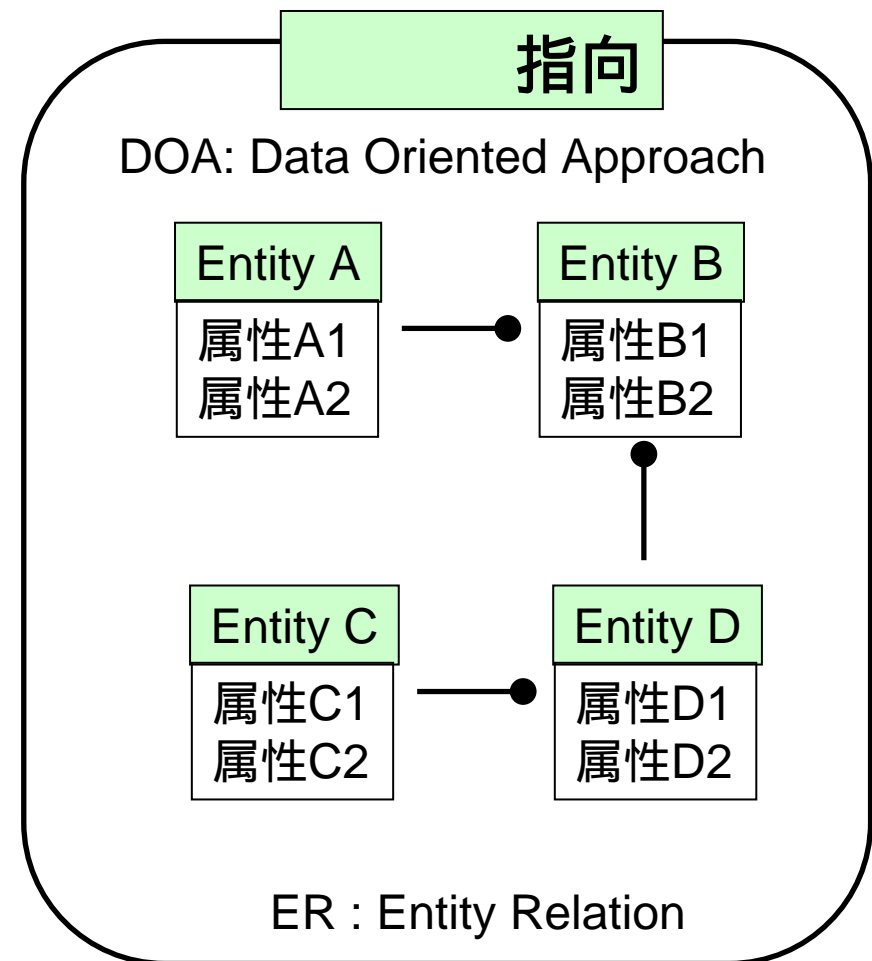
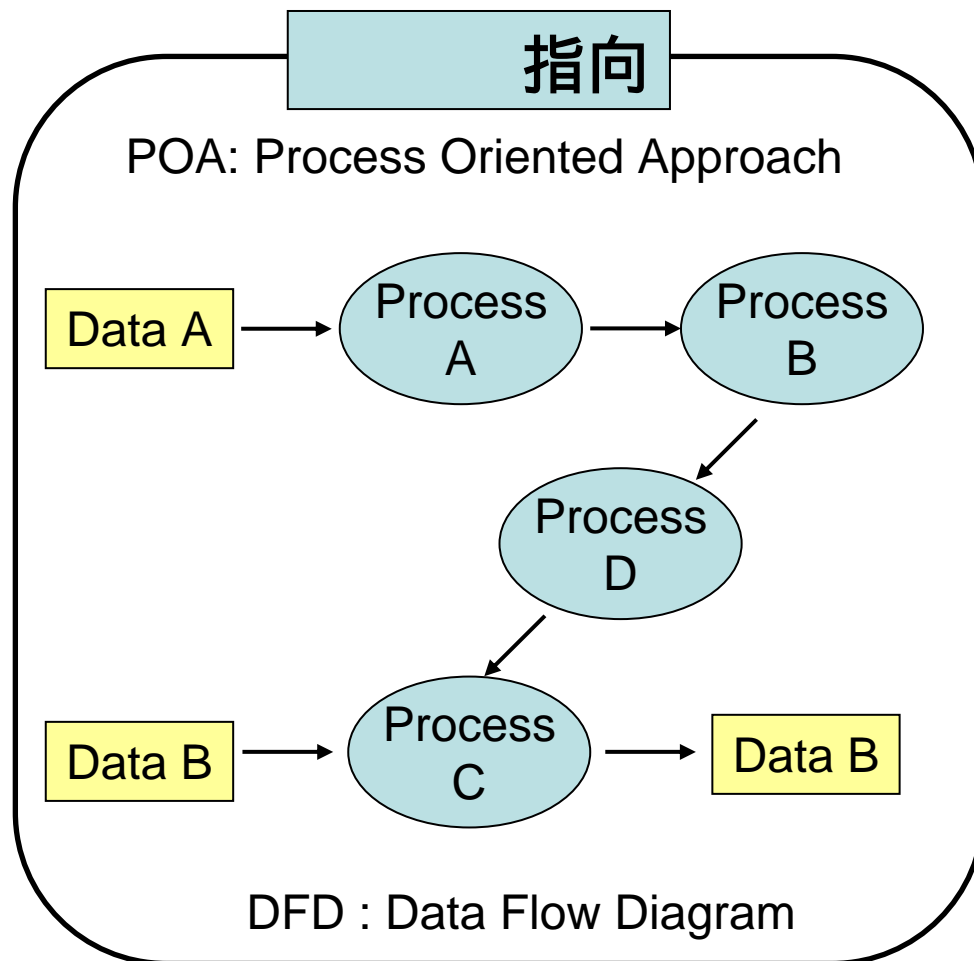
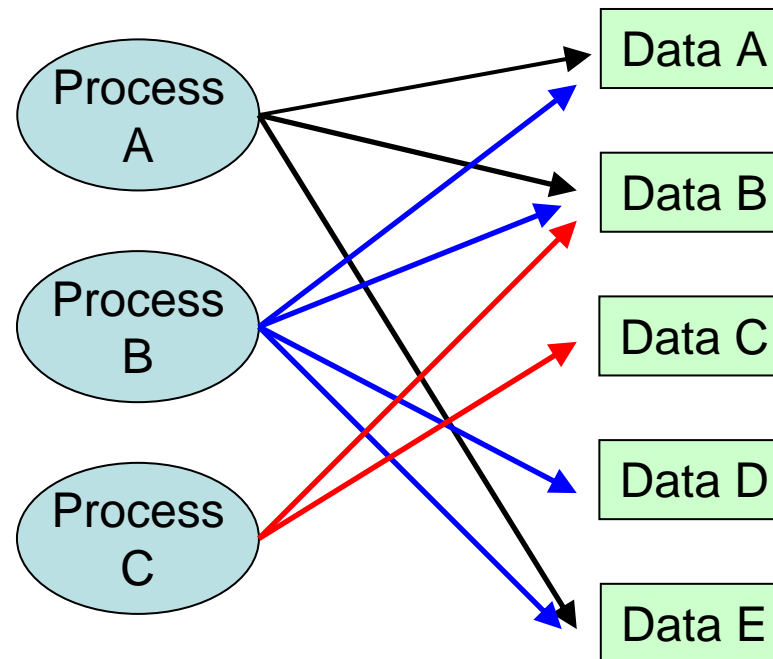


7.1.1 従来のシステム開発手法



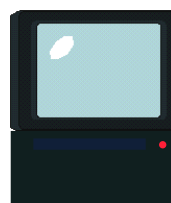
7.1.2 従来手法のデメリット



プロセスとデータが独立している
一方の変更に対し他方への影響調査が必要
影響範囲をすべて点検 必要に応じて修正

7.1.3 オブジェクト指向

ー従来手法のデメリット解決へー



TV

オブジェクトの概念

チャンネル, ブラウン管, 電源

電源ON, 映像表示, 電源OFF



属性()と操作()を一体で考える

7.1.4 オブジェクト指向の概念

	オブジェクトの属性や操作を他のオブジェクトから隠す(メリット:仕様変更の最小化)
抽象化(一般化)	より一般的なものに置き換える (例:14インチブラウン管テレビ テレビ)
インスタンスとクラス	インスタンス:具体的なもの (例:京大太郎、小泉純一郎) クラス:抽象化された要素・枠組み (例:人間)
	オブジェクトの属性・操作を引き継ぐ (例:乗用車・商用車・軍用車 車)
多態性(ポリモフィズム)	異なるオブジェクトに同一の操作を行う (例:読む 本・新聞・手紙・論文)

7.2.1 UMLとは？

UML ()

UMLは乱立する表記法を統一するために作られた
Unified(統一)という言葉の由来
UMLはオブジェクト指向によるシステム開発で用いられる
さまざまなモデルの表記法を標準化

年

OMG (Object Management Group:オブジェクト指向技術の
標準化団体)の標準へ

オブジェクト指向業界での表記法の

7.2.2 UML以前

それぞれ異なる表記法

Booch法

Grady Booch

Grady Booch



1) 参照

オブジェクト
名

オブジェクト
名

オブジェクト
名

OMT法



James
Rumbaugh

2) 参照

オブジェクト
属性
操作

オブジェクト
属性
操作

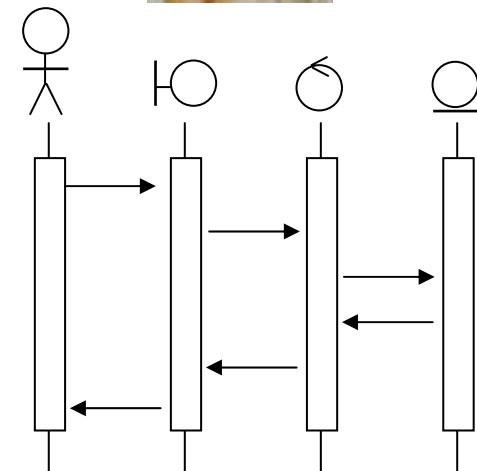
オブジェクト
属性
操作

OOSE法

Ivar Jacobson



3) 参照



1) <http://www-106.ibm.com/developerworks/library/i-booch/>

2) <http://www-306.ibm.com/software/rational/bios/rumbaugh.html>

3) <http://www.jaczone.com/postcards/>

7.2.3 UMLの歴史

1994	BoochとRumbaughがモデリング技法統一へ
1995	Jacobsonが統一作業に参加
1996	UML0.9
1997(Jan)	UML1.0
1997(Sep)	UML1.1
1997(Nov)	UML1.1がOMG標準に
1998	UML1.2
1999	UML1.3
2001	UML1.4
2003	UML1.5
2003	UML2.0ドラフト公開

OMG : Object Management Group (オブジェクト指向技術標準化団体)

7.2.4 UMLのダイアグラム

UMLの主なダイアグラム

ユースケース図	システムの機能とそのユーザを表現
クラス図	システムの静的な構造を表現
オブジェクト図	システムのある時点における静的な構造を表現
相互作用図 (シーケンス図・コラボレーション図)	オブジェクト間の相互作用を表現
ステートチャート図	オブジェクトの状態遷移を表現
アクティビティ図	処理や業務の流れを表現
コンポーネント図	コンポーネント間の依存関係を表現
配置図	システムの物理的な構成を表現
パッケージ図	パッケージ間の依存関係を表現

