オブジェクトモデル分科会

インタフェースの仕様化とオブジェクト指向モデリング

2002年3月19日

（株）エクサ
児玉公信

目　次

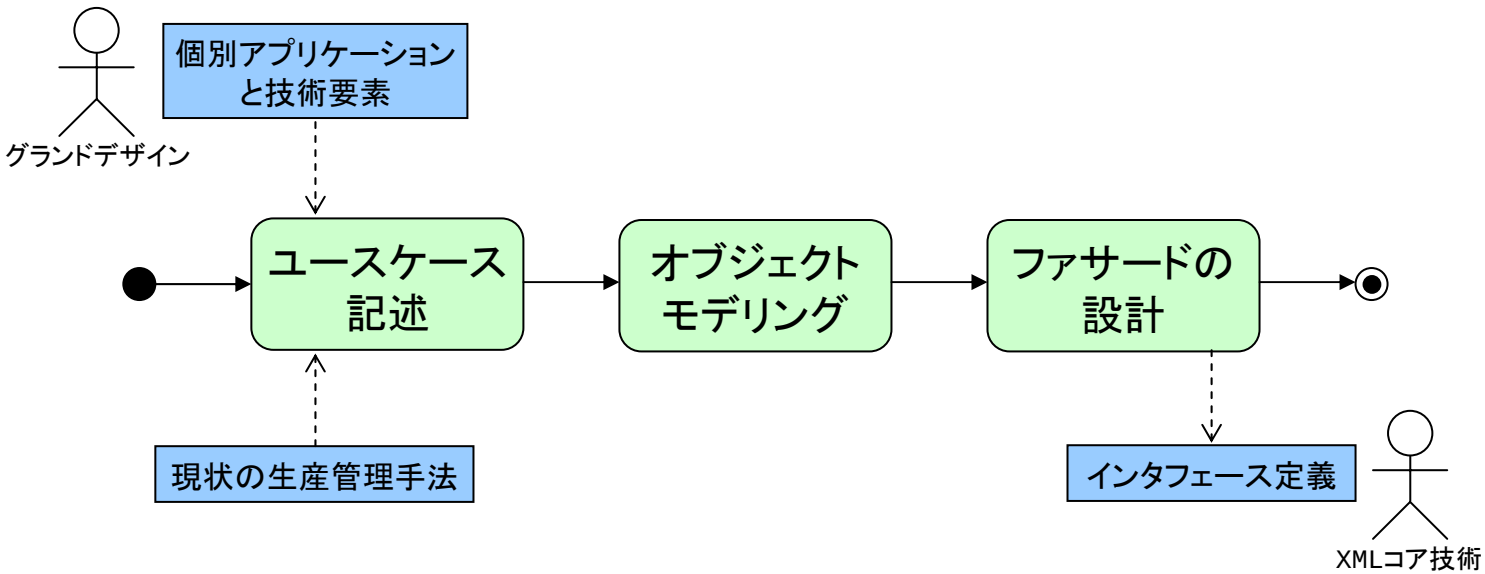
はじめに

広義のAPSが対象

APSと周辺コンポーネントの相互作用の記述

ユースケースとオブジェクトモデルによる理解

インタフェースの定義



1. APSの相対的位置づけ

1.1 APSの境界とビジネスコンポーネント

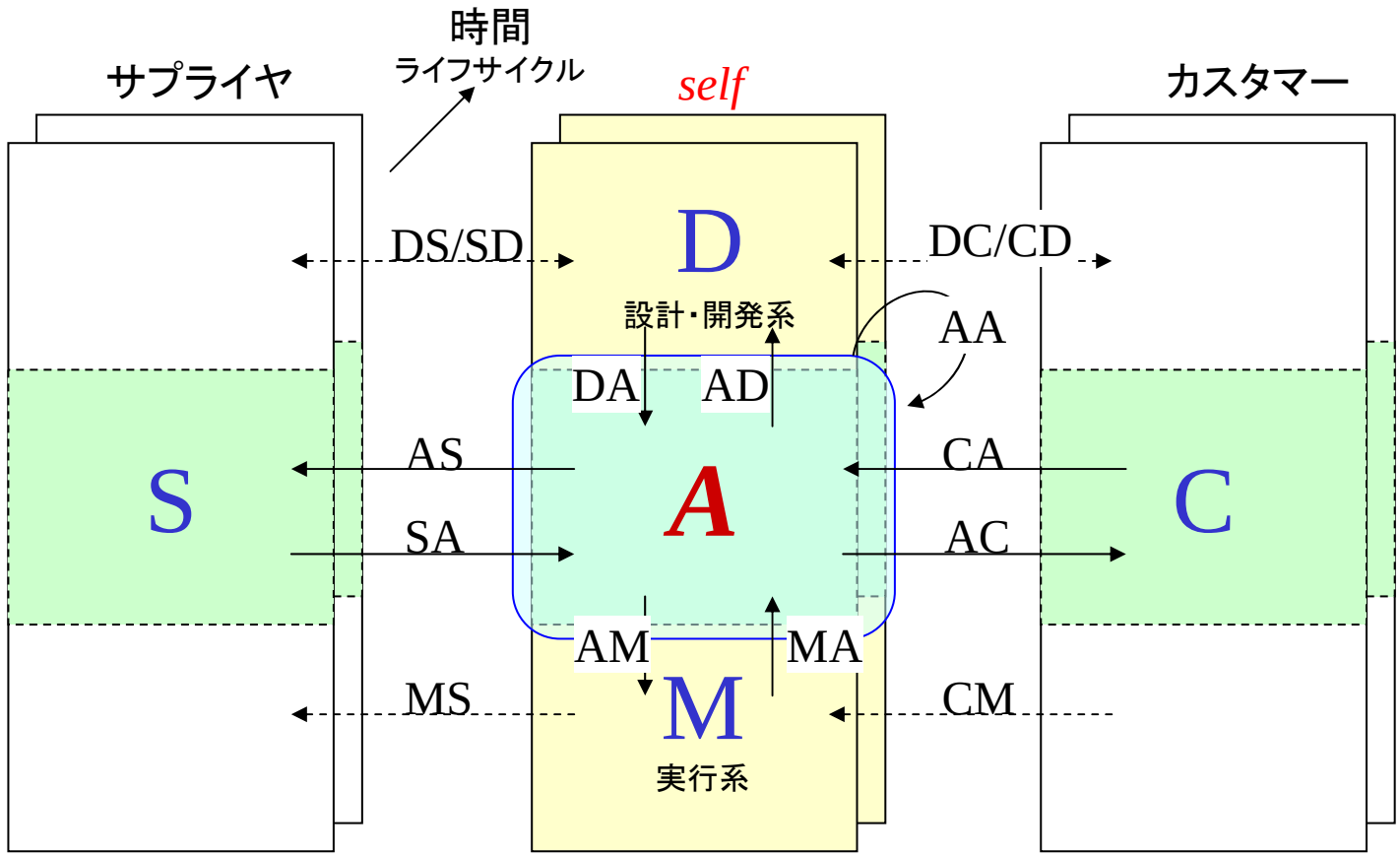
1.2 相互作用のパターン

1.3 コンポーネント間の相互作用

1.1 APSの境界とビジネスコンポーネント

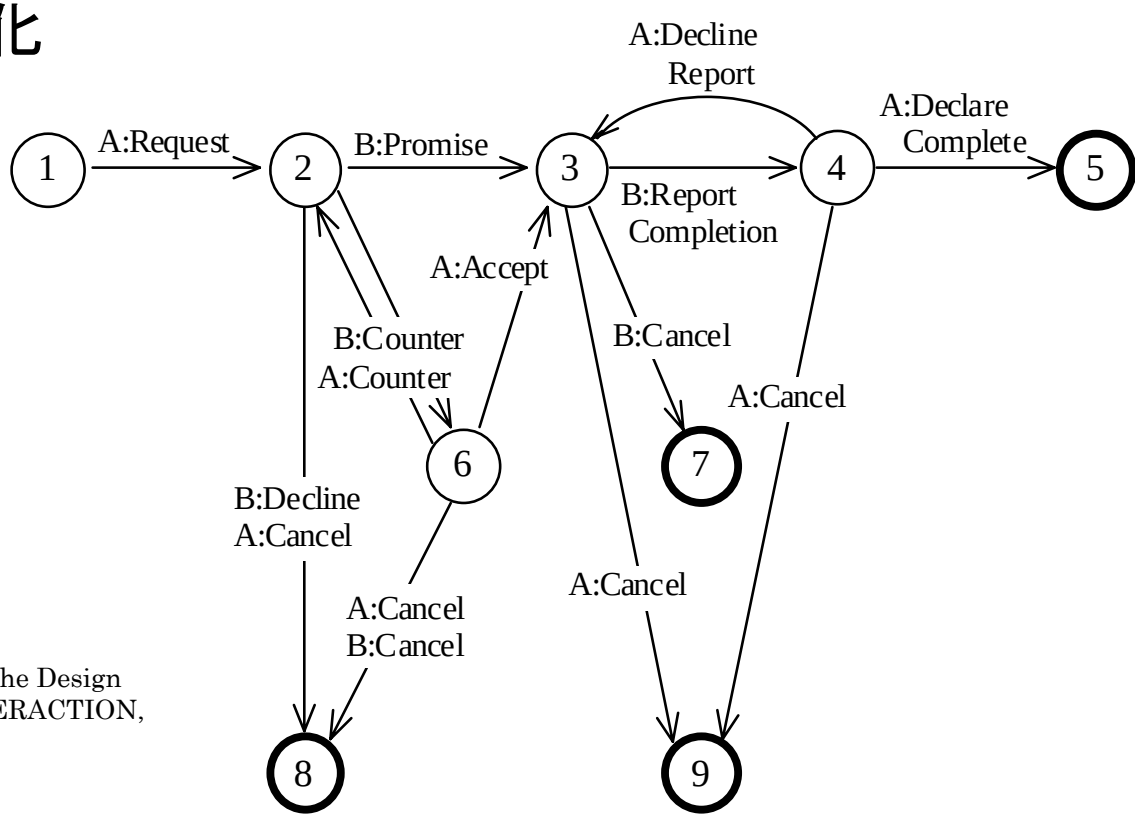
❑広義のAPS

ロビジネスコンポーネント間のインタラクション



1.2 相互作用のパターン

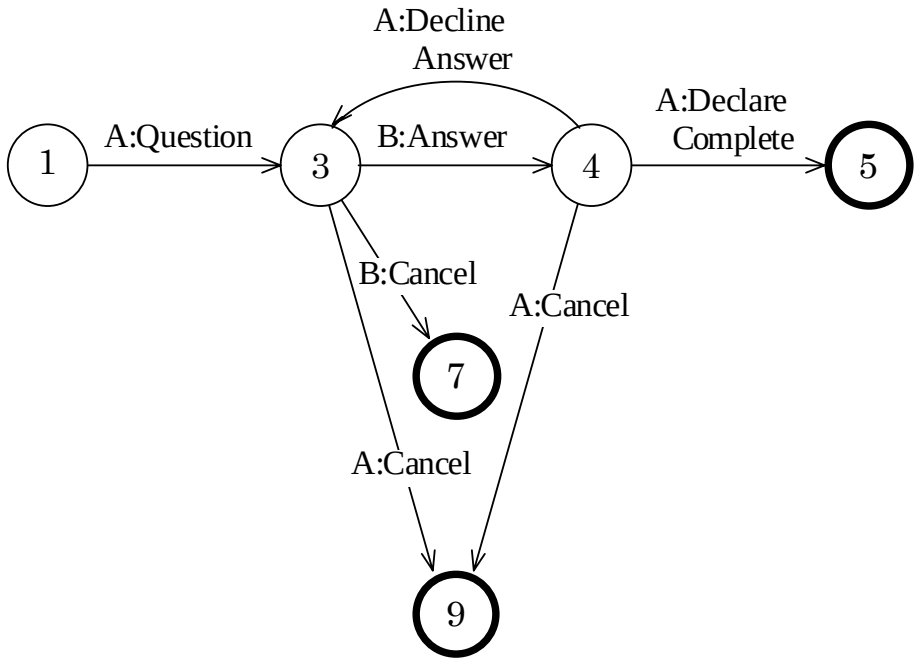
- 2人のAgent間の対話
  - A→B
  - 互いに状態を変化



Winograd, T.: "A Language/Action Perspective on the Design of Cooperative Work", HUMAN-COMPUTER INTERACTION, Vol.3, 3-30(1987-1988)

