--	------

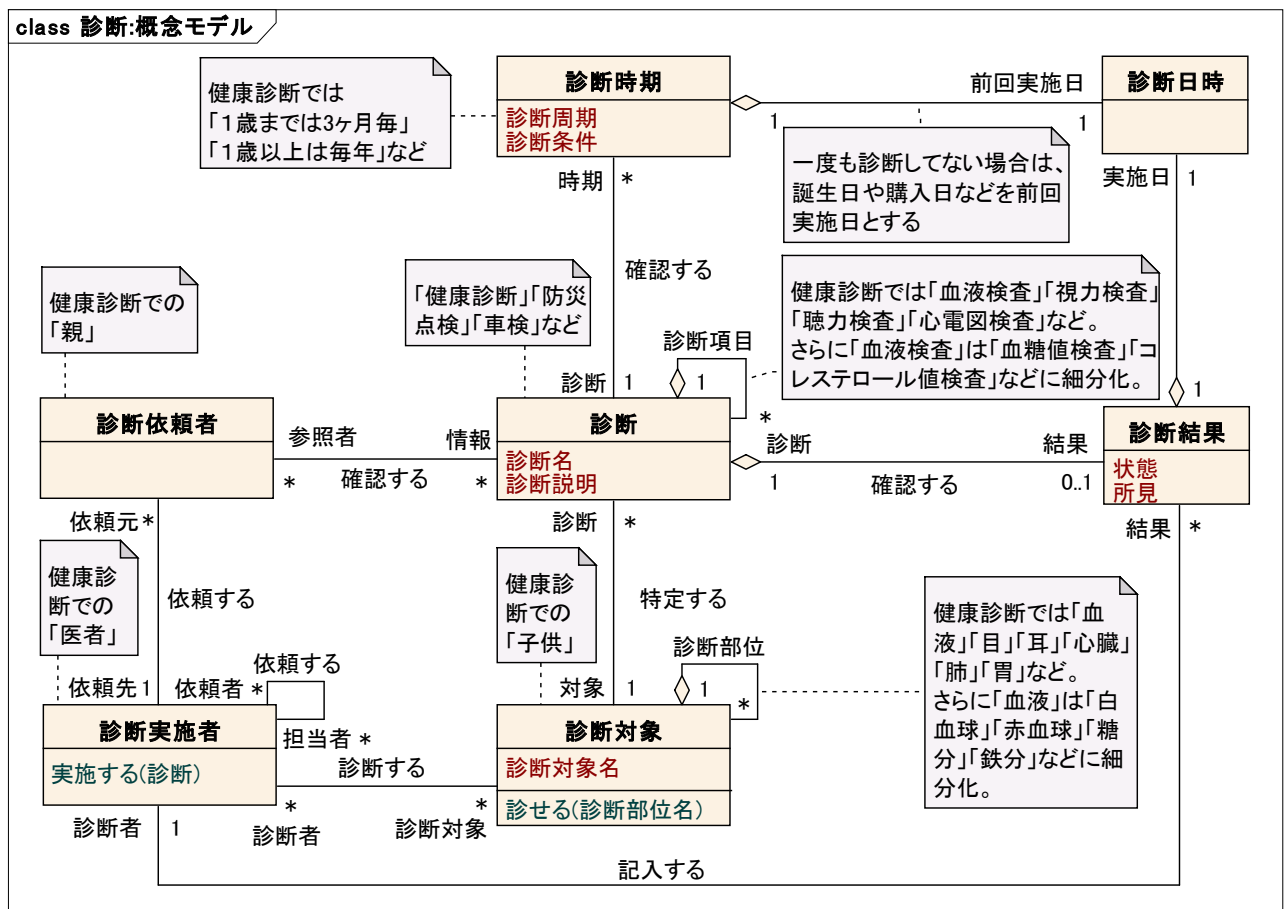
### 5. 診断シナリオ

共通項目を使って、診断するときの手順(シナリオ)を考えてみます。

- ◆ 登場人物(アクター)は「診断依頼者」「診断実施者」「診断対象」の3人。
- ◆ 本人が本人の診断を依頼すれば「診断依頼者」と「診断対象」は同一です。
- ◆ 「診断依頼者」は、「診断時期」になった「診断」がないか定期的に確認し、あれば「診断実施者」に「診断対象」の「診断」を依頼します。
- ◆ 「診断時期」に関係なく、「診断対象」が調子悪くなったり、なんとなく気になったときに「診断依頼者」が「診断実施者」に「診断対象」の「診断」を依頼することもあります。
- ◆ 「診断実施者」は、「診断項目」毎に「診断対象」を「診断」し、「診断結果」を記入して「診断依頼者」に通知します。
- ◆ 「診断対象」は「診断部位」を診せます。
- ◆ 「診断依頼者」は、「診断結果」を確認し、問題があれば治療や修理などを施します。

