

システム開発プロジェクト現場における モデリングの実際

(株)日立製作所ソフトウェア事業部 チーフアーキテクト 桐越 信一

Contents

1. SOAとコンポーネントの事例
2. コンポーネントベースモデリングの背景
3. コンポーネントベースモデリングの事例
4. プロジェクト推進のポイント

Contents

- 1. SOAとコンポーネントの事例**
2. コンポーネントベースモデリングの背景
3. コンポーネントベースモデリングの事例
4. プロジェクト推進のポイント

1-1 一般的に定義されているSOAとは

ポイント

- SOA 技術とは、「サービスの組合せ」でシステムを構築すること
 - ・ アーキテクチャとして（分散処理、業務の部品化）
 - ・ アプローチとして（業務プロセスの可視化、サービスの再利用）

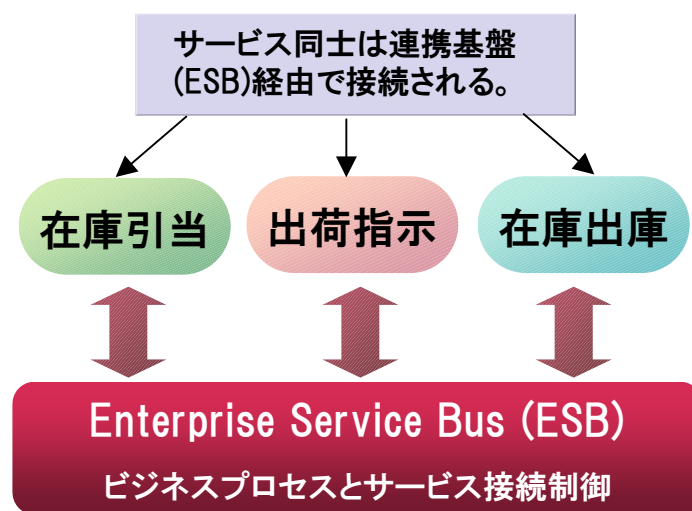
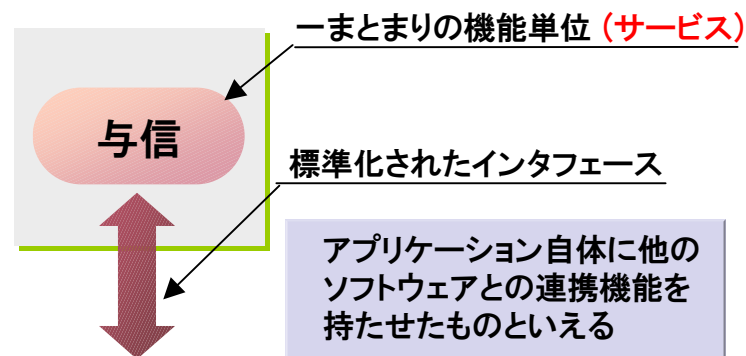
■サービスとは？

- ・ 外部から**標準化された手順**によって呼び出せる**一連のソフトウェア群(=コンポーネント)**
- ・ **単体で人間にとって意味のある単位の機能**を有するソフトウェア群