7.2.5 UML2.0

UML1.5から大幅に変更

2004年3月:ドラフト最終調整段階

2004年6月12日:仕様最終確定

2004年7月:公式仕様

UML2.0のダイアグラム

静的ダイアグラム

- ・クラス図
- ・オブジェクト図
- ・コンポーネント図
- ・配置図
- ・パッケージ図
- ・コンポジット構造図 (新規)

動的ダイアグラム

- ・ユースケース図
- ・ステートチャート図
- ・アクティビティ図
- ・シーケンス図
- ・コラボレーション図
- ・コミュニケーション図
(従来のコラボレーション図) (新規)
- ・相互作用概念図 (新規)
- ・タイミング図 (新規)

7.2.6 UMLの利点

UML

オブジェクト指向を使ったシステム開発
分析から設計、実装まで終始一貫して利用可能

(メリット)

分析から実装までをすべて「
単位で表現

を統一

分析から実装までが で統一

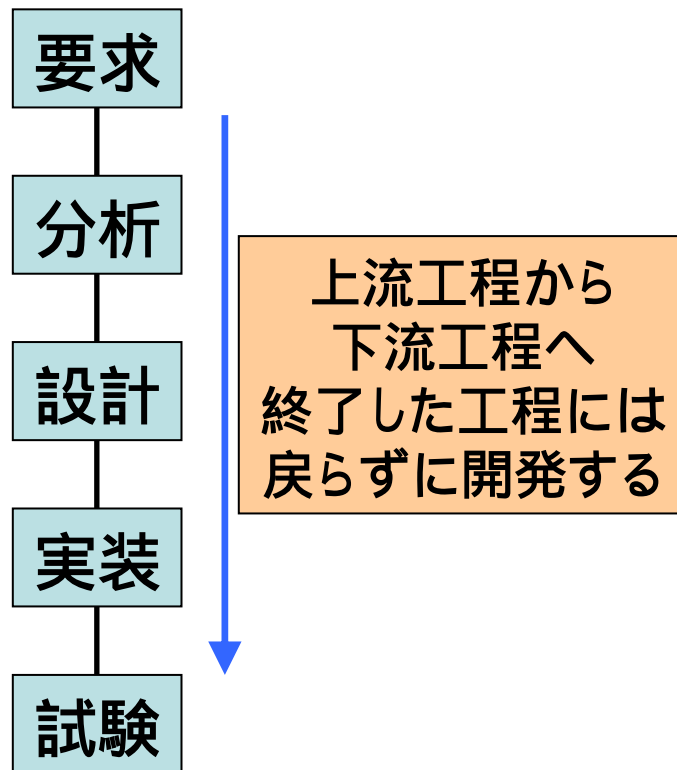
分析と設計のどの部分に対応するのか・設計と
実装のどの部分に対応するのか

不具合が出た際

分析や設計の を上げていくことが可能

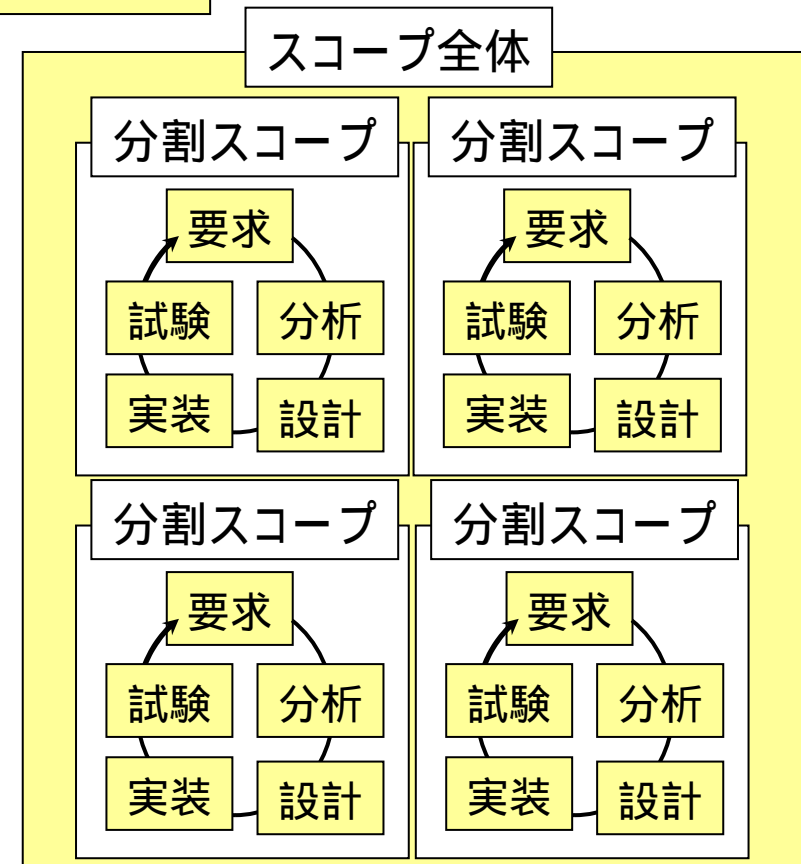
7.3.1 開発プロセス

Water Fall型



- ・スケジュール管理困難
- ・仕様変更困難(コスト大)

反復型



- ・仕様変更容易
- ・責任範囲(オブジェクト)明確

7.3.2 モデリング

モデル: ある対象を分析して整理し、表現したもの
モデリング: モデルを作成する作業

Step1: 要求モデリング

ユーザの要求把握 (システム化の対象範囲を明確に)

Step2: 分析モデリング

システム化の対象整理 (システムの構造を明確に)

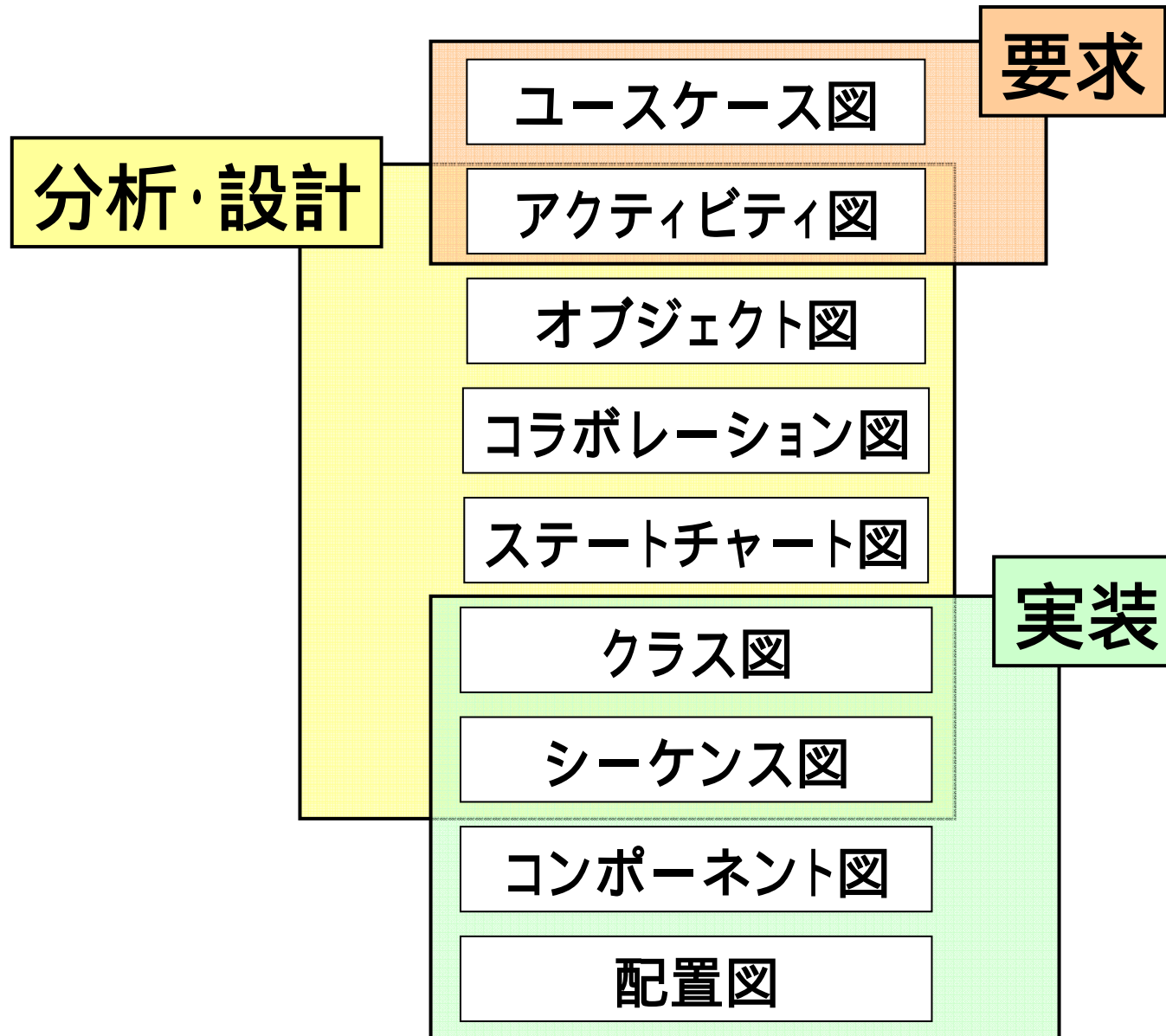
Step3: 設計モデリング

システム化の実現方法定義 (システム内部の仕様設計)

Step4: 実装モデリング

システムの構成要素定義 (構成・配置・動作を記述)

7.3.3 設計工程とUML



7.4.1 ユースケース図(1)

ユースケース

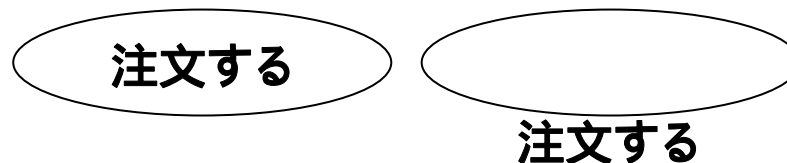
ユーザなどシステム外部から見たシステムの振る舞いを表す。システムの振る舞いとはシステムがどのように動作し、反応するかということ

アクター

システムと相互作用する外部の実体を抽象化したもの

表記法

楕円で表し、内側または下にユースケースを記述例)