### 静的モデル

共通項目を使って、概念モデルのクラス図を書いてみます。







## モデルカタログ：部品編

部品編のモデルカタログです。

## C001 目標制御

### はじめに

目標制御は、特定の制御対象・システムに対して式やテーブルといった最適化が施された形態で実装されることが多くあります。このモデルは、目標制御の実装形態はブラックボックスとし、その中身に依存しない形にしました。このモデル（部品編目標制御）では、登場するクラス数が少なく、モデルの全体と対象物が把握しやすいものとなっています。

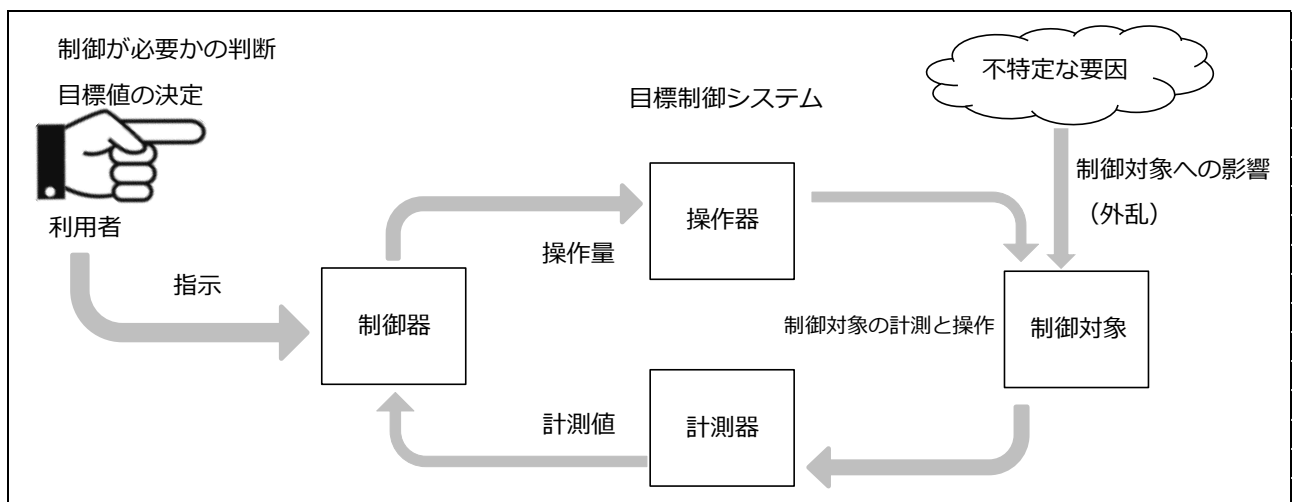
### 要求仕様

目標制御システムとは、制御対象の計測値が目標値となるように制御する仕組みで、エアコンの温度調節や自動車の ABS など数多くの製品に組み込まれています。

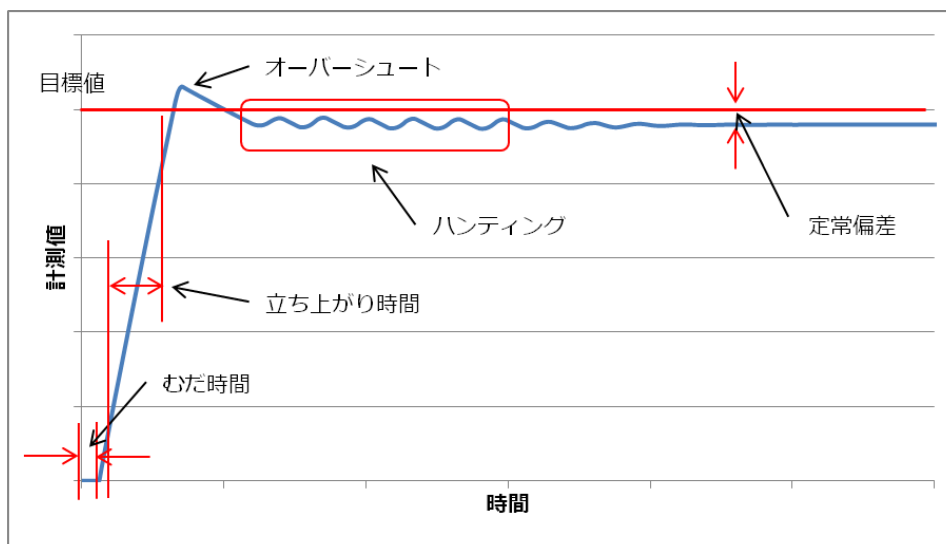
### 目標制御システムの基本的な構成と動作

目標制御システムは計測器、操作器、制御器で構成されます。利用者は制御が必要な状況で、目標制御システムに制御の開始を指示します。目標制御システムは、制御中は一定時間周期で制御対象を計測し、それに応じた操作を行うというフィードバックを繰り返します。

制御対象は目標制御システム以外の不特定な要因からも影響を受けますが、その都度出力を修正することで、フィードバックを繰り返すことで、うまく制御を行えるようになっています。