1.2 相互作用のパターン

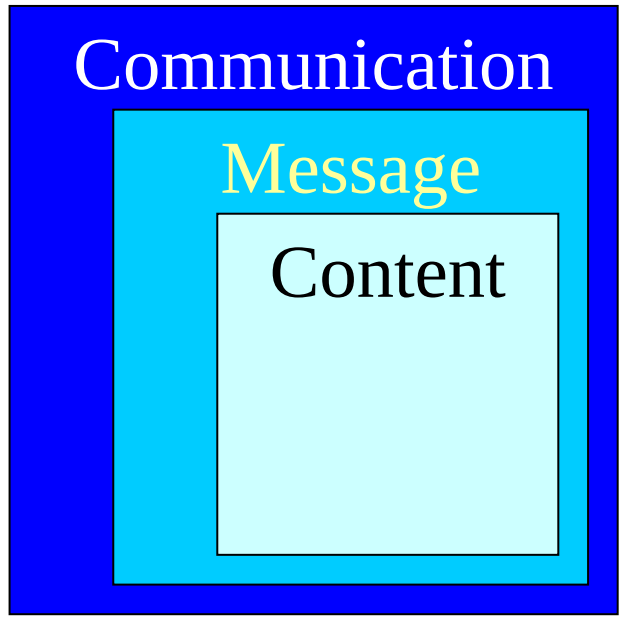
- 2人のAgent間の対話
  - A→B
  - Bは情報源



1.2 相互作用のパターン

- KQML
  - ソフトウエアエージェントの通信言語
  - Knowledge Query and Manipulation Language

```
(ask :content (geoloc lax (?long ?lat))
        :language loom
        :ontology std_geo
        :from map_agent
        :to geo_server
        :label fm-query-10-23-92-14-09-07)
(tell :content "geoloc(lax, [42.45,83.21)"
        :language standard_prolog
        :ontology std_geo)
```



Finin, T.,: KQML -- A Language and Protocol for Knowledge and Information Exchange, International Conference on Building and Sharing

of Very large-scale Knowledge Bases, Tokyo, December, 1993

2. オブジェクトモデルの位置づけ

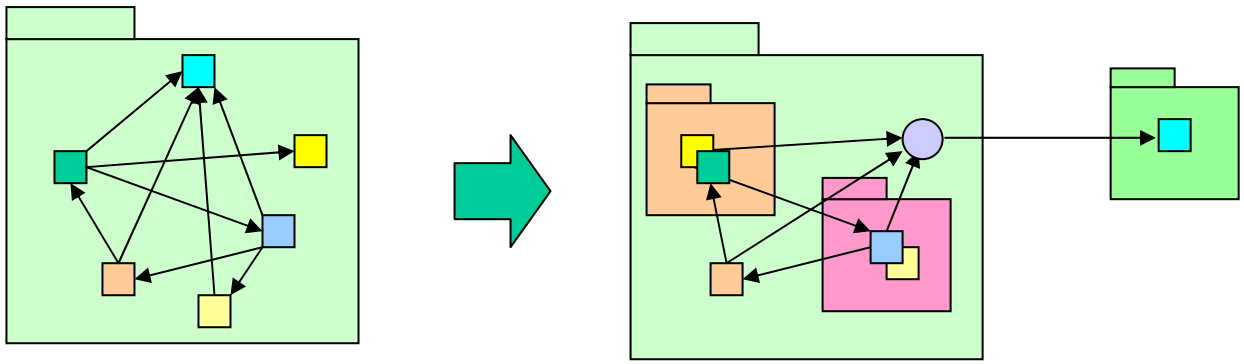
2.1 インタフェースとオブジェクトモデル

2.2 ユースケースとインタフェース

2.3 インタフェースとXMLの関係

2.1 インタフェースとオブジェクトモデル

- インタフェースの意味
- 境界のあるところにインタフェースがある（接面）
  - なぜ境界を設けるのか
    - 複雑さのコントロール
      - コミュニケーション量の最小化（coupling）
      - 機能の凝集化（cohesion）
      - オーバヘッドの減少（機能のサイズ）
    - できるだけ適切な境界を探す



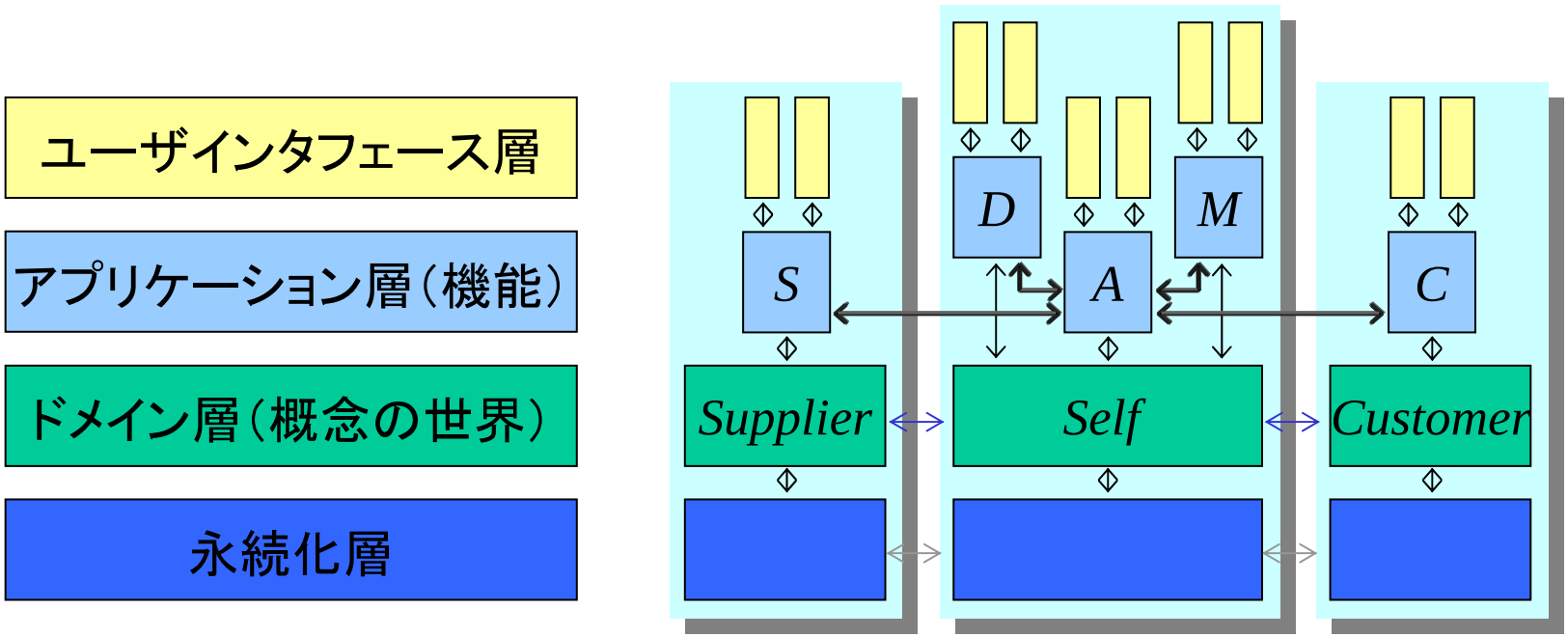
2.1 インタフェースとオブジェクトモデル

- オブジェクトモデルがなぜ必要か
  - 何をインタフェースするのか
    - 働きかけ（performative）とcontent
      - データとオブジェクト
      - ビジネスコンポーネントがそれぞれに持つ世界知識
        - ドメイン知識（オントロジー）
    - 世界知識の一部の参照（世界全部を渡せない）
  - referenceから必要な知識を効率的にたぐっていく
    - 地図が必要
  - ユースケースの追加・変更にも強い
    - インタフェースを追加するだけ
    - 既存のインタフェースに影響を及ぼさない

ユースケースに対応

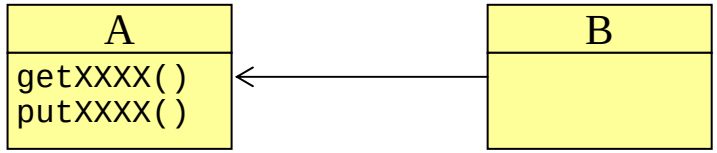
2.1 インタフェースとオブジェクトモデル

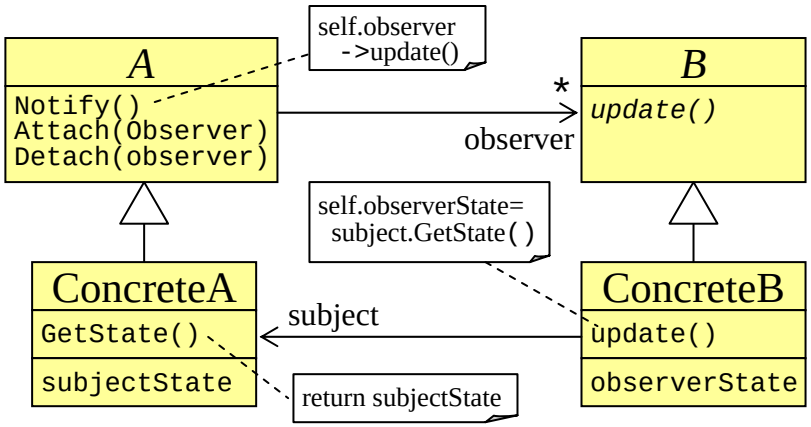
- ドメインモデルとソフトウェアアーキテクチャ
  - 自己組織化
    - Cohesionの高いモジュールへ
  - 実装から独立して議論したい

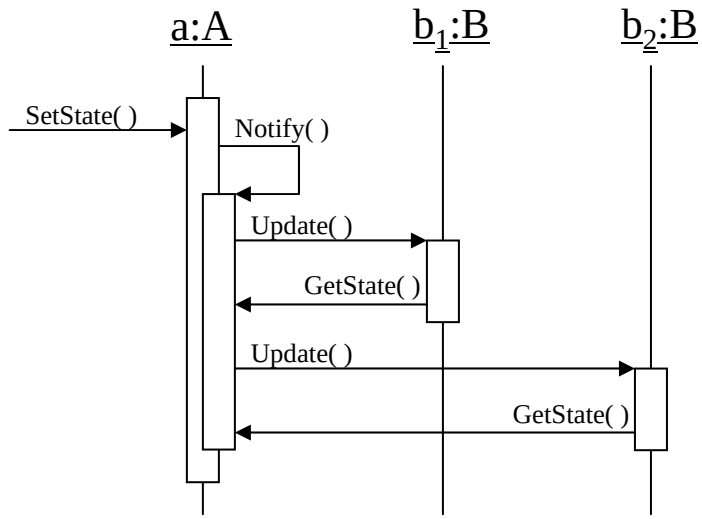


2.1 インタフェースとオブジェクトモデル

- 境界のつなぎ方
  - 一方向の可視性
    - read only
  - Observer pattern（MVCモデル）
    - モジュール間の相互作用