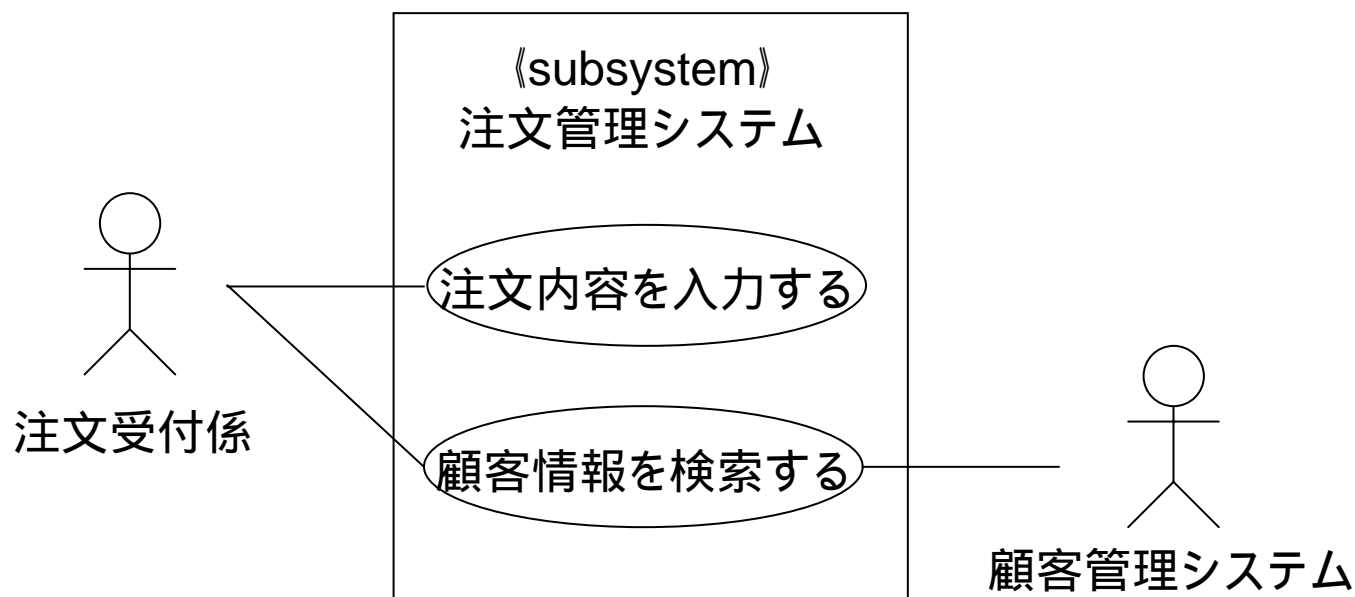


7.4.1 ユースケース図(2)

ユースケース図

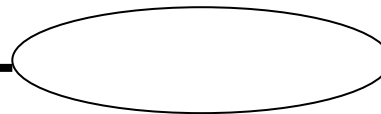
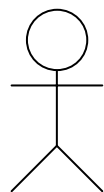
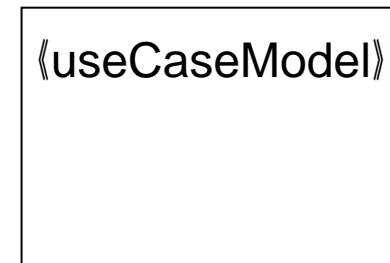
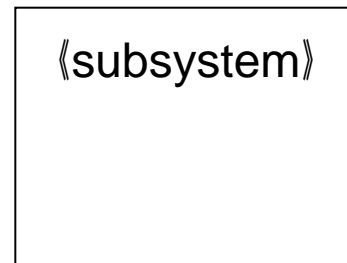
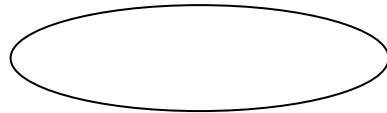
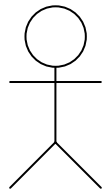
システムが提供する機能とそれに関連する外部要素
(ユーザなど)を表す

例)「注文管理システム」:「注文内容入力」と「顧客情報検索」を提供
ユーザは「注文受付係」/「顧客管理システム」を利用



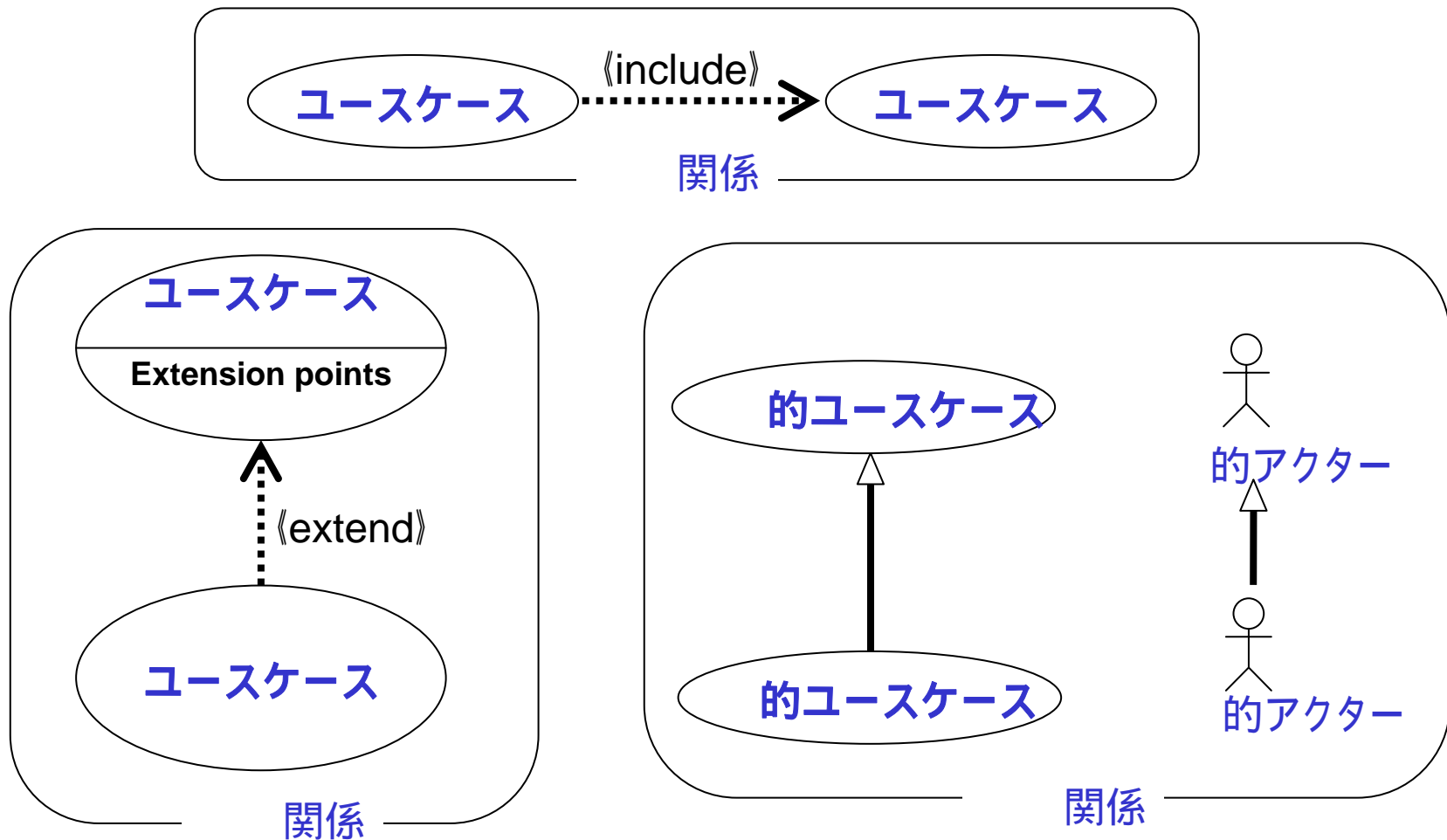
7.4.1 ユースケース図(3)

ユースケース図の要素(1)



7.4.1 ユースケース図(4)

ユースケース図の要素(2)



7.4.1 ユースケース図(5)

ユースケース図を使ったモデリング(1)

「世界中の酒」の通信販売を行っている兄弟社は、
昨今の焼酎ブームを当て込み個人向け販売情報システム
を導入することになった。
販売業務のみをシステム化の範囲とする。

7.4.1 ユースケース図(6)

シナリオ

受注担当者が受注すると本システムでは、在庫管理システムや顧客管理システムを使って、在庫や顧客情報を確認する。

営業担当者の場合は、さらに見積もりを作成したり、商品カタログを参照する。商品カタログでは、通常商品と季節商品向けのカタログが用紙されている。見積もり作成では、顧客管理システムを使って、顧客情報を確認し、場合によっては、割引を行う。

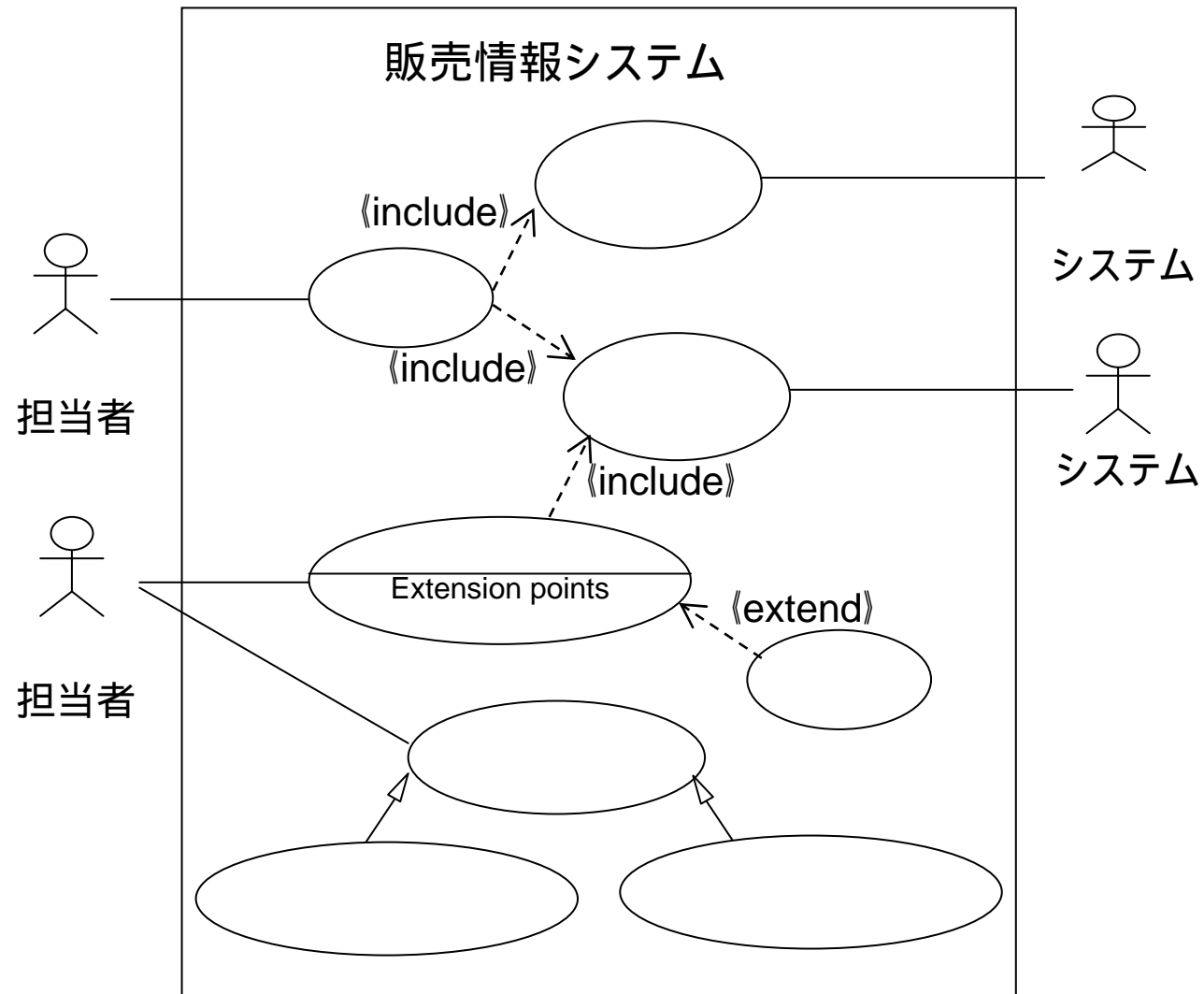
仕入れ担当者は、仕入れするにあたって、在庫管理システム、売上げ情報システムや価格情報システムを使って、在庫、売上げ情報や価格情報を確認する。