### 目標制御の実例



目標制御を開始すると、操作器の操作量が制御対象に伝わるまでの時間（むだ時間）を経過した後、計測値は目標値に近づいていきます。このときの目標値に近づくまでの時間が立ち上がり時間です。計測値が目標値に達したときに、勢いがついていたりすると、オーバーシュートが発生します。その後、ハンテイングをしながら、徐々に計測値は目標値付近で安定するようになります。最終的に安定したときの計測値と目標値の差が定常偏差として残ることになります。

目標制御を実現するアルゴリズムには、ON/OFF 制御、PID 制御、ファジィ制御などの制御方式があり、制御方式によって、目標値への近づき方の特徴に違いがあります。注：これらの制御方式については付録を参照してください。

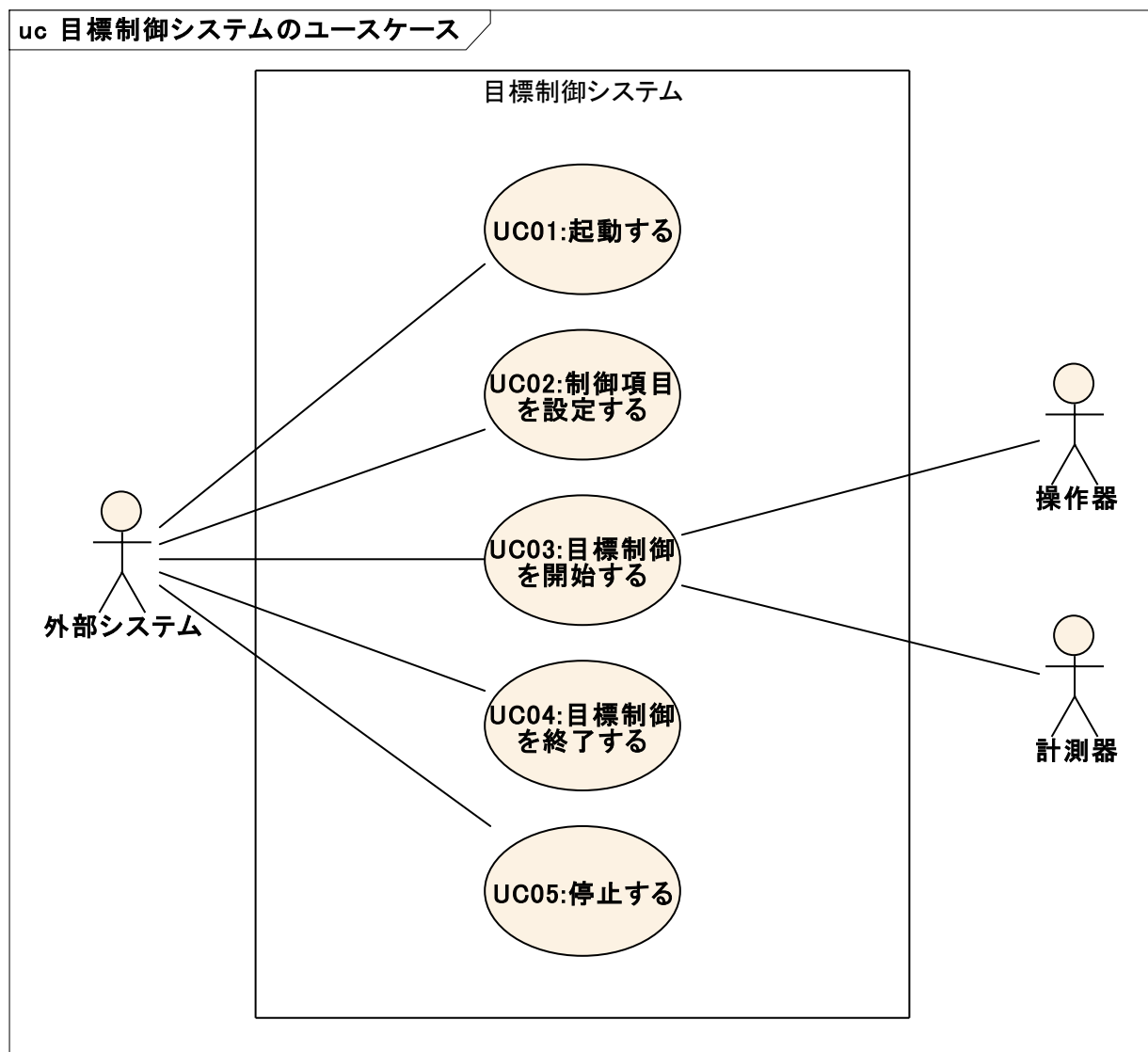
## 用語定義

すでにいくつかの用語を使用していますが、あらためて用語の定義を行います。しかし、それだけではイメージが付きにくいと思いますから、具体的な製品（例としてエアコンとABS）で、何に該当するかもあわせて以下に示します。