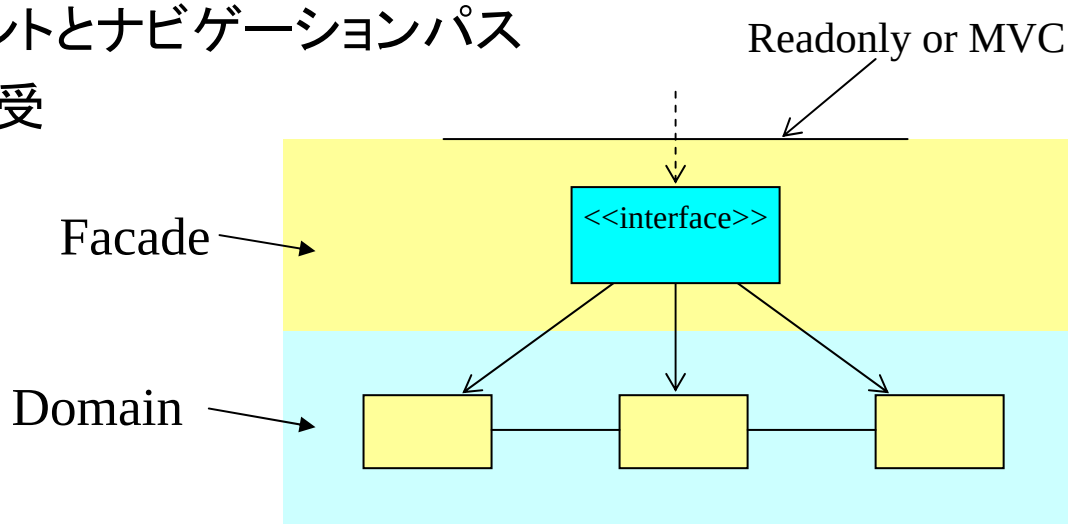






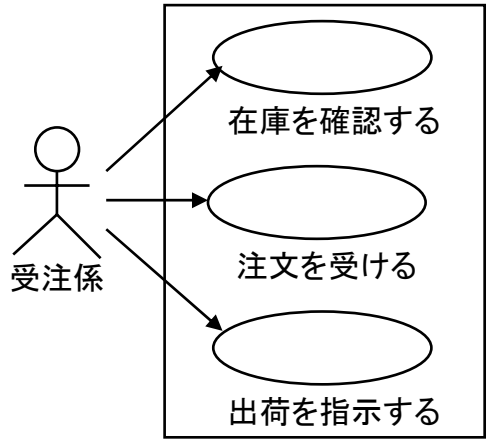
2.1 インタフェースとオブジェクトモデル

- データの見せ方
  - Façade
    - モジュールが持っている内部データ構造ではなくて，どう見せるか
    - 効率や実装上の都合
  - プロトコル
    - 全体のデータ構造を一度に渡すのは困難
    - エントリポイントとナビゲーションパス
    - イベントの授受



2.2 ユースケースとインタフェース

- ユースケース
  - モデルの機能的側面の記述の一部
  - モデルの静的側面を検証，補強するものとして使う



ユースケース名
ユースケースの目的
ユースケースのアクタ
変換内容
　　事前条件（pre-condition）
　　主系列（main sequence）
　　事後条件（post-condition）
　　代替系列（alternative sequence）
備考

2.2 ユースケースとインタフェース

❑ユースケース記述の例

ユースケース名：**内示注文を受ける**
目的：予定の取引をあらかじめ把握し，製造準備をする。
アクタ：営業
事前条件：なし
手順：

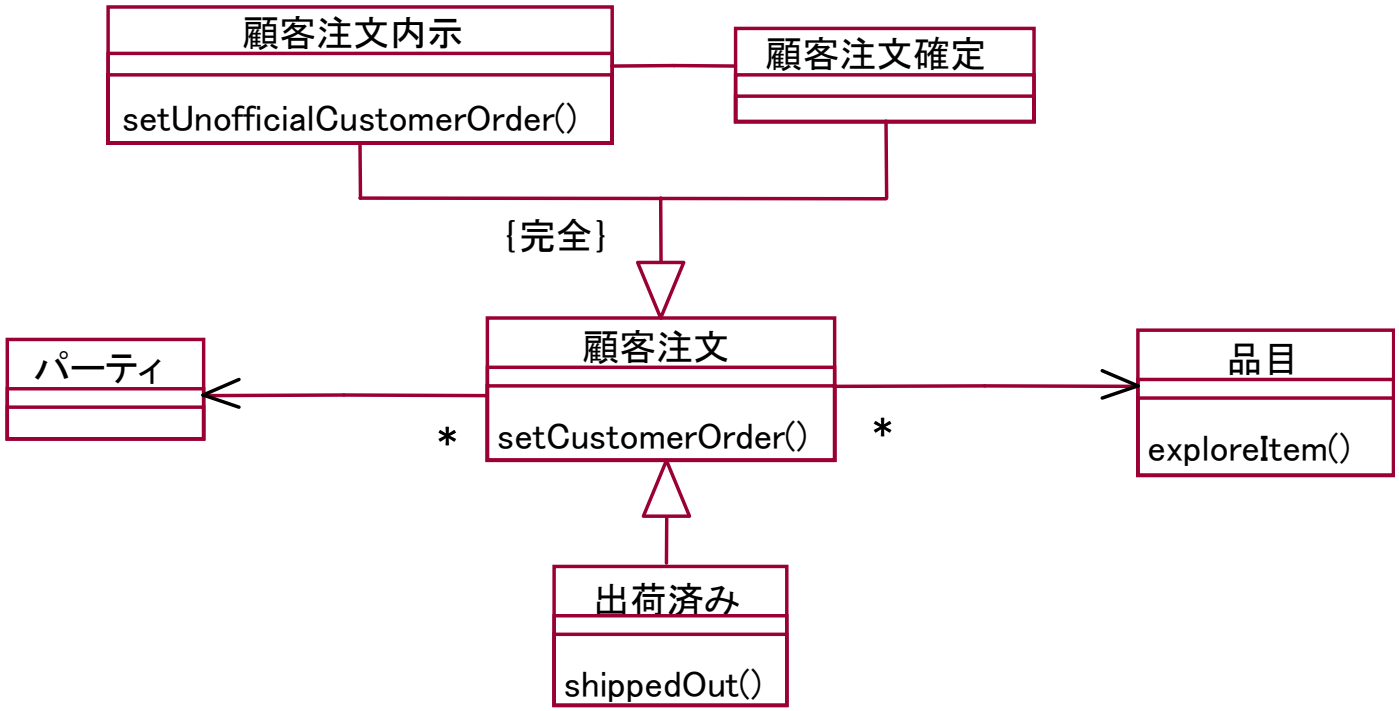
① アクタは，システムに新たな内示があることを伝える。
② システムは，アクタに内示の内容を求める。内示の内容は{顧客，品目，数量，納品場所，希望納期}とする。ただし希望納期は当日＋10日以降翌々月までとする。
③ システムは，アクタに内容の確認を求める。
④ アクタは，（顧客に内示の内容を確認し）システムに確認したことを伝える。
⑤ システムは，アクタに内示を記録したことを伝える。