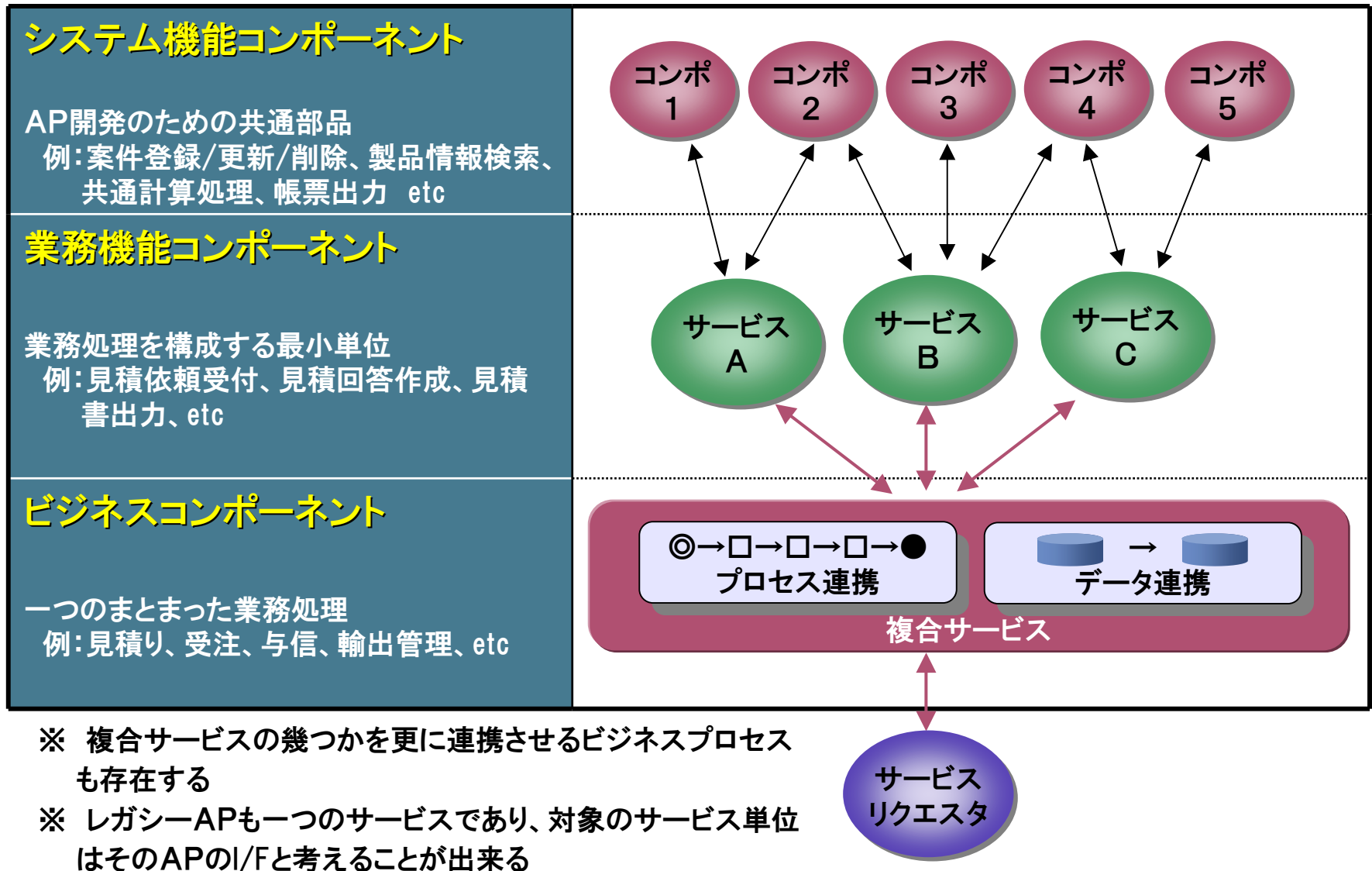
■組合せとは？

- ・ 業務に合わせて**必要なサービスを緩やかに結合**することで柔軟なシステムを構築
- ・ 業務をサービスの集合としてとらえることで、**プロセスの可視化**を実現
- ・ サービスの**部品化**を進めることで、**再利用**を促進

「サービス」として分解された一連の業務処理を組合せて、業務プロセス全体を構築する考え方がSOA

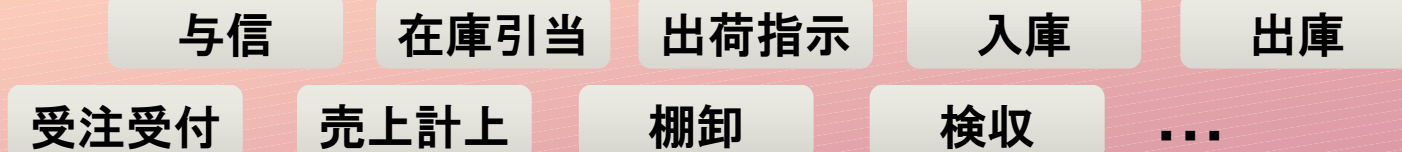


1-2 SOAにおけるサービスとコンポーネントの関係



1-3 業務機能とシステム機能コンポーネントの事例

業務機能Component



業務

採番管理

カレンダー
管理

共通計算
管理

帳票

Excel入出力

SVF生成

CSV入出力

変換

文字列

フォーマット/
コード

ファイル
(ZIP)