7.4.1 ユースケース図(6)

要求

ユースケース図を使ったモデリング(2)

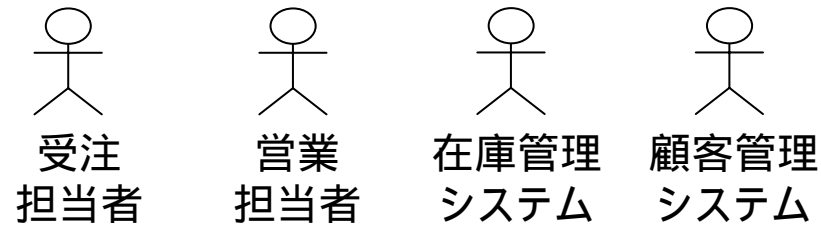
最終形



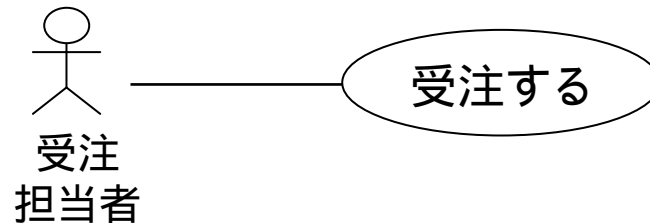
ユースケース図を使ったモデリング(3)

手順1

アクターの候補を選出する
(システム利用者と外部システム)



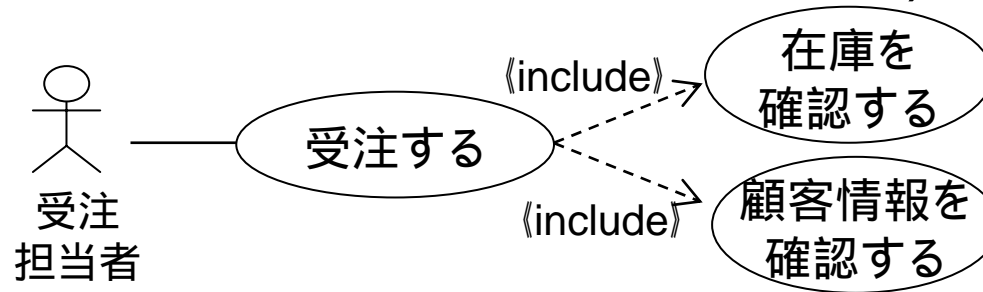
アクターごとのユースケースを考える
(システムの機能を検討)



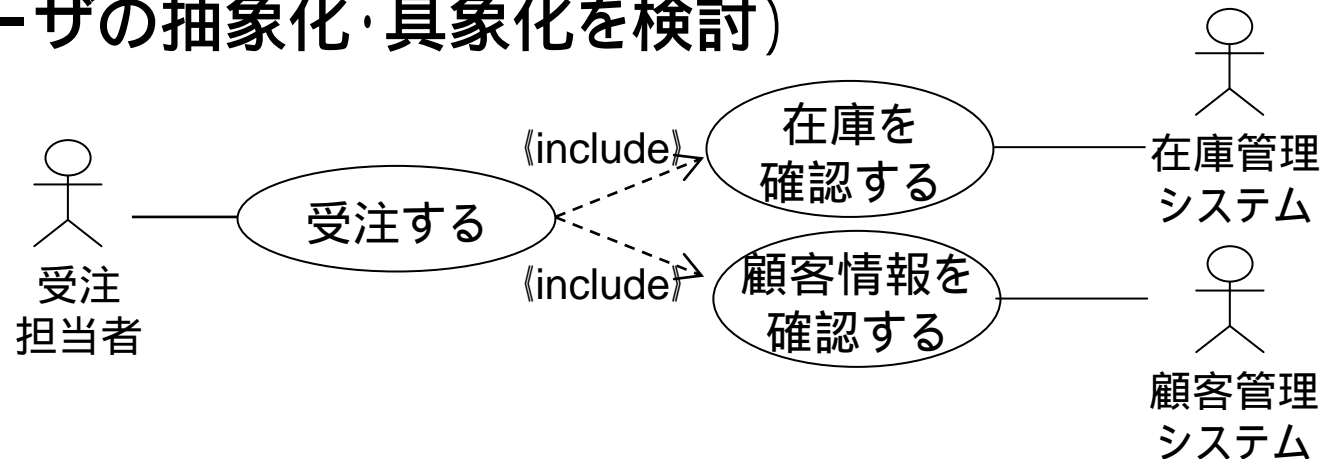
ユースケース図を使ったモデリング(4)

手順2

包含されるユースケースを抽出
(機能に関連する外部システムを検討)



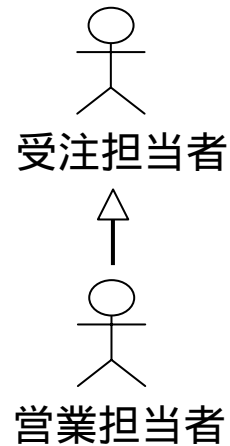
外部システムをアクターとして抽出
(ユーザの抽象化・具象化を検討)



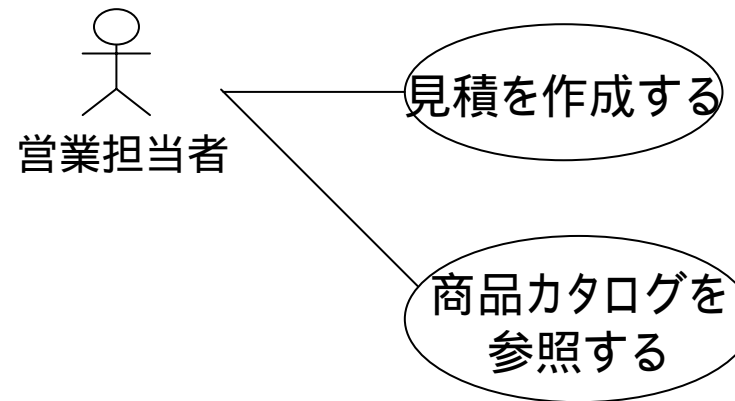
ユースケース図を使ったモデリング(5)

手順3

営業担当者と受注担当者の汎化関係を定義
(システムの機能を検討)



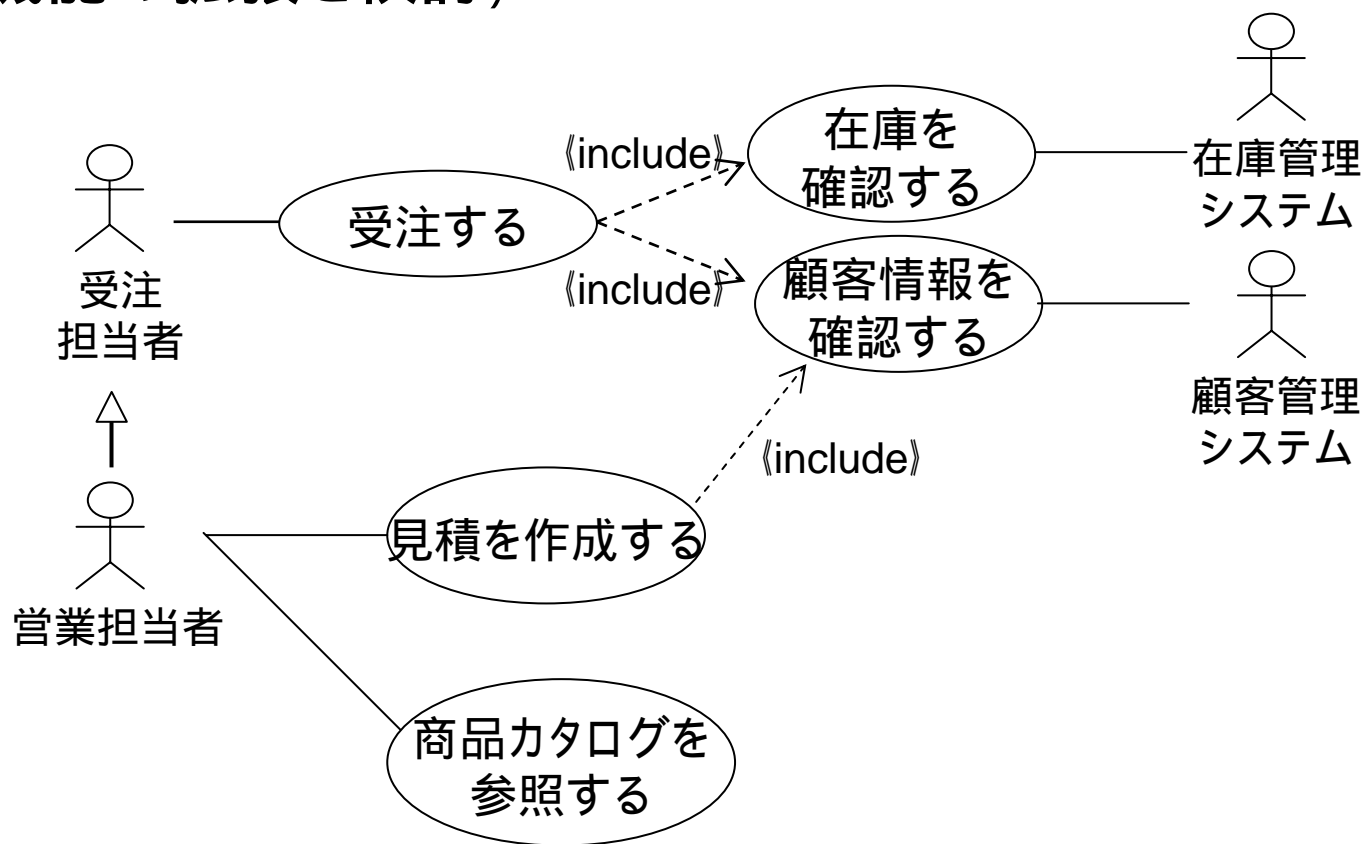
営業担当者のユースケースを抽出
(機能の包含関係を検討)



ユースケース図を使ったモデリング(6)