事後条件：内示内容が記録されている。
代替系列：

A. 手順②にて，顧客が妥当でない場合は，…

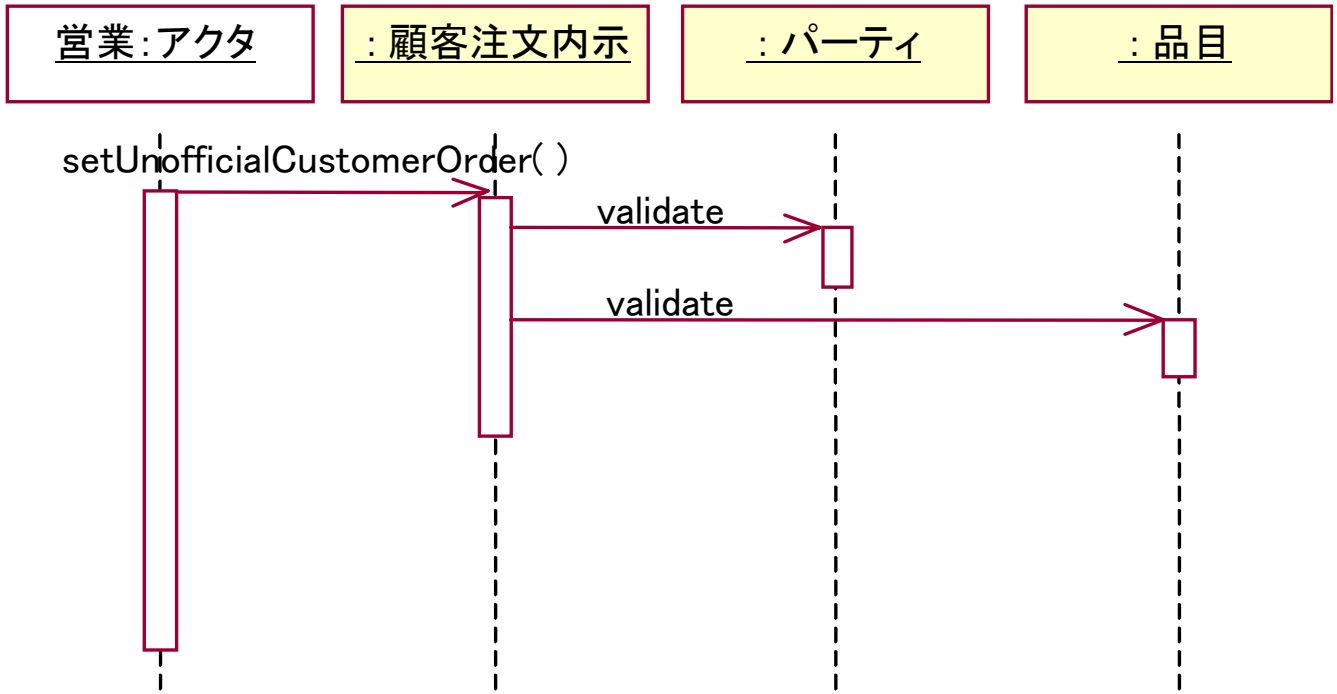
2.2 ユースケースとインタフェース

❑静的モデル



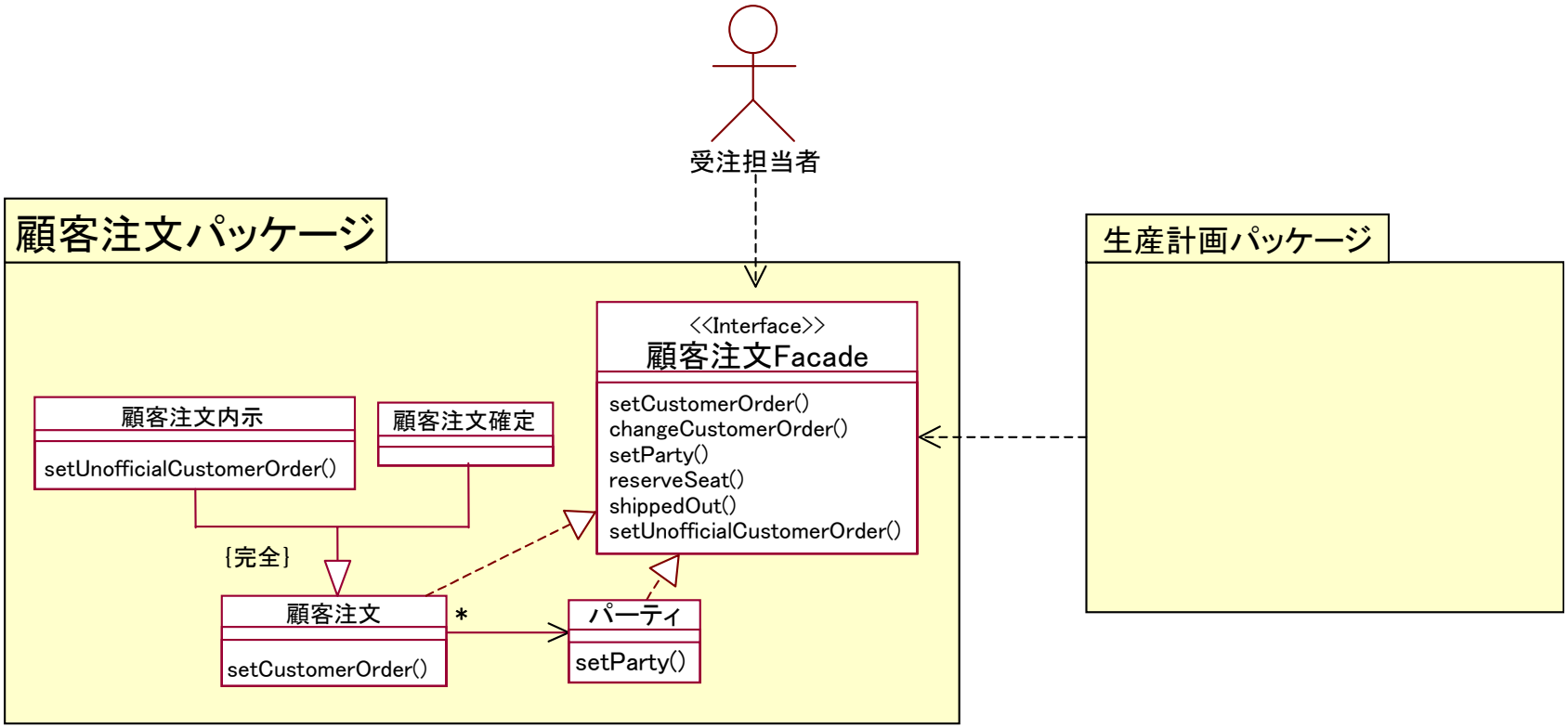
2.2 ユースケースとインタフェース

- ユースケース「内示を受ける」のシーケンス図
  - シナリオレベル



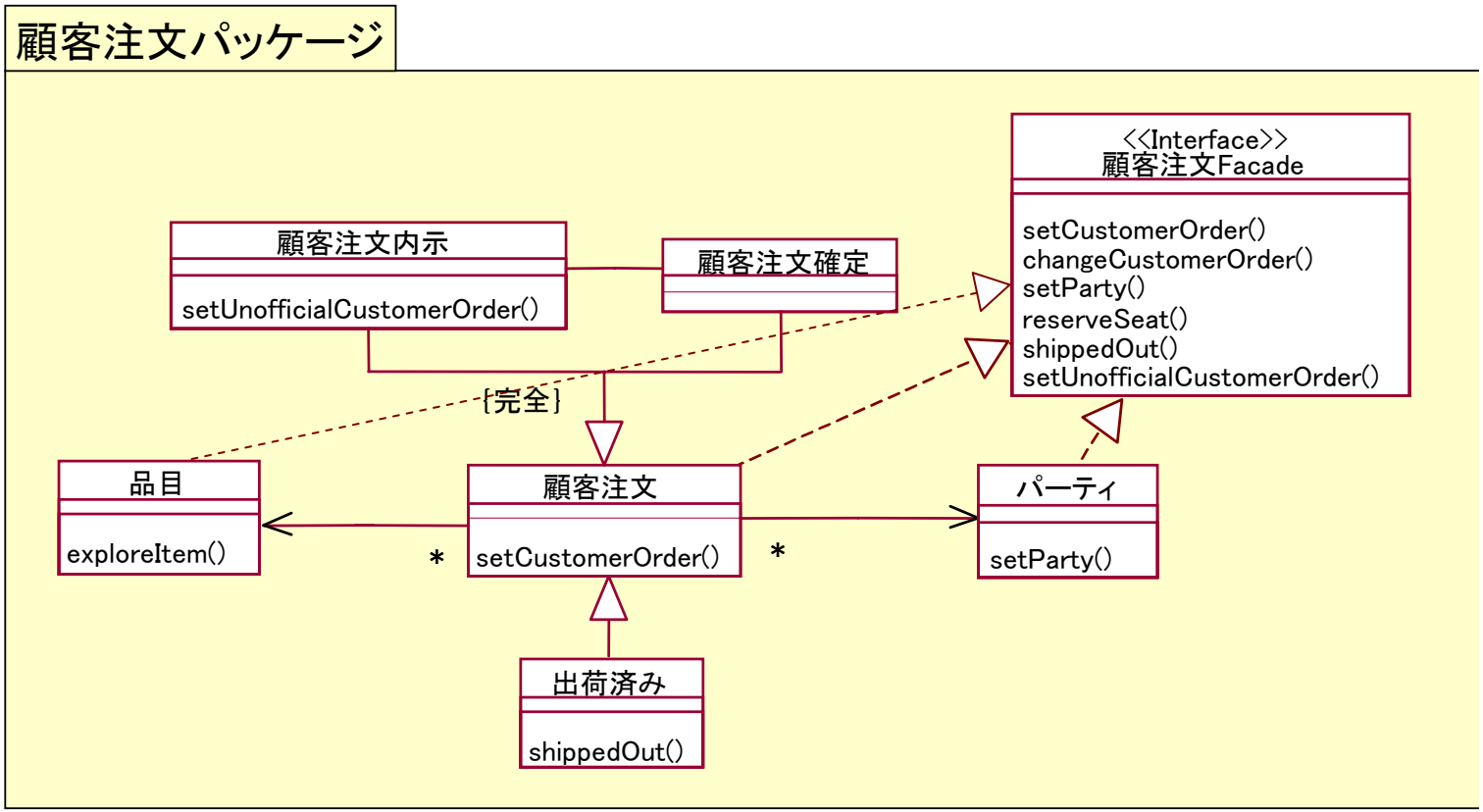
2.3 インタフェースとXMLの関係

❑ファサードとインタフェース



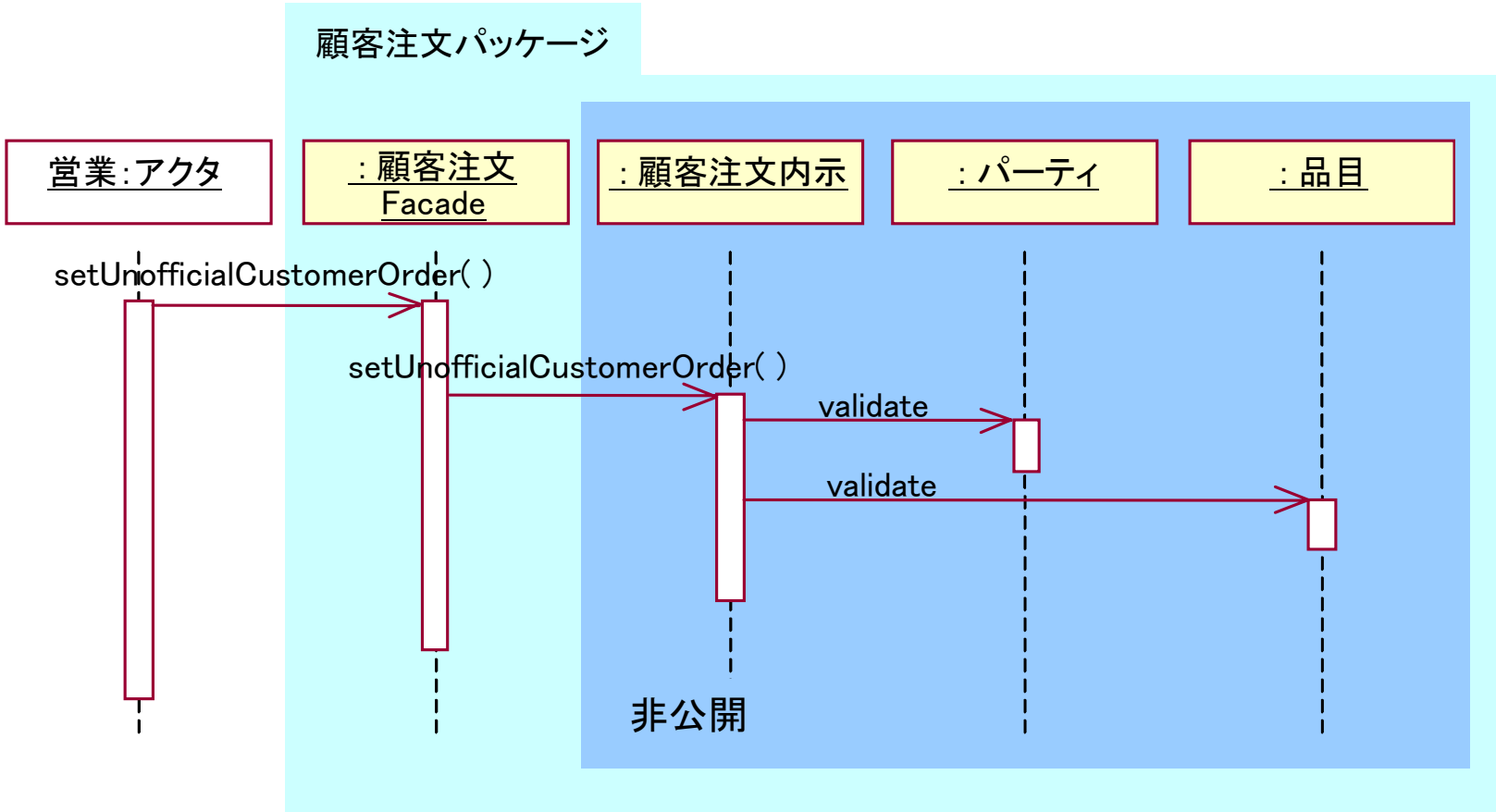
2.3 インタフェースとXMLの関係

❑ファサードとインタフェース



2.3 インタフェースとXMLの関係

❑ファサードとインタフェース



2.3 インタフェースとXMLの関係

- XMLによるインタフェースの定義

```
<order name="Z001">
  <customer ref="パーティ">
    <location ref="納品場所"/>
  </customer>
  <item ref="品目"/>
  <qty value="数量"/>
  <duetime><time value="希望納期"/></duetime>
</order>
```