用語	用語の意味	エアコンの場合	ABS の場合
目標制御システム	制御対象を制御するための仕組み	－	－
制御対象	目標制御を行う対象	室温	車輪速度
目標値	目標とする制御対象の値	指定された室温の値	指定された車輪速度の値
計測値	計測された制御対象の値	現在の室温の値	現在の車輪速度の値
外乱	制御対象に加えられる予測できない影響	室外との熱のやり取り、等	路面との摩擦、等
制御器	目標制御を司る部品	－	－
入力器	利用者からの指定内容や指示を受け付ける部品	入力パネル	ブレーキペダル
計測器	制御対象を計測する部品	室温センサ	車輪速度センサ
操作器	何からのアクチュエータを駆動させることで、制御対象を操作する部品	コンプレッサー	油圧サーボ
操作量	操作器がもつ指定可能な量で、結果的に制御対象を操作する量	コンプレッサー出力	油圧サーボ出力
制御方式	目標制御を実現する方式、アルゴリズム	－	－

## ユースケース

本カタログモデルが実現する範囲のユースケースを示します。



補足：アクター「外部システム」とは、最終的な製品（エアコン等）に含まれるプログラム（目標制御システムを使用するプログラム）を指します。

## ユースケース一覧

No.	ユースケース名	概要／備考
UC01	起動する	目標制御システムを起動する。
UC02	制御項目を設定する	目標値やその他の制御項目を変更する。 操作量の計算途中にも制御項目を変更できます。
UC03	目標制御を開始する	制御対象の計測値が目標値となるように制御する。 一定時間周期で制御対象の計測と操作を繰り返すことで、 制御対象は徐々に目標値に近づいていく。
UC04	目標制御を終了する	制御対象の操作を止める。
UC05	停止する	目標制御システムを停止する。

## モデル一覧

モデル名	概要	ポイント
機能に着目したモデル	なし	
エンティティに着目したモデル	目標制御システムを構成する部品をエンティティとして捕らえたモデルです。	部品を中心としたシンプルなモデルです。
役割に着目したモデル	目標制御システムについて必要な役割に着目しました。実現方法にとらわれず「現実と目標を一致させる」という目標制御の本質を抽象化したモデルです。	オブジェクト指向を意識したモデルです。
状態に着目したモデル	なし	
メタファを使ったモデル	なし	

## エンティティに着目したモデル

### モデリングのコンセプト

静的な構造として目標制御を行うのに必要な構成部品に着目しました。

また、要求仕様に登場する ON/OFF 制御、PID 制御、ファジィ制御といった複数の制御方式に対応できるように考慮しました。

### 分析モデル

#### 1. 制御対象および計測、操作について

実際の制御対象（エアコンであれば室温で、ABS なら車輪速度です）の今の状態の保持やそれに関する情報を保持するクラスとして「制御対象クラス」を定義しました。制御対象クラスは属性として目標値と計測値を持ちます。なお、「計測値」属性は実際には操作であり、操作を呼ぶと計測器クラスの「計測する」操作を呼び出します。

制御対象には一般的に、制御対象特有の上限値・下限値、あるいは計測器が持つセンサからくる上限値・下限値（計測限界）が存在します。そのため、制御対象クラスは範囲クラスを保持する形となります。

#### 2. 制御方式について

制御器が制御を行う際、どのように制御するか？（制御対象への出力をどうするか？）（例えば計測値が目標値に近づいてきたから、出力を弱めるといったこと）を考える必要があります。これを制御方式クラスが行います。制御方式には ON/OFF 制御、PID 制御、ファジィ制御とさまざまな方法がありますから、制御方式クラスは抽象クラスとなります（ストラテジーパターン）。制御方式クラスが操作量を算出する際には、元となるデータ（制御パラメータ）が必要となりますので、制御方式クラスは制御パラメータクラスを保持することになります。

制御方式クラスでは操作量算出時に必要な内部データ（PID 制御であれば例えば、偏差（目標値と現在値の差）の時間積分値）が必要ですが、これは実際の制御方式に依存するため、ここでは抽出していません。制御パラメータが持つデータについても同様で、PID 制御であれば比例係数、積分係数、微分係数の 3 つを持つこととなりますが、これも制御方式に依存するものですのでここでは抽出していません。注：PID 制御の詳細については付録を参照してください。