手順4

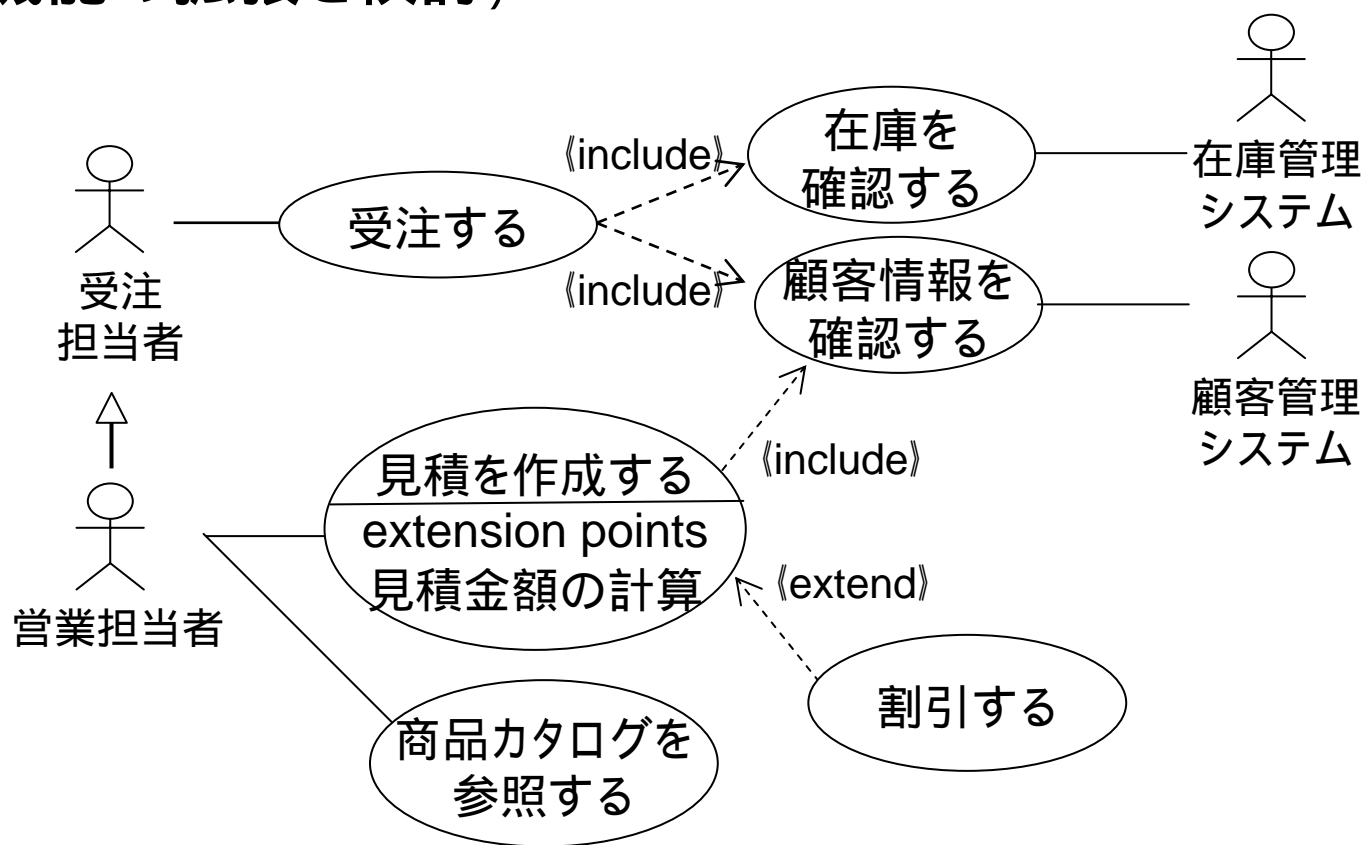
「見積を作成する」から「顧客情報を確認する」へ包含関係
(機能の拡張を検討)



ユースケース図を使ったモデリング(7)

手順5

「見積を作成する」を拡張する
(機能の拡張を検討)



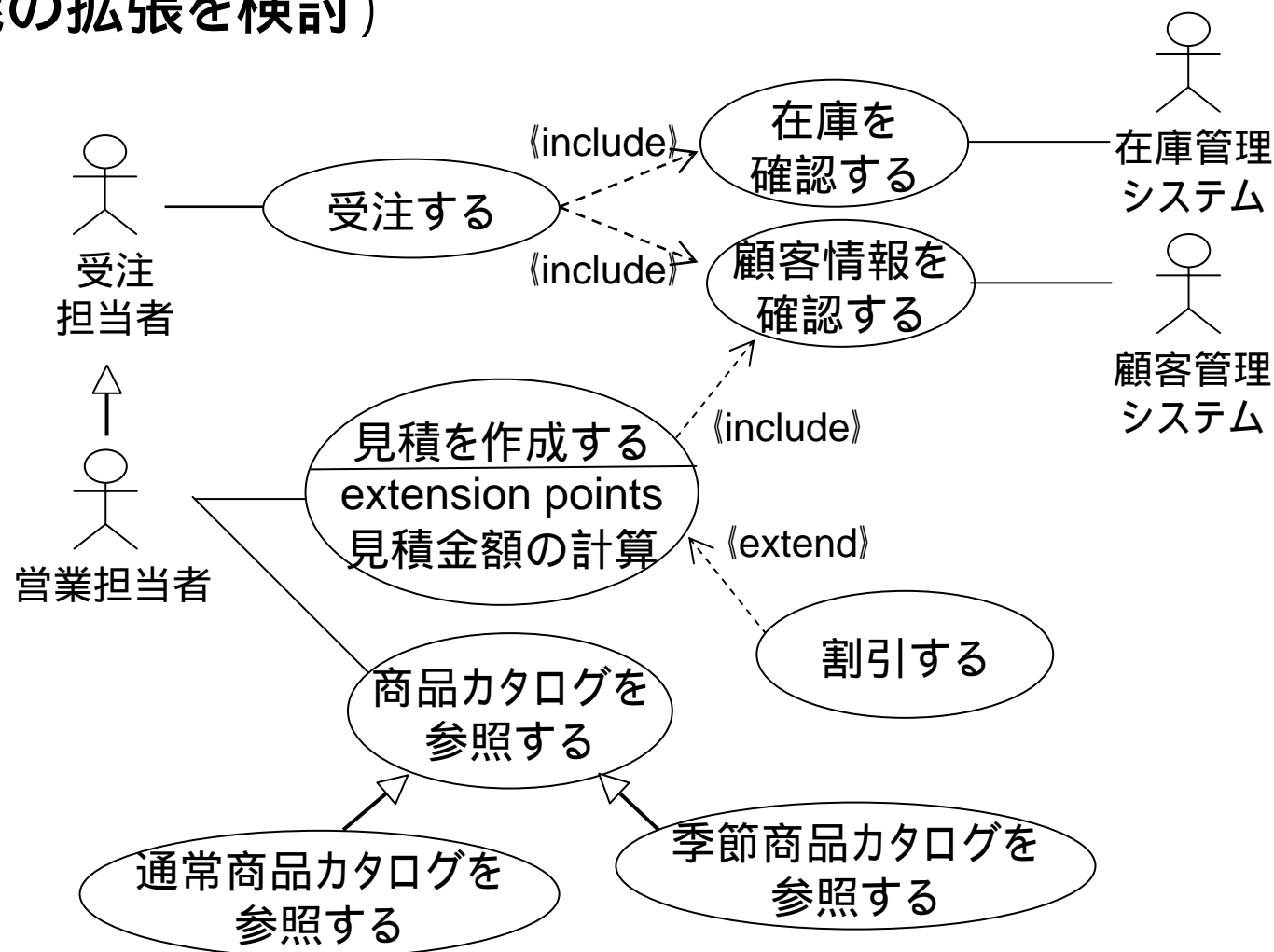
7.4.1 ユースケース図(12)

要求

ユースケース図を使ったモデリング(8)

手順6

「商品カタログを参照する」の特化したユースケースを抽出
(機能の拡張を検討)



7.4.2 アクティビティ図(1)

アクティビティ図

処理の手順を表す

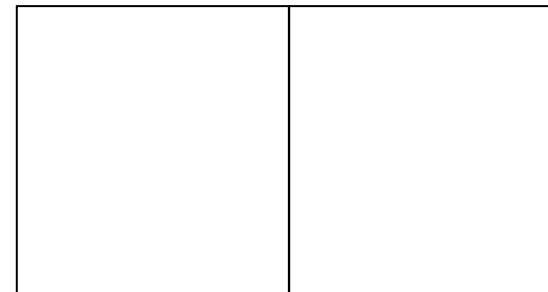
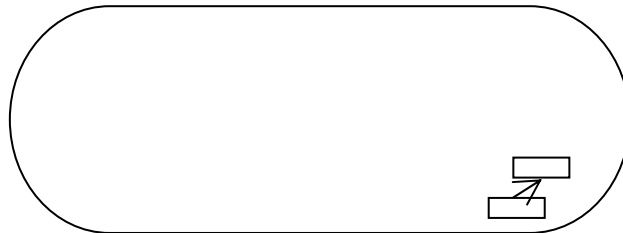
アクティビティ図の要素(1)