3. ユースケース記述

3.1 現状の生産管理システム

3.2 新しい生産管理システム

3.1 現状の生産管理システム

- 現状の要求を満たしつつ，新しい要求にも対応できること

現在の生産管理技術
- 厚板
- 自動車
- 乳製品
- ローリング計画

新しい生産管理技術
- 集約（グループテクノロジー）
- 引き当て管理（座席予約）
- 仕様決定管理
- 製品切替管理
- 計画連携

グランドデザイン「個別アプリケーションと技術要素」から推定

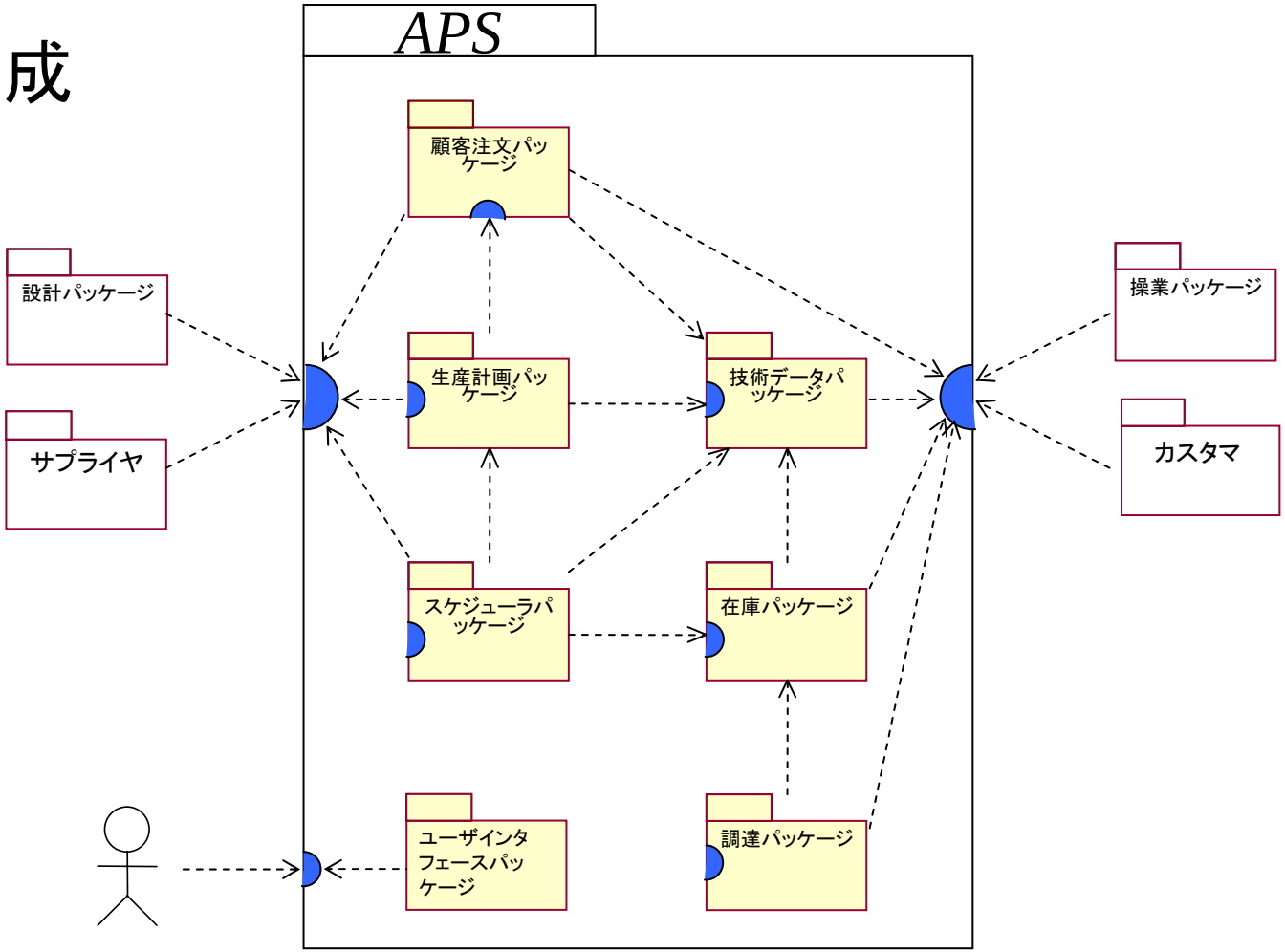
4. オブジェクトモデル

4.1 静的モデル

4.2 動的モデル

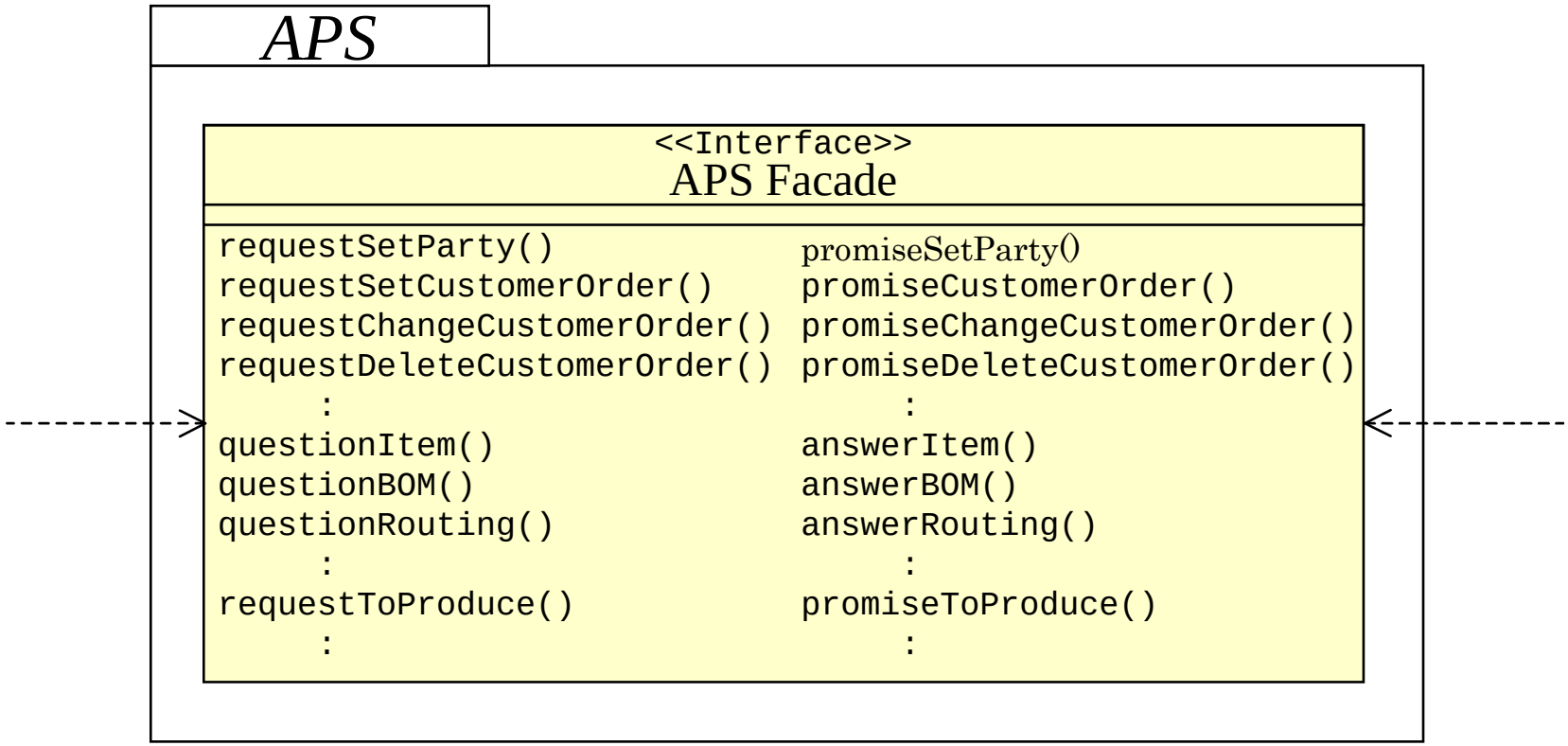
4.1 静的モデル

❑全体構成



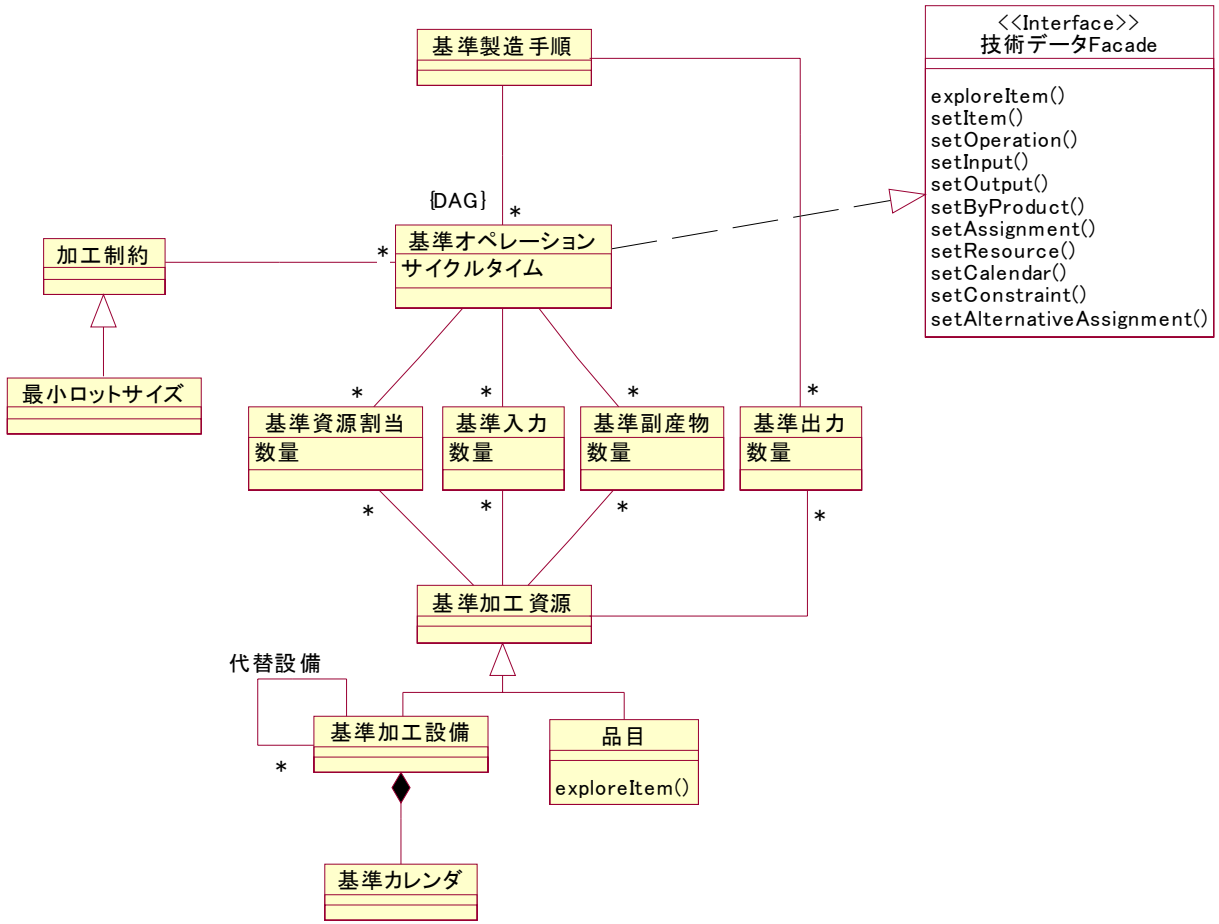
4.1 静的モデル

- APSファサード



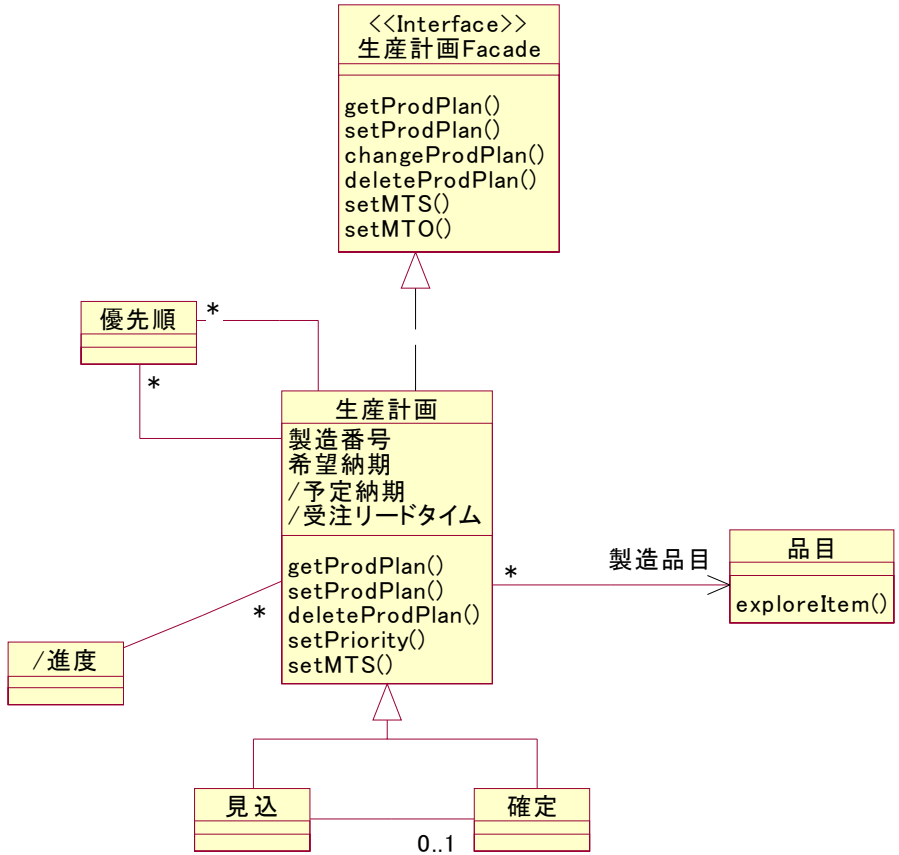
4.1 静的モデル

❑技術データパッケージ



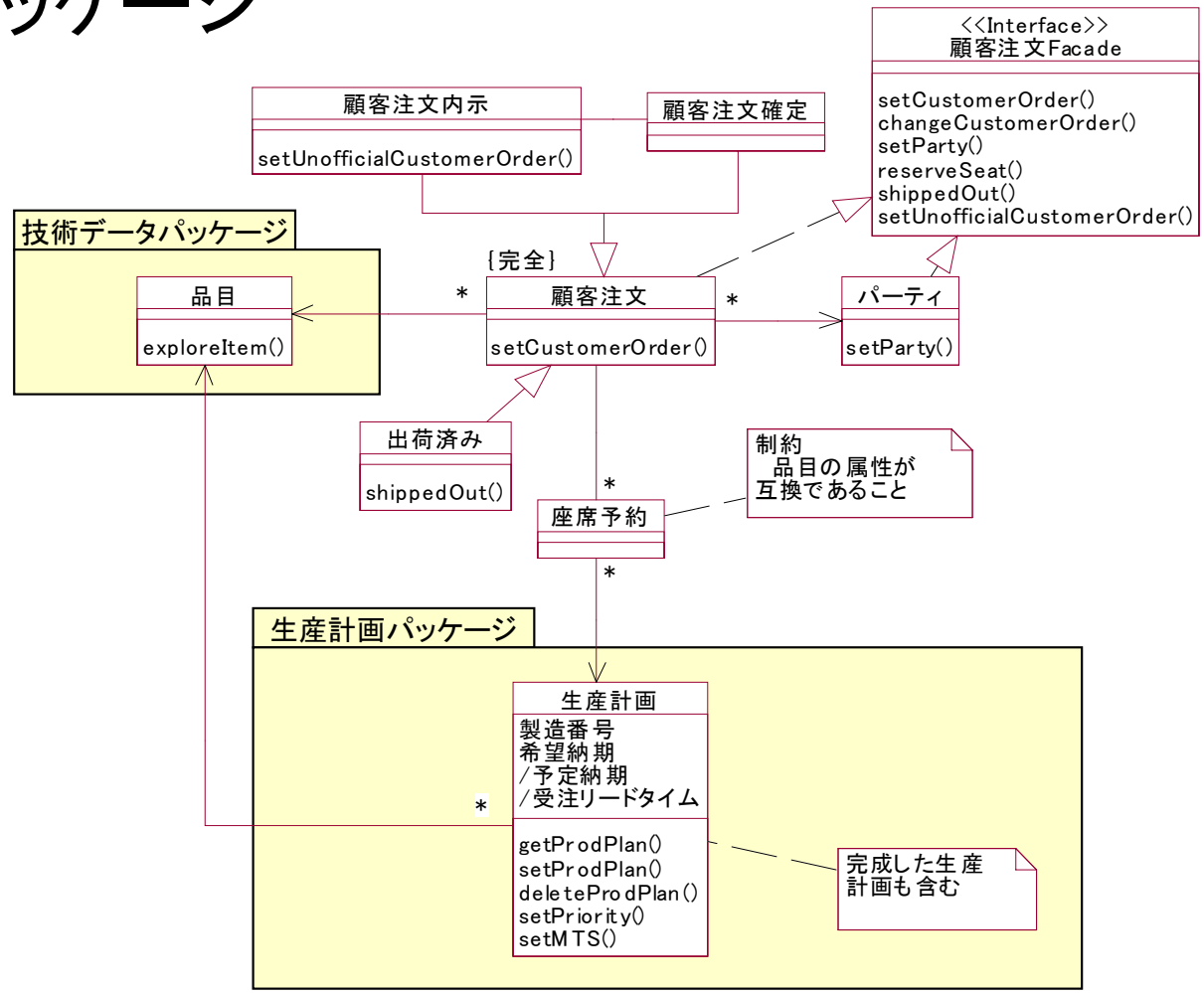
4.1 静的モデル

❑生産計画パッケージ



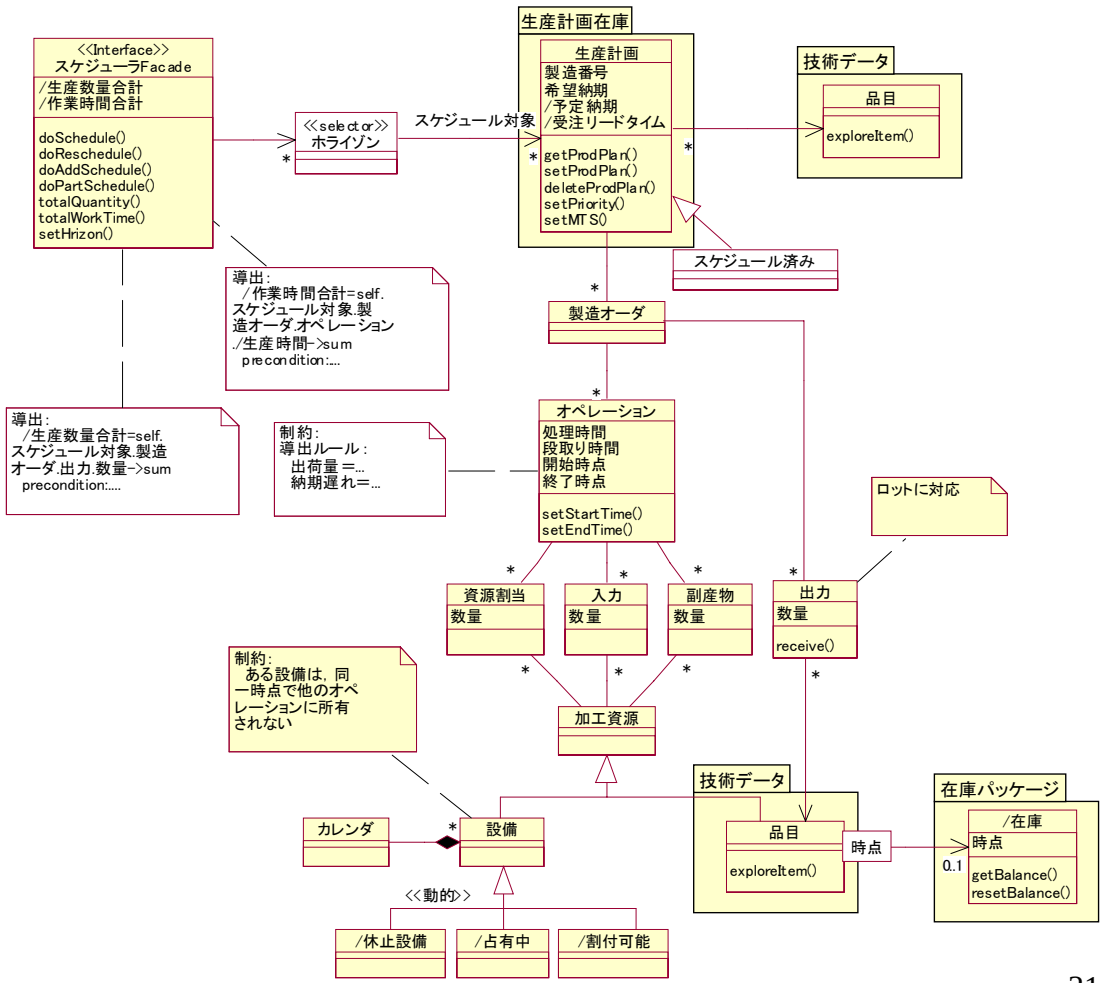
4.1 静的モデル

❑顧客注文パッケージ