#### 3. 定期的な制御について

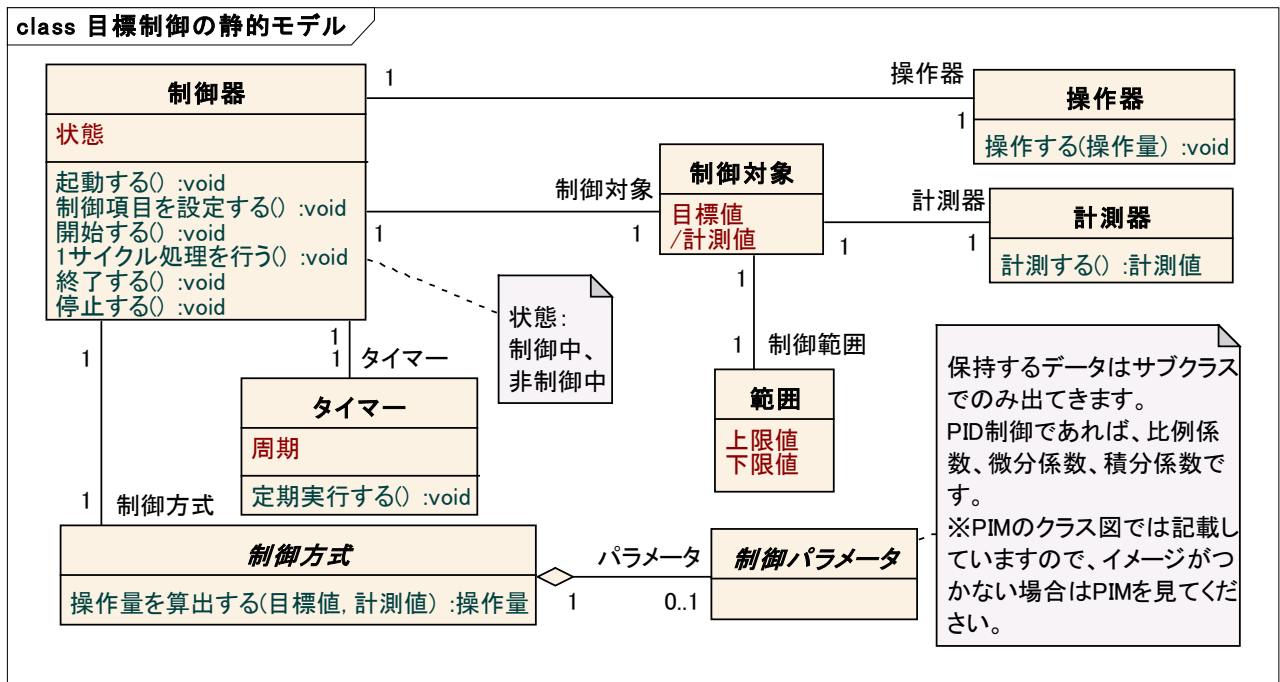
制御は定期的に行う必要があるためタイマーが必要となりますから、それを定義しました。分析の動的モデルではタイマーが目標制御システムで定義される操作を直接呼び出す形としますが、本来は、一般にタイマーは目標制御部品とは別の部品であり、タイマーが目標制御システムで定義している操作を直接呼び出すことはできません。これについては PIM で検討します。

#### 4. その他

制御対象を実際に制御する場合、その良し悪し、具体的には立ち上がり時間、オーバーシュート量、ハントニング量、定常偏差の量などを、考える必要があります。しかし、これについては制御方式クラスのサブクラスが考えることであるため、クラス構成を抽出する際には考慮対象外としました。