状態

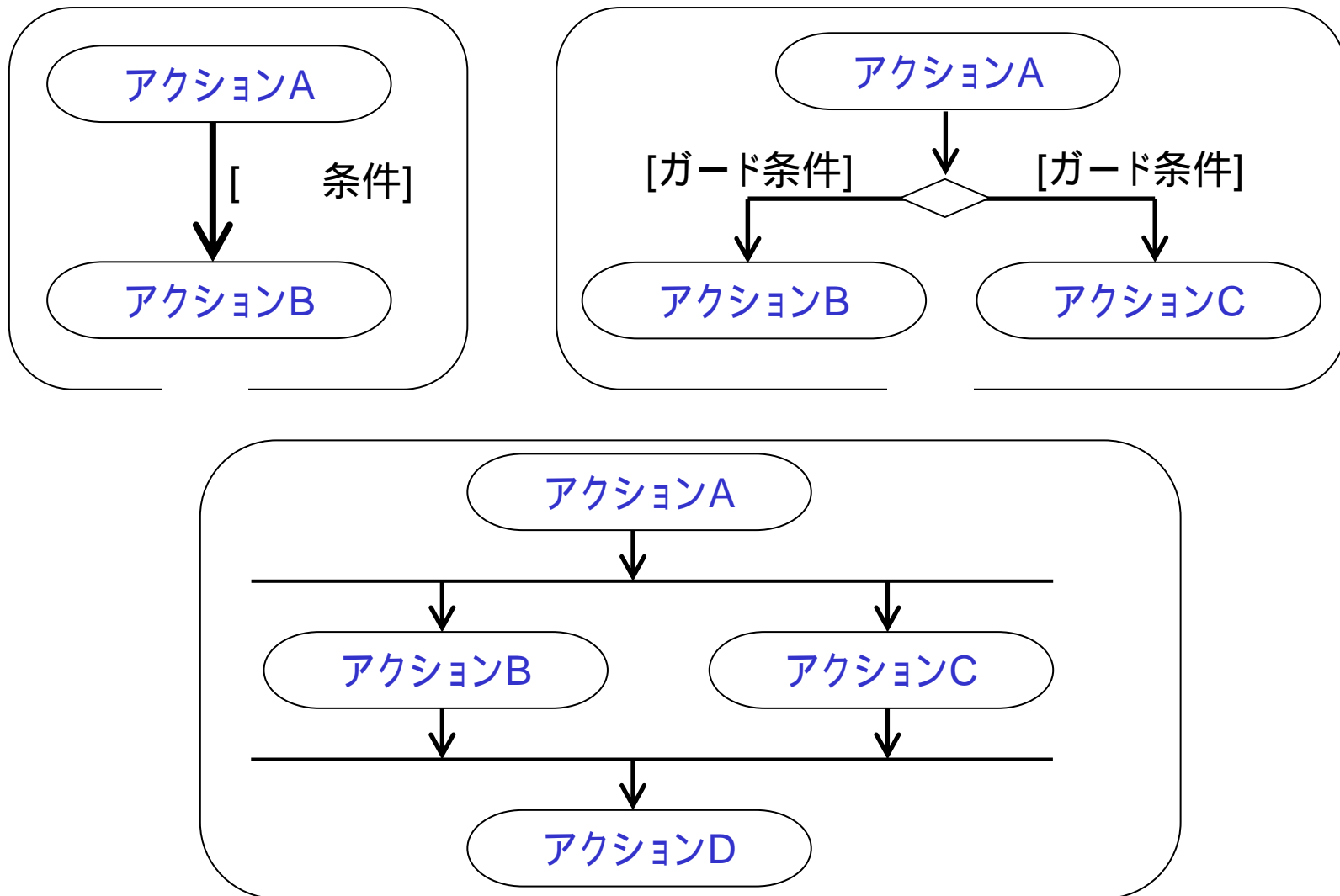


状態



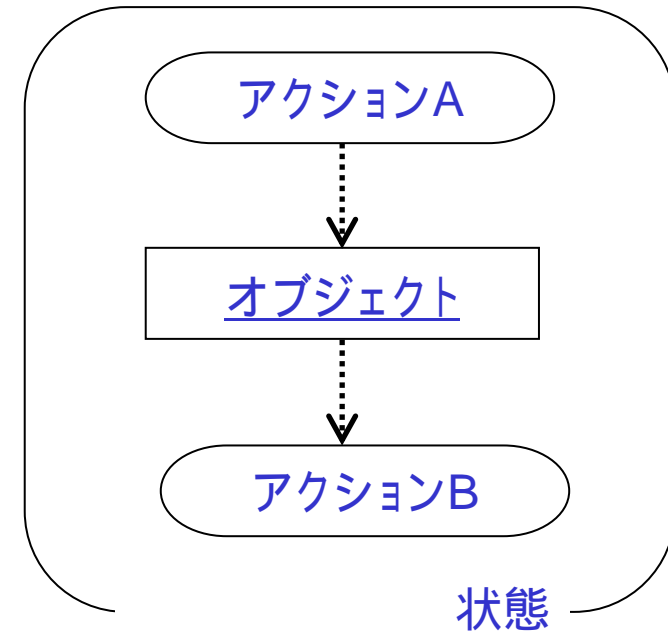
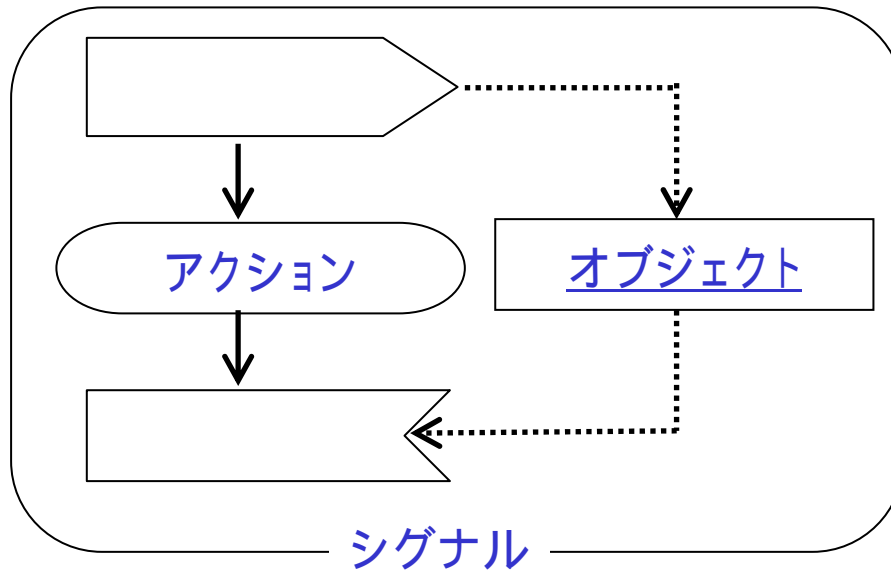
7.4.2 アクティビティ図(2) 要求

アクティビティ図の要素(2)



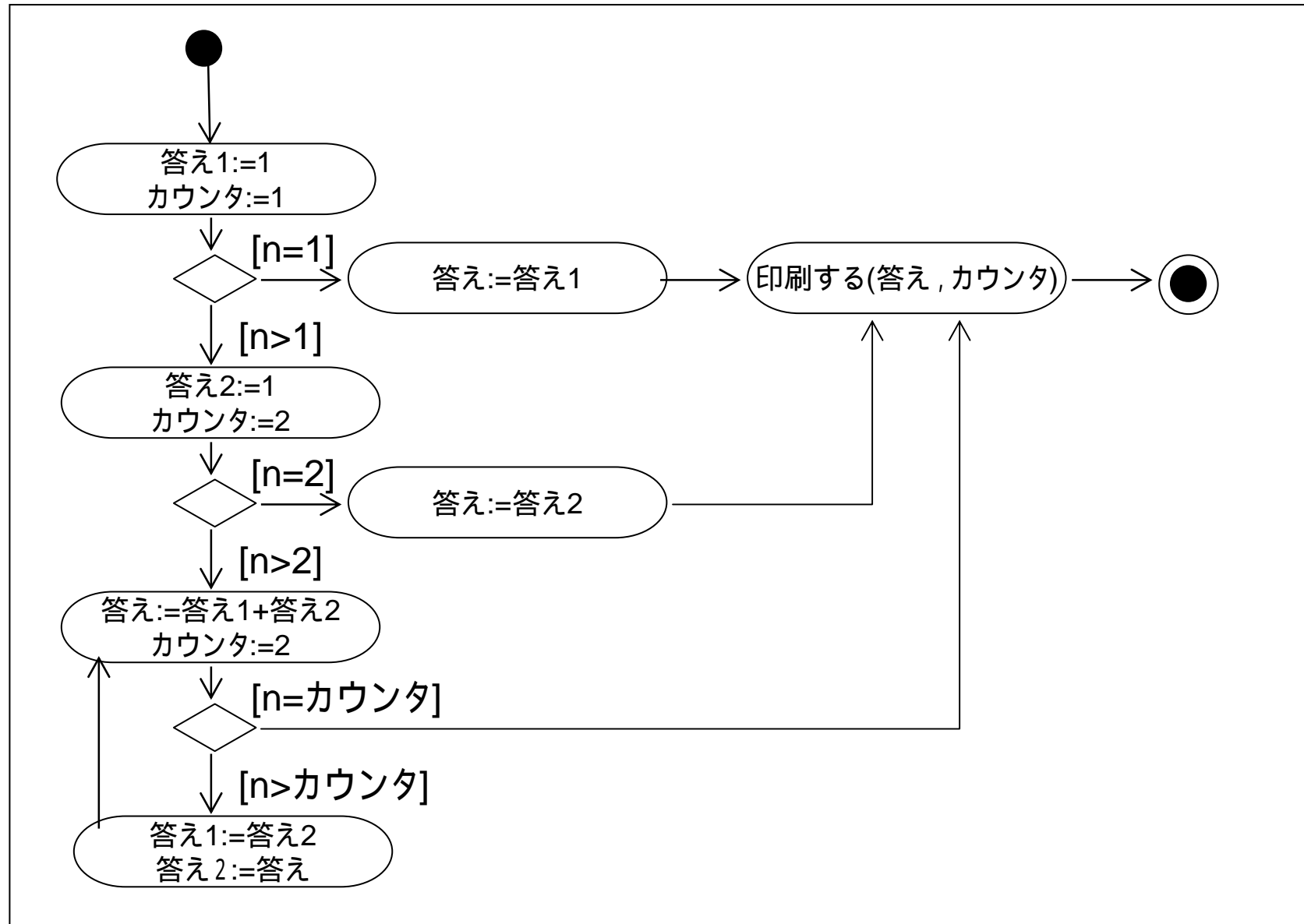
7.4.2 アクティビティ図(3)

アクティビティ図の要素(3)



アクティビティ図を使ったフィボナッチ数列計算の表現

要求



7.4.2 アクティビティ図(4)

アクティビティ図を使ったモデリング例

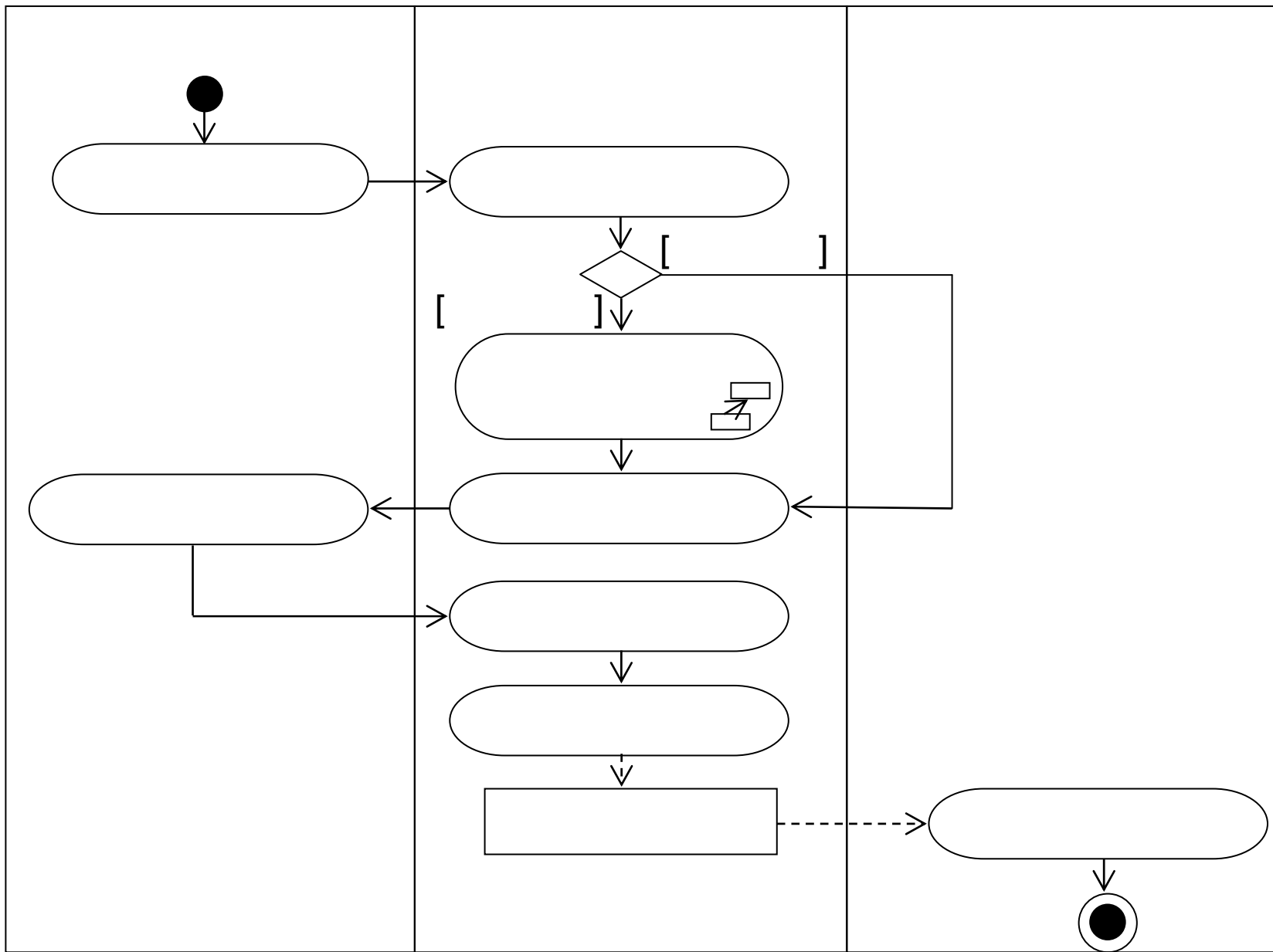
「世界中の酒」の通信販売を行っている兄弟社は、大口販売管理システムを導入することになった。
販売管理システムのうち、受注情報登録における業務の流れを表現する