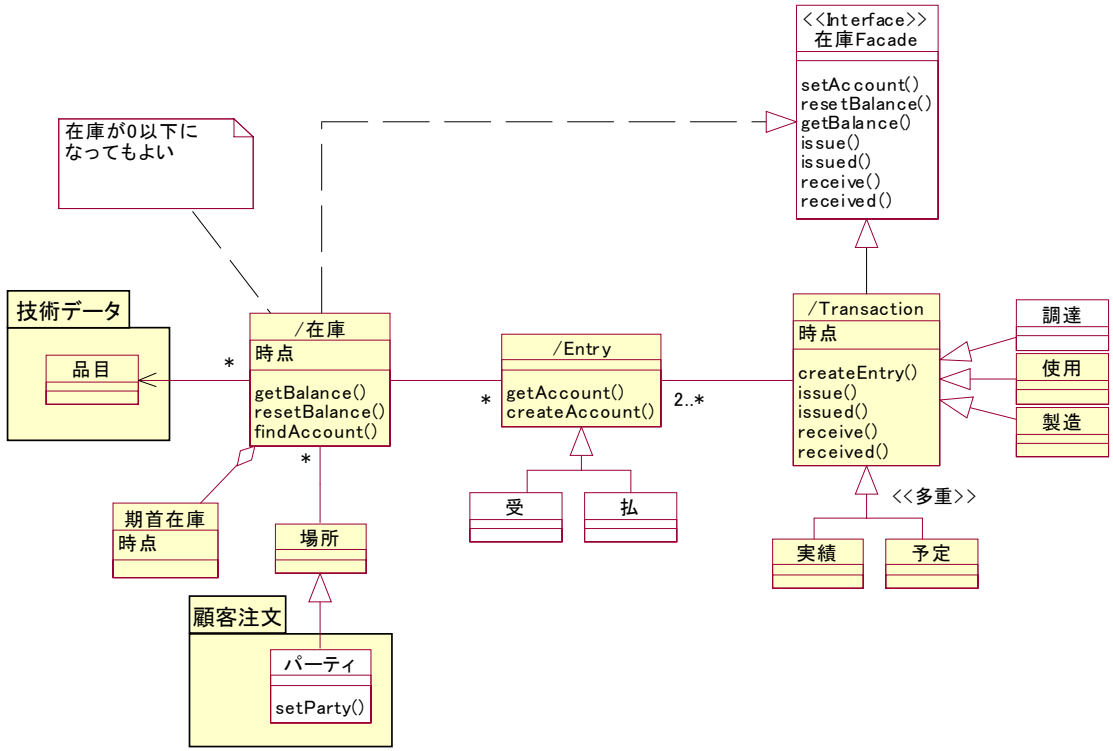
4.1 静的モデル

❑スケジューラパッケージ



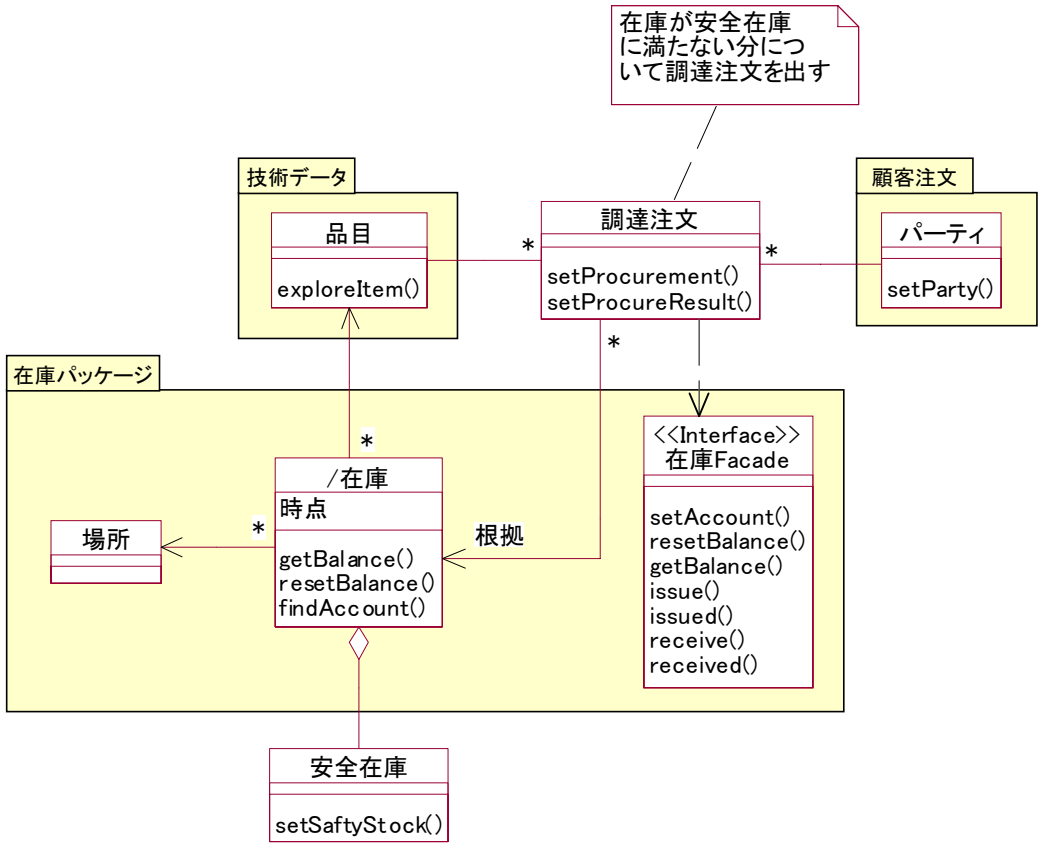
4.1 静的モデル

❑在庫パッケージ



4.1 静的モデル

❑調達パッケージ



5. インタフェースの定義

5.0 メッセージとメソッドコール

5.1 Customerとのインタフェース

5.2 Supplierとのインタフェース

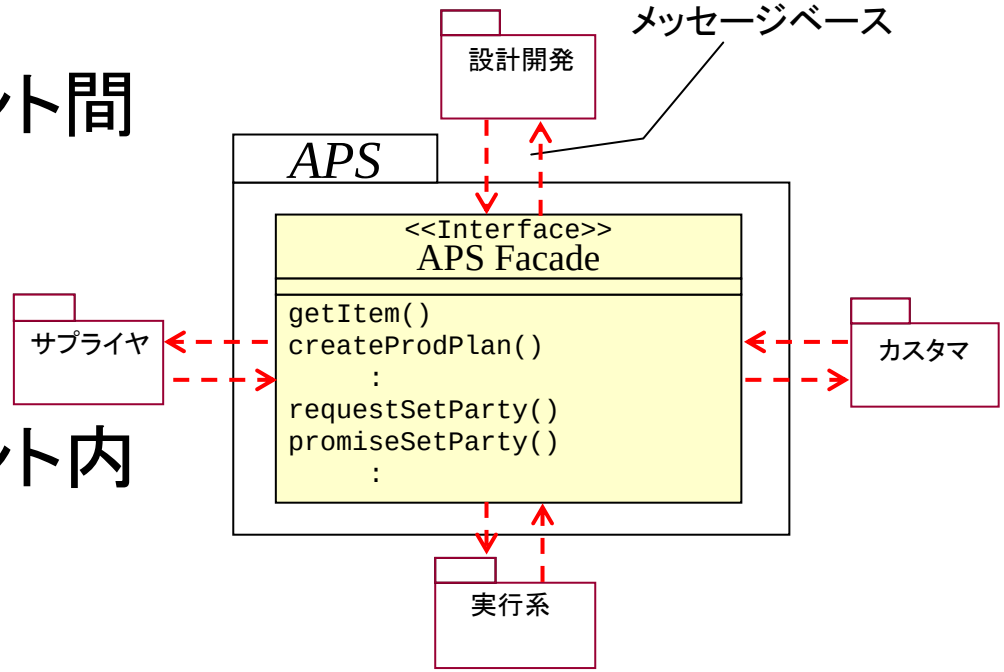
5.3 設計開発機能とのインタフェース

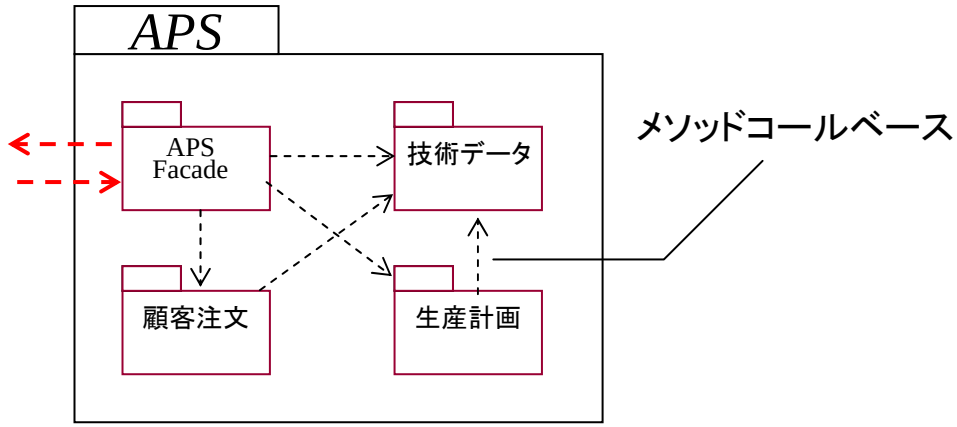
5.4 実行系とのインタフェース

5.5 APS内部のインタフェース

5.0 メッセージとメソッドコール

- ビジネスコンポーネント間
  - メッセージ
    - 要求メッセージ
    - 返答メッセージ
- ビジネスコンポーネント内
  - メソッドコール
    - 呼び出し
    - return





5.0 メッセージとメソッドコール

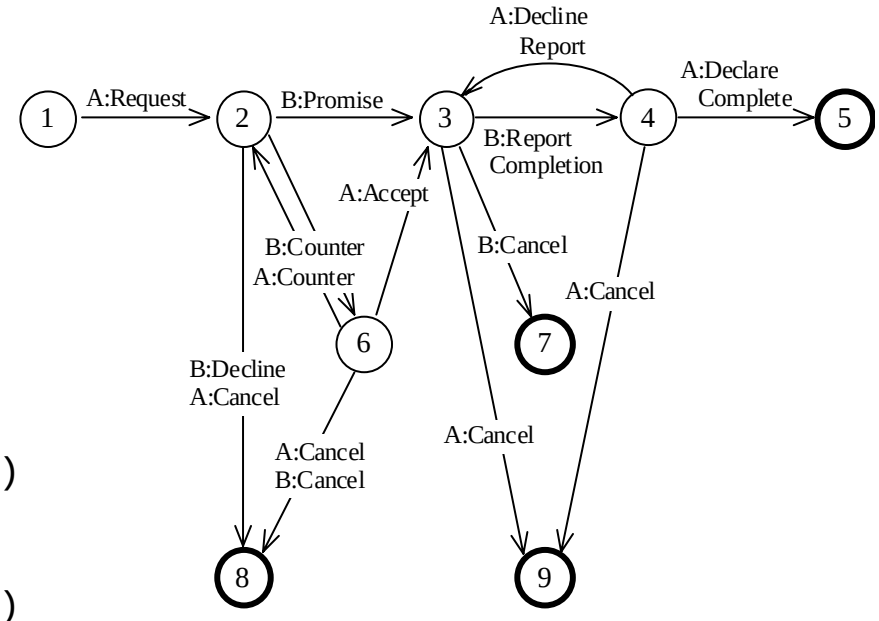
❑あり得る相互作用

☐明示的定義

```
requestCustomerOrder()
promiseCustomerOrder()
```

☐暗黙的定義

```
requestCancelCustomerOrder()