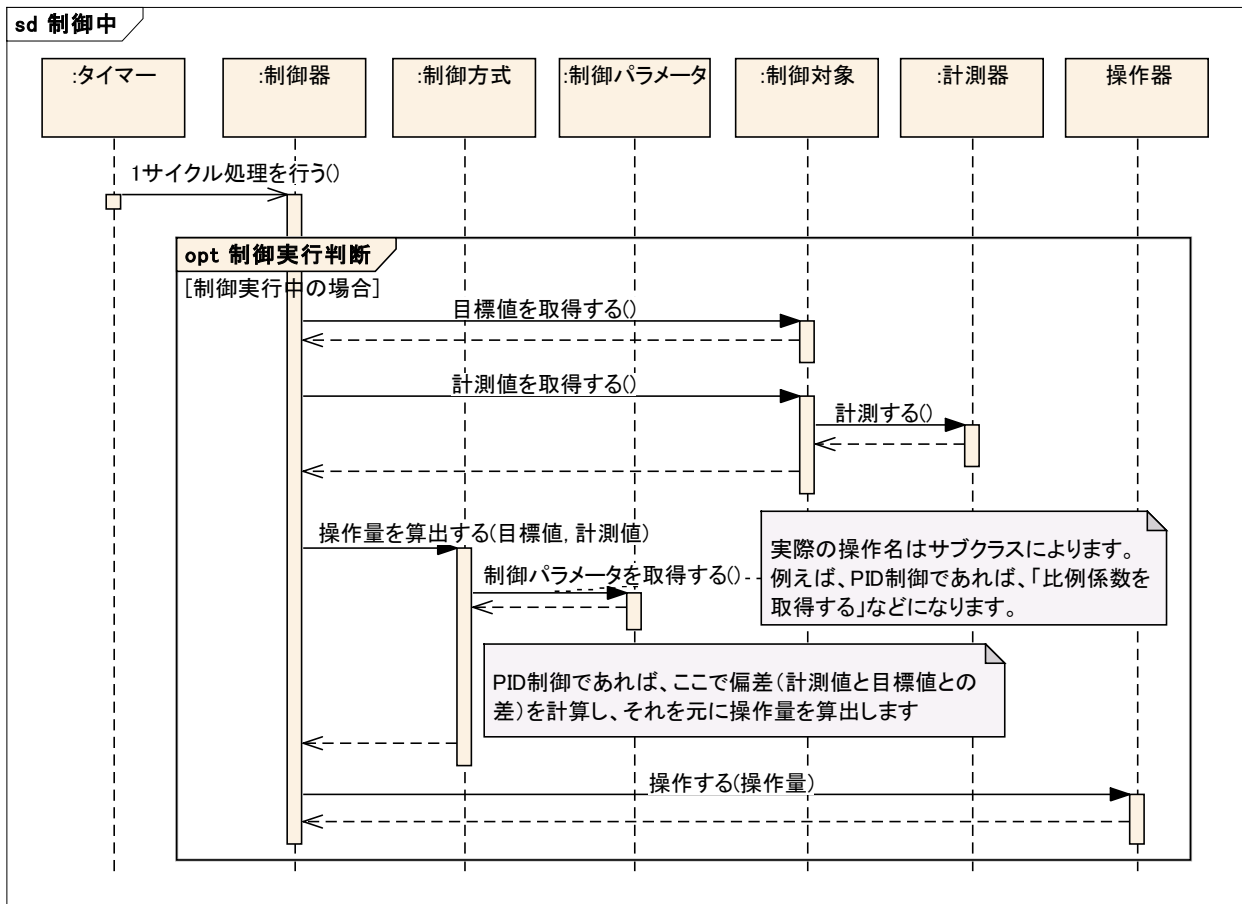
## 静的モデル

### 目標制御システムのクラス構造



## 動的モデル

### 「制御中」の相互作用



## 役割に着目したモデル

### モデリングのコンセプト

目標制御の理想と現実を近づける機能に着目しモデリングしました。

目標制御の目的である「対象の現実を理想に近づける」という点をアーキテクチャの中心に据え、それぞれの役割を擬人化したモデルです。このようなモデリングの考え方もあるというサンプルとして、分析モデルのみ提示します。

### 分析モデル

#### 1. 制御対象および計測、操作について

実際の制御対象（例えばエアコンにおける室温、ABS における車体速度）を表すクラスとして、「対象」クラスを設けました。「対象」クラスには「理想」と「現実」2つのロールがあります。「理想」のロールは目標とする制御対象の値を保持し、「現実」のロールは計測された制御対象の値を保持します。

「対象」クラスは「更新させる人」クラスによって定期的にその内容を更新されます。「現実」のロールには制御対象の計測（更新）時刻が必要なため属性として持ち、「理想」のロール側には必要ないため、属性に多重度を示しています。

また、目標制御の入力数と出力数はそれぞれ1つずつのモデルです。

#### 2. 制御方式について

制御を行う際、どのように制御するか（例えば計測値が目標値に近づいてきたら、出力を弱めること）を考える必要があります。これを制御アルゴリズムクラスが行います。

制御アルゴリズムクラスの制御方式には ON/OFF 制御、PID 制御、ファジィ制御とさまざまな方式があるため、制御アルゴリズムクラスは抽象クラスです。（ストラテジーパターン）。制御アルゴリズムクラスは先の「理想」と「現実」を用いて、操作量を決定します。

#### 3. 定期的な制御について

目標制御を定期的に行うため「タイマー」クラスを定義しました。「タイマー」クラスは「一致させる人」クラスを呼び出し、「一致させる人」クラスは理想と現実を制御アルゴリズムに渡して定期的な制御を行います。