7.4.2 アクティビティ図(5)

アクティビティ図を使ったモデリング例

最終形

要求



アクティビティ図を使ったモデリング例

手順1

業務手順を整理する

- ・「誰が」、「何を」しているのかに注目(業務担当で考える 汎化)
 - 顧客名を伝える(顧客)
 - 顧客情報を確認する(受注係)
 - 注文内容を尋ねる(受注係)
 - 注文内容を伝える(顧客)
 - 注文内容を登録する(受付係)
 - 出荷を依頼する(受注係)
 - 商品出荷を行う(倉庫係)

7.4.2 アクティビティ図(7)

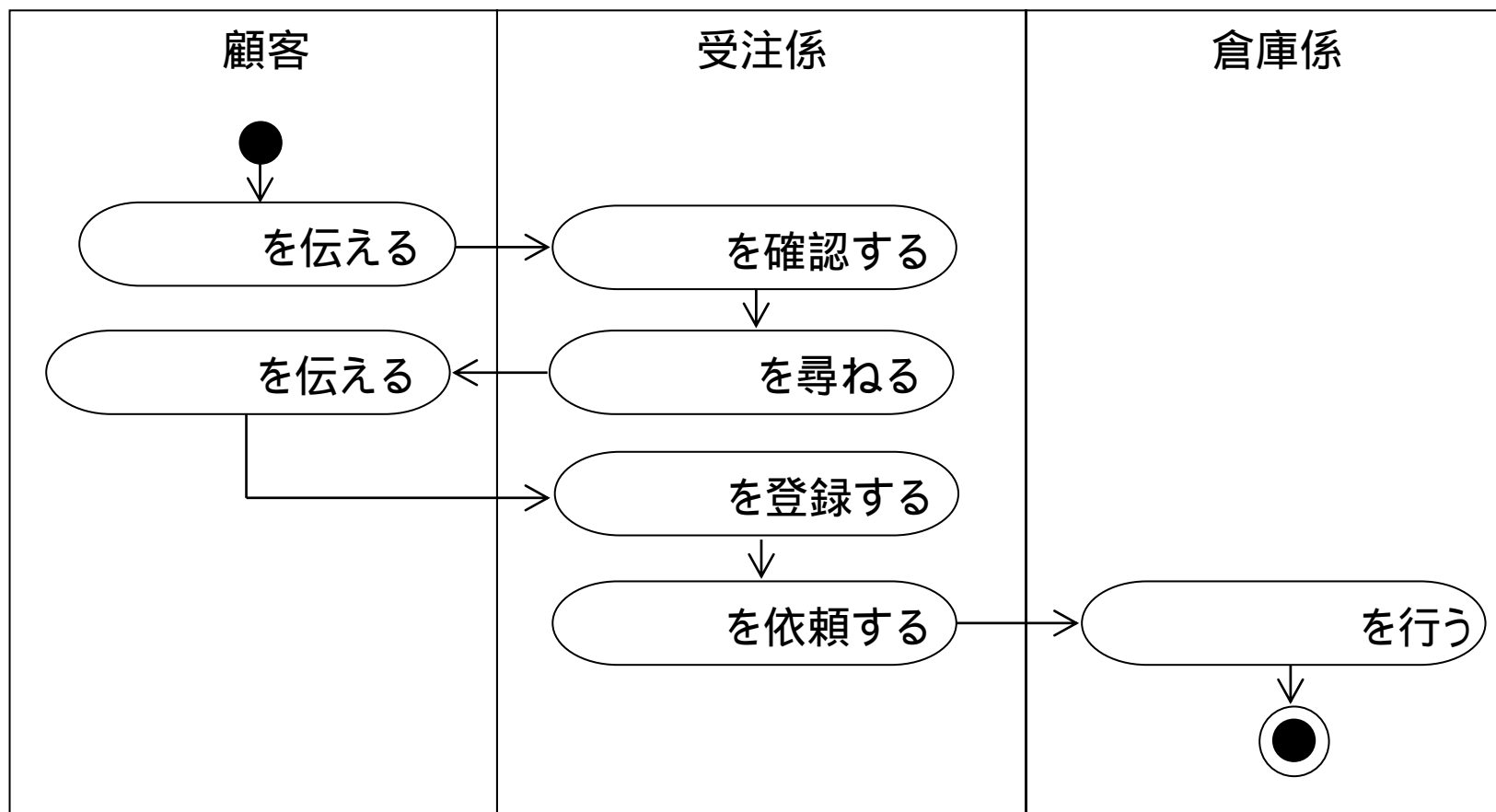
要求

アクティビティ図を使ったモデリング例

手順2

業務手順を表現する

- ・スイムレーン(誰が)とアクション状態(何を)で表現する



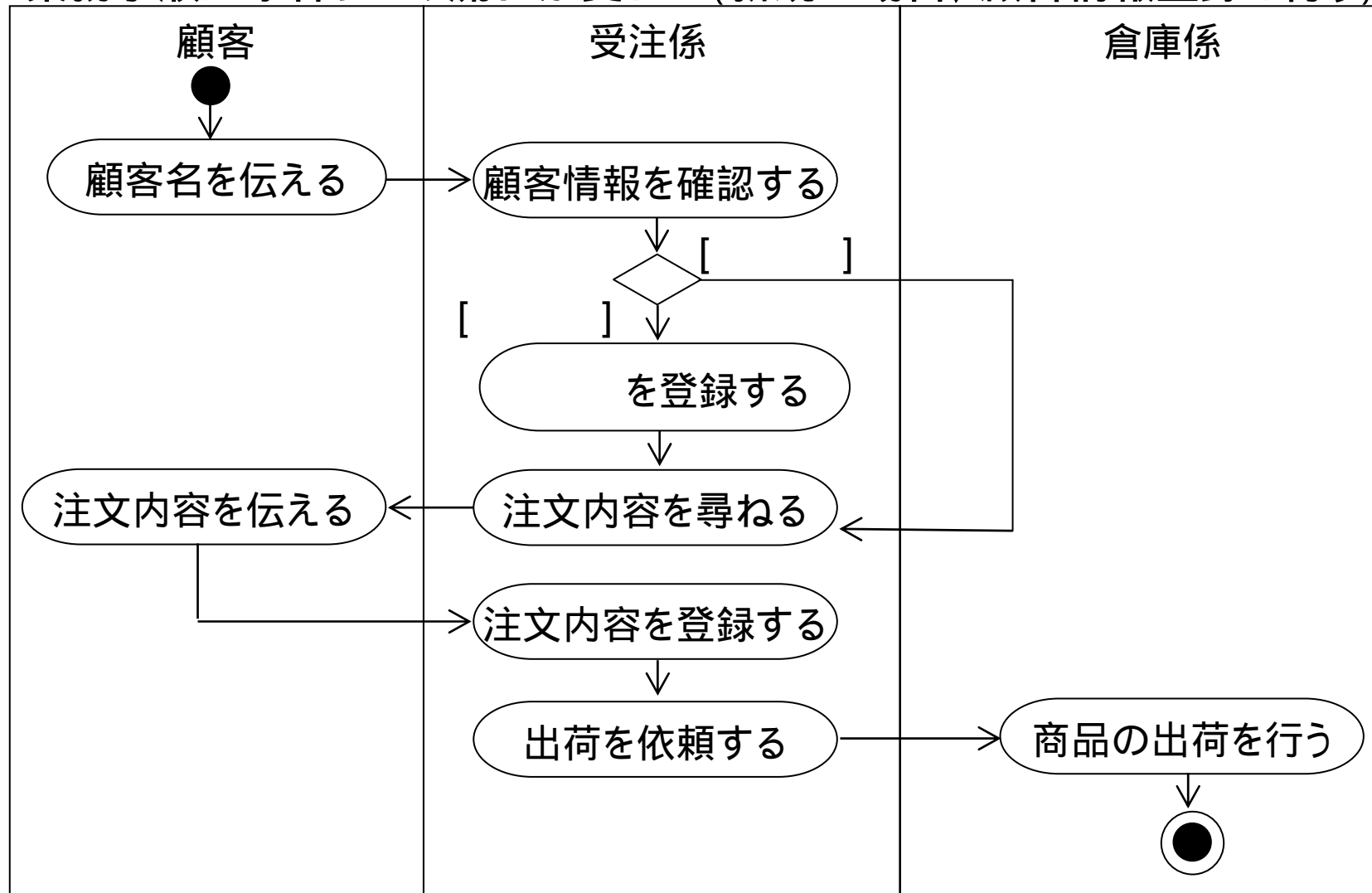
7.4.2 アクティビティ図(8)

アクティビティ図を使ったモデリング例

手順3

要求

業務手順の分岐を表現するー ガード条件を表現する
・業務手順は条件により流れが変わる(新規の場合、顧客情報登録を行う)