counterCustomerOrder()
anticounterCustomerOrder()
acceptCounterCustomerOrder()
cancelCustomerOrder()
cancelCounterCustomerOrder()
requestCancelCounterCustomerOrder()
requestCancelPerformCustomerOrder()
cancelPerformCustomerOrder ()
declineCustomerOrder()
promiseCounterCustomerOrder()
declineReportCustomerOrder()
declareCompleteCustomerOrder()
```

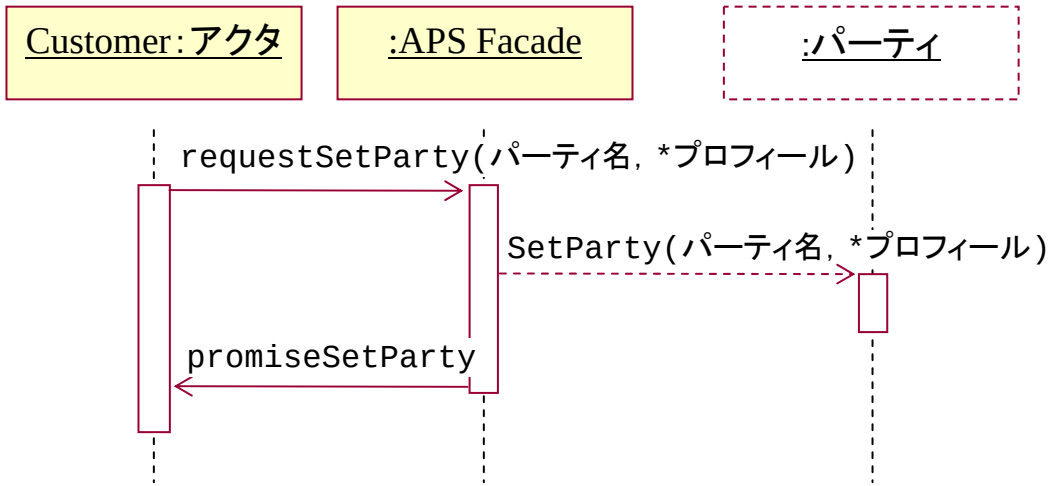


5.1 Customerとのインタフェース

❑CA/AC

ロ顧客の設定

CA	AC
requestSetParty(パーティ名, *プロフィール)	promiseSetParty()
requestChangeParty(パーティ名, *プロフィール)	promiseChangeParty()
requestDeleteParty(パーティ名)	promiseDeleteParty()
requestGetParty(パーティ名)	promiseGetParty()