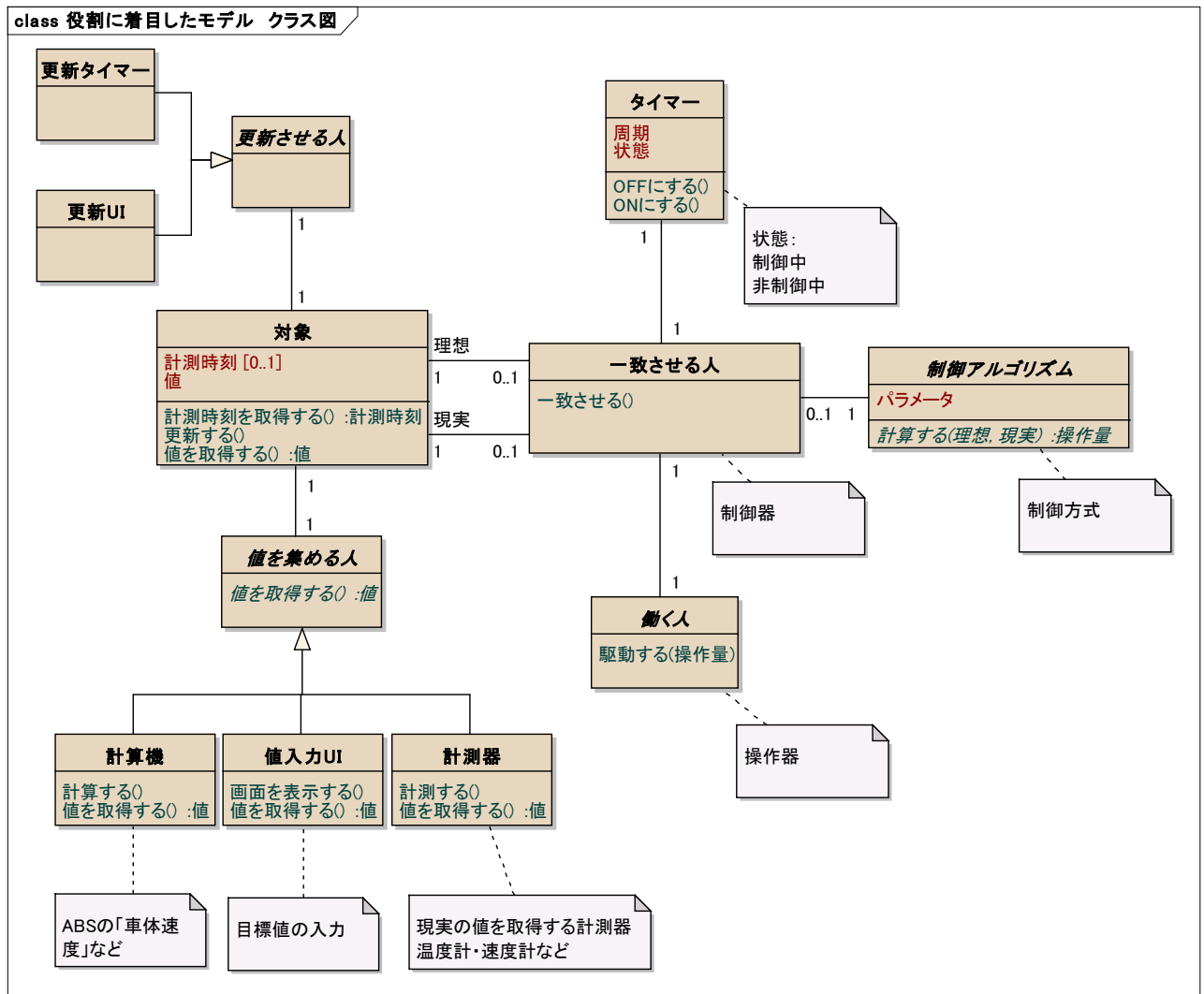
また、別のタイマーとして、対象クラスを定期的に更新する、「更新させる人」クラスを定義しました。

「更新させる人」クラスが先の「タイマー」クラスと別に存在する理由として、「現実」のロールでの制御対象の計測は、「一致させる人」クラスの制御間隔とは異なる間隔で実行される場合があること、また「理想」のロールで保持している目標とする制御対象の値が不定期に変更される可能性があるためです。

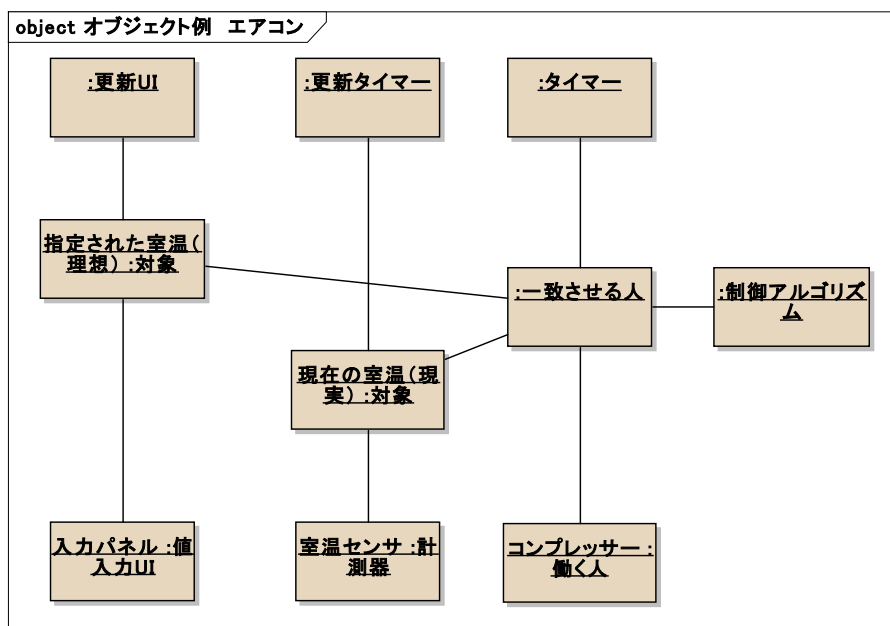


## 静的モデル

### 目標制御のクラス構造



### オブジェクト例 エアコン



## おわりに

2010年10月に第一版のモデルカタログを出してから、約1年半。この間、メンバーを再構成して第二版のモデルカタログの作成に取り組んできました。第二版のモデルカタログ作成にあたっては、第一版の反省を踏まえ、「モデルの品質向上」と「カタログとしての見やすさ」を目標に掲げ、メンバー全員で熟考を重ねてきました。