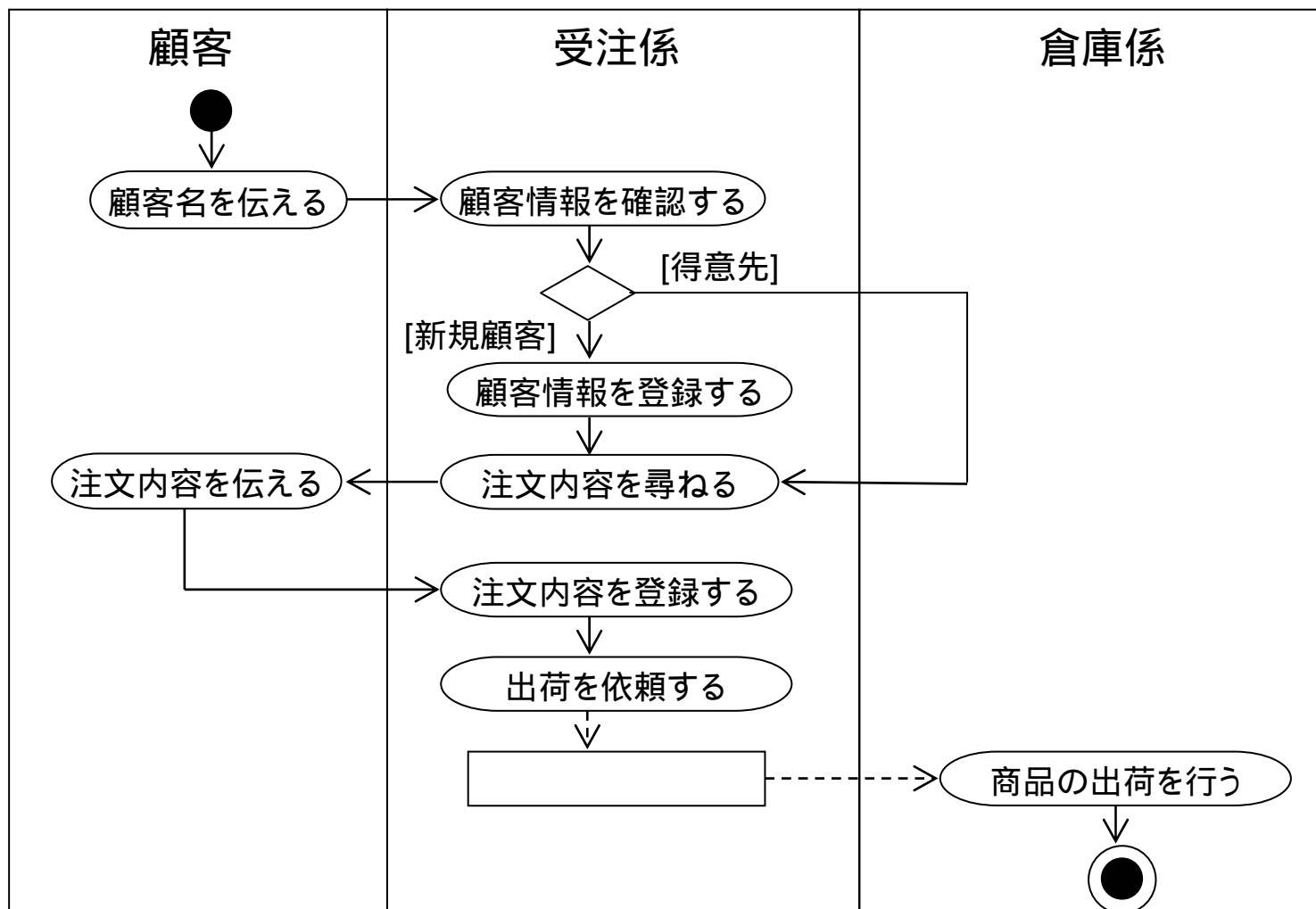
7.4.2 アクティビティ図(9)

アクティビティ図を使ったモデリング例

手順4

要求

- データの受け渡しを表現するー オブジェクトフロー状態を表現
- 受注係から倉庫係に「出荷指示書」というデータが流れる



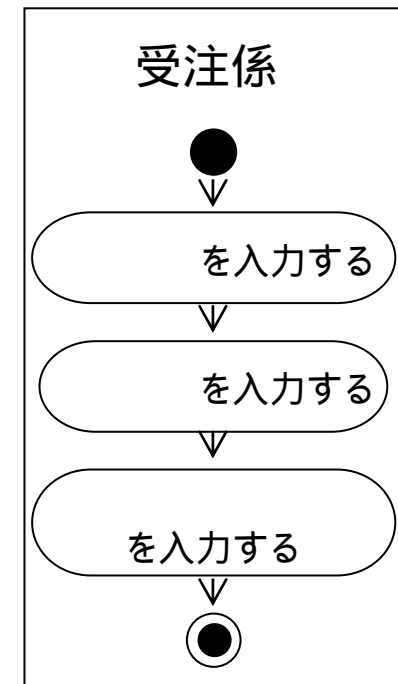
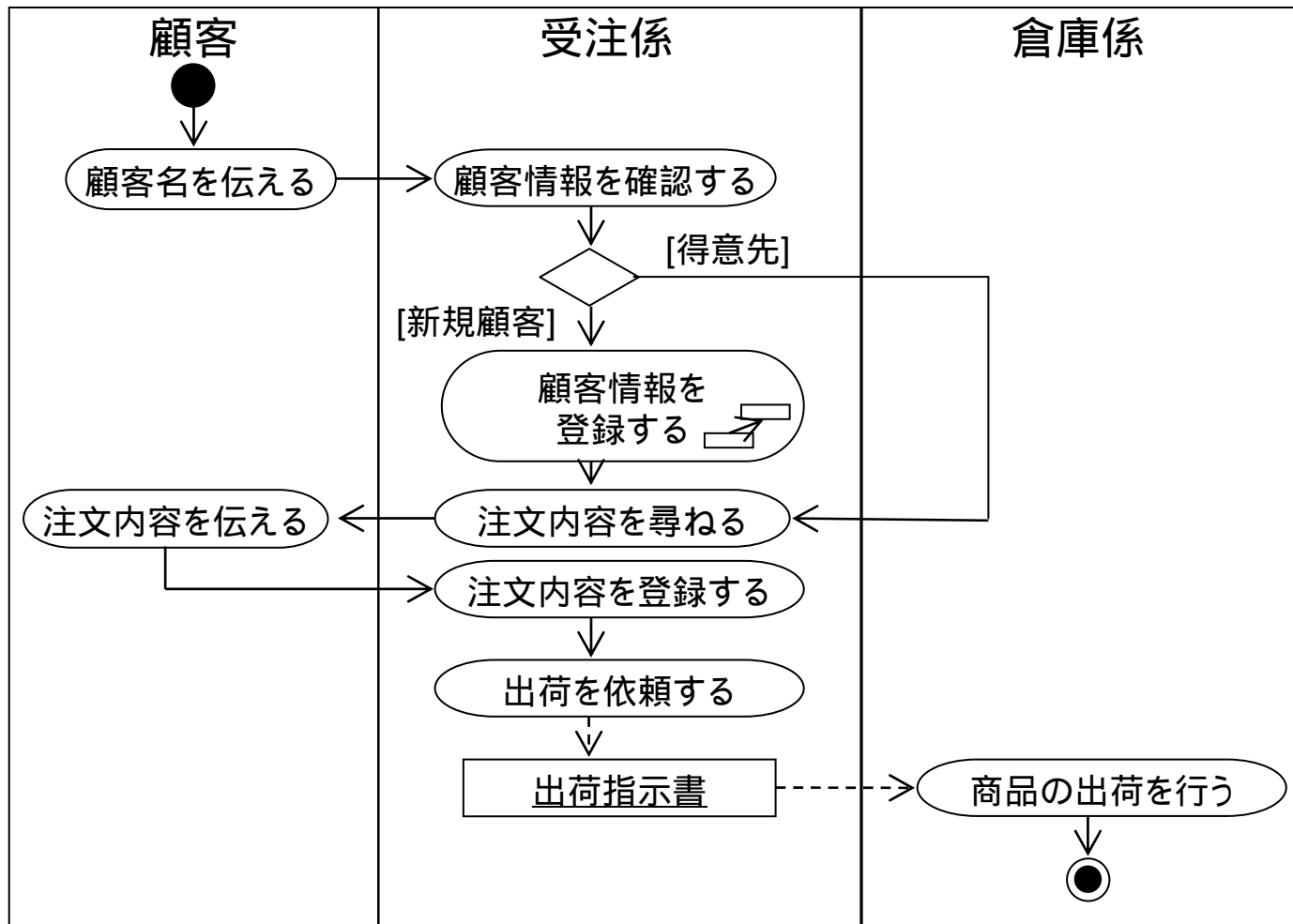
7.4.2 アクティビティ図(10)

アクティビティ図を使ったモデリング例 手順5

要求

業務手順の詳細表現するー サブアクティビティ状態を表現

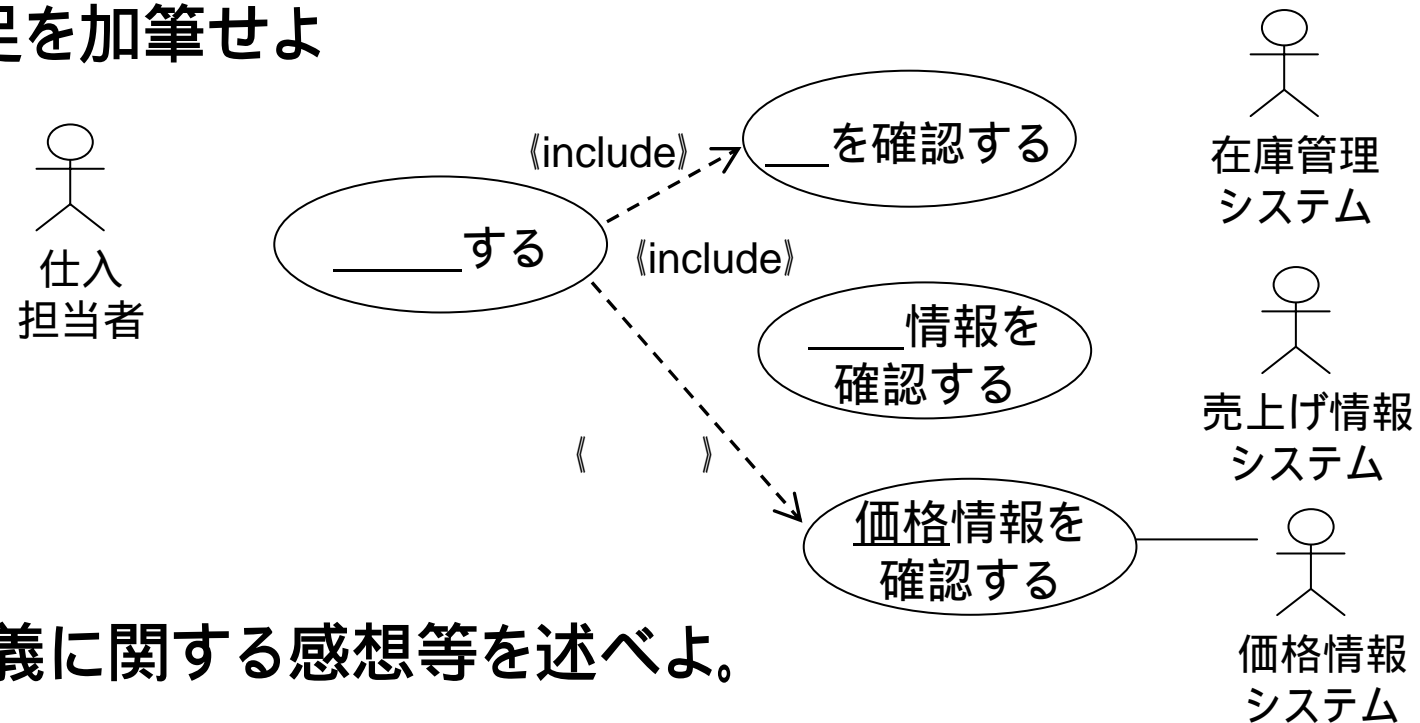
- ・「顧客情報登録」の中には「顧客名入力」、「顧客住所登録」などが含まれるサブアクティビティ状態を用いる。



小テスト(氏名:)

学籍番号:

(1) 兄弟会社の仕入担当者のユースケース図について
不足を加筆せよ



(2) 講義に関する感想等を述べよ。