5.1 Customerとのインタフェース

❑CA/AC

ロ顧客注文要求

注文番号=requestSetCustomerOrder(パーティ, 品目群, 用途・使用条件値, 数量, 納品場所, 希望納期, 受注日)
promiseCustomerOrder()

requestChangeCustomerOrder(注文番号, [パーティ], [品目群], [用途・使用条件値], [数量], [納品場所], [希望納期], [受注日])
promiseChangeCustomerOrder()

requestDeleteCustomerOrder(注文番号)
promiseDeleteCustomerOrder()

注文番号=requestSetUnofficialCustomerOrder(パーティ, 品目群, 用途・使用条件値, 数量, 納品場所, 希望納期, 受注日)
promiseSetUnofficialCustomerOrder()

5.1 Customerとのインタフェース

- CA/AC
  - 納期回答
    `promiseDuedate(注文番号)`
  - 注文品の受け取りの通知
    `notifyReceipt(品目群, 用途･使用条件値, 数量, 受領場所, 受領日)`
  - 提案要求を出す
    `requestForProposal(*要求仕様)`
    `submitProposal(*提案)`

5.2 Supplierとのインタフェース

- SA/AS
  - CA/ACと同様

5.3 設計開発機能とのインタフェース

- DA/AD
  - 製品仕様の問い合わせ
    - 品目を照会する
      - questionItem(品目群, 用途・使用条件値)
      - answerItem(Boolean)
    - 品目展開
      - questionBOM(品目群, 用途・使用条件値, ライン)
      - answerBOM(/*品目構成)
    - 製造手順の提示
      - questionRouting(品目群, 用途・使用条件値, ライン)
      - answerRouting(/*製造手順)
    - 工程ごとのオペレーション
      - questionOperation(品目群, 用途・使用条件値, ライン, 工程)
      - answerOperation(*オペレーション)

5.4 実行系とのインタフェース

- MA/AM
  - AM
    - 作業指示を行う
      - `requestToProduce(ホライゾン)`
      - `promiseToProduce()`
    - 出荷指示を行う
      - `requestToShip(顧客注文)`
      - `promiseToShip()`
    - 進度を問う
      - `questionRouting(品目群, 用途・使用条件値, ライン)`
      - `answerProgress(生産計画, 進度/工程)`
  - MA
    - 製造実績を報告する
      - `reportProduction(生産計画, ライン, 工程, オペレーション, 品目群, 用途・使用条件値, 数量, 実績発生日)`

5.5 APS内部のインタフェース

❑技術データパッケージ

□品目群の定義

setItem(品目群名，用途・使用条件種リスト)

setDefaultItem(品目群名，デフォルト用途・使用条件値リスト)

setItem(品目コード，品目名)

□オペレーションの定義

setOperation(製造品目群，ライン，工程，オペレーション，用途・使用条件値，製造数量，サイクル時間，段取り時間，歩留まり，加工費用)

setInput(オペレーション，投入品目，その用途・使用条件値，数量)

setResource(オペレーション，基本加工資源，数量)

setConstraint(オペレーション，加工制約ルール)

□加工設備の定義

setCalendar(基本加工資源，設備能力，基本カレンダ)

setResourceAggregation(基本加工資源，集約方法)

□問い合わせ

5.5 APS内部のインタフェース

❑生産計画パッケージ

□生産計画の登録，変更，削除

製番=createProdPlan(製造品目群，用途・使用条件値，計画納期，計画数量，仮／確定)

setProdPlan(製番)

changeProdPlan(製番，製造品目群，用途・使用条件値，計画納期，計画数量，仮／確定)

deleteProdPlan(製番)

□生産計画の分割，集約

生産計画=splitProdPlan(生産計画)

生産計画=aggregateProdPlan(生産計画のコレクション)

□生産ロット編成

ロット=makeLotPlan(製番，製造品目群，用途・使用条件，計画数量)

□生産計画の照会

reference=getProdPlan([製番]，[品目群]，[期間])

生産計画のコレクション=getProdPlanBySimilarity(生産計画のコレクション，品目群，用途・使用条件値，期間)

5.5 APS内部のインタフェース

- 生産計画パッケージ
  - 生産ロットのシーケンシング
    - `doSequencing(ロットのコレクション)`
    - `changeSequencing(ロットのコレクション)`

5.5 APS内部のインタフェース

❑顧客注文パッケージ

□注文の登録，変更，削除

注文番号=setCustomerOrder(パーティ，品目群，用途・使用条件値，数量，納品場所，希望納期，受注日)

注文番号=setUnofficialCustomerOrder(パーティ，品目群，用途・使用条件値，数量，納品場所，希望納期，受注日)

changeCustomerOrder(注文番号，[パーティ]，[品目群]，[用途・使用条件値]，[数量]，[納品場所]，[希望納期]，[受注日])

deleteCustomerOrder(注文番号)

□顧客または仕入先の登録

パーティ=setParty(パーティ名，プロフィール)

パーティ=changeParty(パーティ名，プロフィール)

パーティ=deleteParty(パーティ名)

パーティ=getParty(パーティ名)

□類似性の基準を示して，顧客注文リストを表示する

$questionCustomerOrderBySimilarity(対象期間，品目群，類似性基準)

5.5 APS内部のインタフェース

- 顧客注文パッケージ
  - 座席を予約する
    - `reserveSeat(製番, 注文番号)`
    - `unreserveSeat(製番, 注文番号)`
  - 顧客注文が完了したことを記録する
    - `shippedOut(注文番号, 出荷日)`

5.5 APS内部のインタフェース

- スケジューラパッケージ
  - スケジューリングホライゾンに入れる生産計画を設定
    - ホライゾン=setHorizon(生産計画のコレクション，対象とする期間)
    - addToHorizon(ホライゾン，生産計画)
    - removeFromHorizon(ホライゾン，生産計画)
  - 生産計画の数量と作業時間の合計を導出
    - 数量=totalQuantity(生産計画のコレクション)
    - 時間=totalWorkTime(生産計画のコレクション)
  - スケジューリングする
    - doSchedule(ホライゾン)
    - doReschedule(ホライゾン)
    - doAddSchedule(生産計画)
    - doPartSchedule(ホライゾン，スケジューリング条件)

5.5 APS内部のインタフェース

- 在庫パッケージ
  - 場所(倉庫)を設定する
    `setStockLocation(場所名)`
  - 期首在庫を設定する
    `setInitialBalance(品目群, 用途・使用条件, 在庫量, 期首)`
    `resetBalance(品目群, 用途・使用条件, 在庫量)`
  - 指定日における指定品目の有効在庫量
    `getBalance(品目群, 用途・使用条件, 指定日)`
  - 入出庫の予定を記録
    `issue(品目群, 用途・使用条件値, 場所, 数量, 予定, 調達／仕様／製造の別)`
    `receive(品目群, 用途・使用条件値, 場所, 数量, 予定, 調達／仕様／製造の別)`
    `issued(品目群, 用途・使用条件値, 場所, 数量, 実績, 調達／仕様／製造の別)`
    `received(品目群, 用途・使用条件値, 場所, 数量, 実績, 調達／仕様／製造の別)`

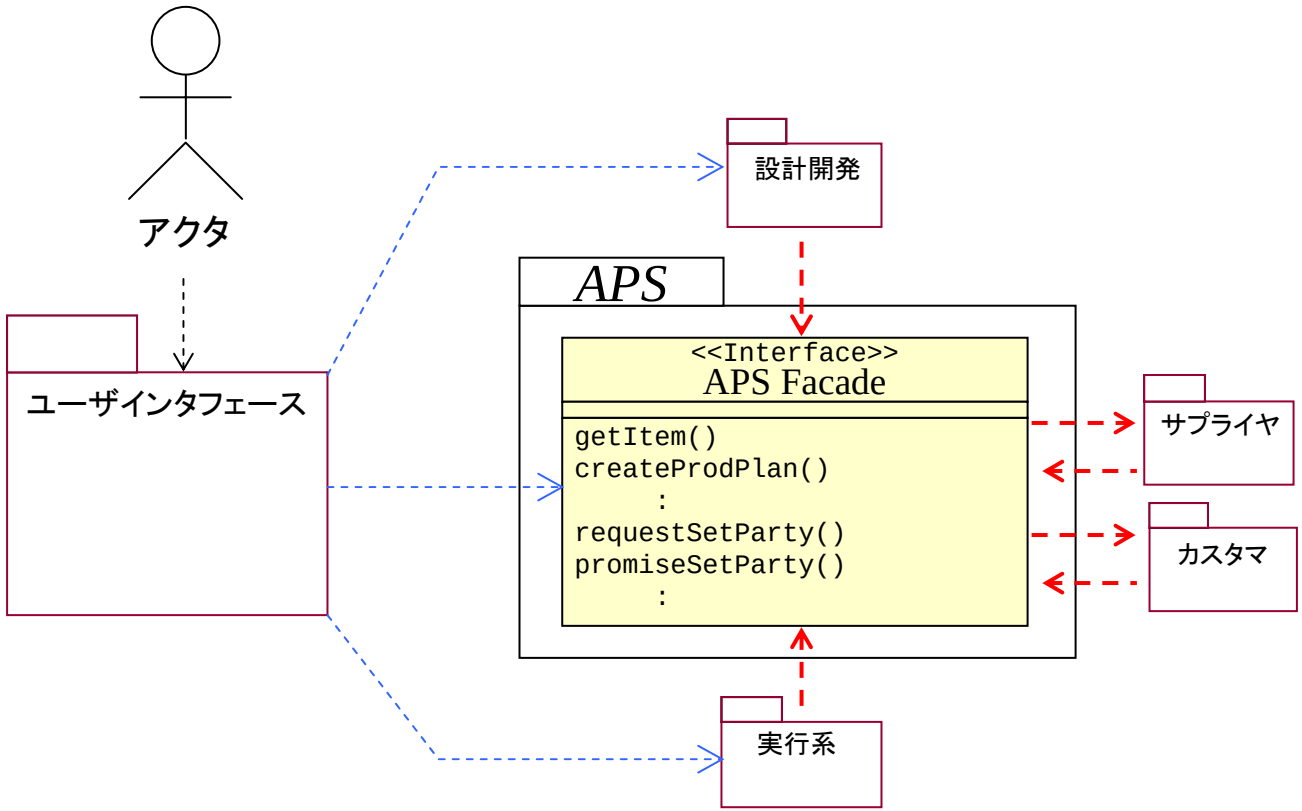
5.5 APS内部のインタフェース

- 調達パッケージ
  - 安全在庫量を設定する

    `setSaftyStock(品目群, 用途･使用条件, 安全在庫量)`

5.6 人とのインタフェース

❑ユーザインタフェース層から



おわりに

- なぜオブジェクトモデルなのか
  - インタフェースそのものの定義
    - ファサード
    - プロトコル
  - 目に見える形で
    - ナビゲーションのための地図
- 新しい生産管理システムのインタフェースの実装に向けて
  - グランドデザインとのすりあわせ
  - XMLコア技術とのすりあわせ