たとえば「モデルの品質向上」では、モデリング経験豊富なメンバーにモデル作成チームの専任レビュアーとして活動してもらったり、ライターズワークショップと呼ばれる厳密なレビューセッションを通じてモデルのブラッシュアップを行ったりしてきました。また、「カタログの見やすさ」については、これまで一冊にまとめていたカタログを、モデルの一覧をサマリして掲載した全体版(本書)と個々のモデルを詳細に説明したモデル版(モデル解説書)に分けるなどの工夫をおこないました。

そして、このような新たな試みを通じて、今回なんとかリリースにこぎ着けたのが「製品編」の「電子オルゴール」と、「部品編」の「目標制御」という2つのモデルです。いずれも担当メンバーやレビューを行った他メンバーの思いがこもった力作となっています。ぜひ、じっくりと読み込んでいただくと嬉しいです。

今回も、第一版同様、所属・仕事内容・経験のそれぞれが異なるメンバーが、業務の合間を縫って集い、作成を行ってきました。前回と違い、今回は各自が担当するモデル作成チームごとの集まりを主体に進めてきましたが、特に今回の2つのモデル担当のメンバーの活動量は相当なものがありました。今回無事リリースできたのは、メンバーのモデリングに対する熱意がそれを上回った証だと思えます。

さて、モデルカタログですが、これからも版を重ねていく予定です。今回の2つのモデルに続き、続々とリリース予定のモデルが控えています。それに向けて、みなさんからのフィードバックやご要望をいただきたいのはもちろんですが、いっしょにモデルを作ってみたいという方も歓迎いたします。このモデリング部会は、会社や業務の枠を超えて、純粋にモデラーとしての活動に専念出来る貴重なフィールドです。モデリングに興味のある方や、UMTPで資格認定されたスキルを実際のモデリングの場で試したい方など、多くの方のみなさんの参加をお待ちしています。ご意見やご参加のお問い合わせについては、以下のURLをご覧ください。

<http://www.umtp-japan.org/>

(上記UMTPサイトから「UMTPについて」→「活動報告」→「組込みモデリング部会」)

最後に、第一版に引き続き、第二版のモデルも、ぜひ、開発の現場において活用して頂ければ、開発メンバー一同嬉しい限りです。

**カタログ作成に携わったメンバー** (あいうえお順、敬称略。社名は参加当時のもの)

池下 弘之	(日本電気株式会社)
石井 基樹	(株式会社 JVC ケンウッド)
石滝 智洋	(オリンパスソフトウェアテクノロジー株式会社)
石田 晴幸	(株式会社富士通コンピュータテクノロジーズ)
大嶋 主悦	(株式会社豆蔵)
大森 淳夫	(パイオニア株式会社)
小黒 登行	(株式会社豆蔵)
金子 翔	(アドソル日進株式会社)
栗田 道広	(スター精密株式会社)
河野 岳史	(スパークスシステムズジャパン株式会社)
小林 健一	(株式会社豆蔵)
品川 正彦	(株式会社日立インフォメーションアカデミー)
清水 鉄矢	(オリンパスソフトウェアテクノロジー株式会社)
白神 一久	(株式会社富士通システム統合研究所)
新家 了訪	(株式会社テイジイエル)
鈴木 茂	(株式会社オージス総研)
高原 大樹	(アドソル日進株式会社)
時田 大介	(スター精密株式会社)
西口 敦子	(NEC ラーニング株式会社)
畑 理介	(株式会社オージス総研)
羽生田 栄一	(株式会社豆蔵)
古谷 聡	(日本プロセス株式会社)
松嶺 恭守	(理想科学工業株式会社)
宮崎 崇	(ムトーアイテックス株式会社)
山本 裕行	(株式会社日立インフォメーションアカデミー)
芳村 美紀	(株式会社エクスモーション) : 副主査
力武 克彰	(仙台高等専門学校)
渡辺 博之	(株式会社エクスモーション) : 主査

## 組込み分野のための UML モデルカタログ

---

初版発行 2010年（平成22年）10月1日

第四版発行 2013年（平成25年）5月1日

発行者 UMLTP, Japan

編著 組込みモデリング部会

印刷

---

UMLTP, Japan

東京都渋谷区代々木1丁目22番1号

<http://www.umltp-japan.org/>