システム 管理

ログ出力

メッセージ
出力

データ
集配信

DB 制御

EBS連携

ストアド
プロシジャ管理

データ
アクセス

連結情報
アクセス

認証

認証

ロール管理

操作

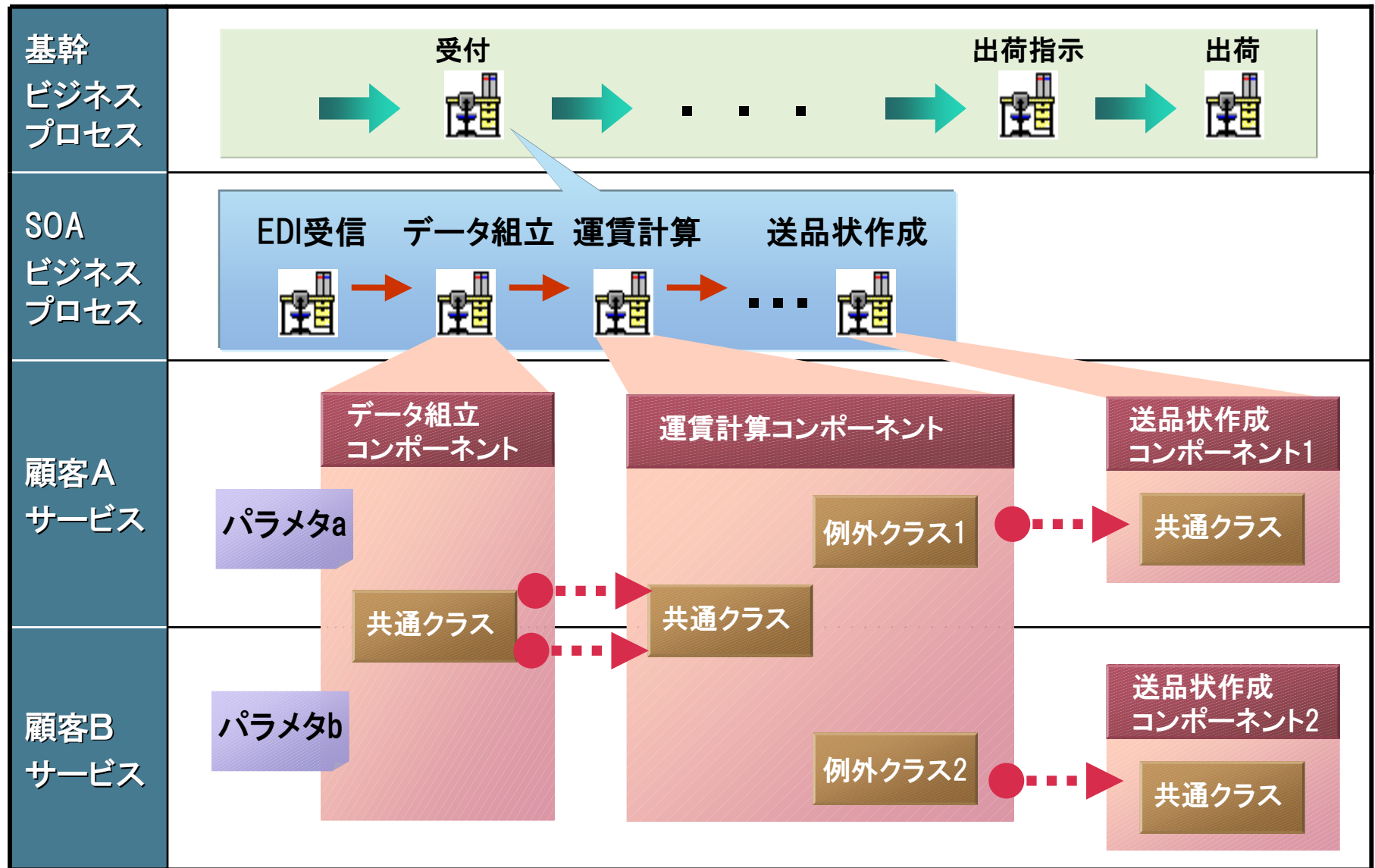
ファイル
アップロード/
ダウンロード

メール送信

...

システム機能Component

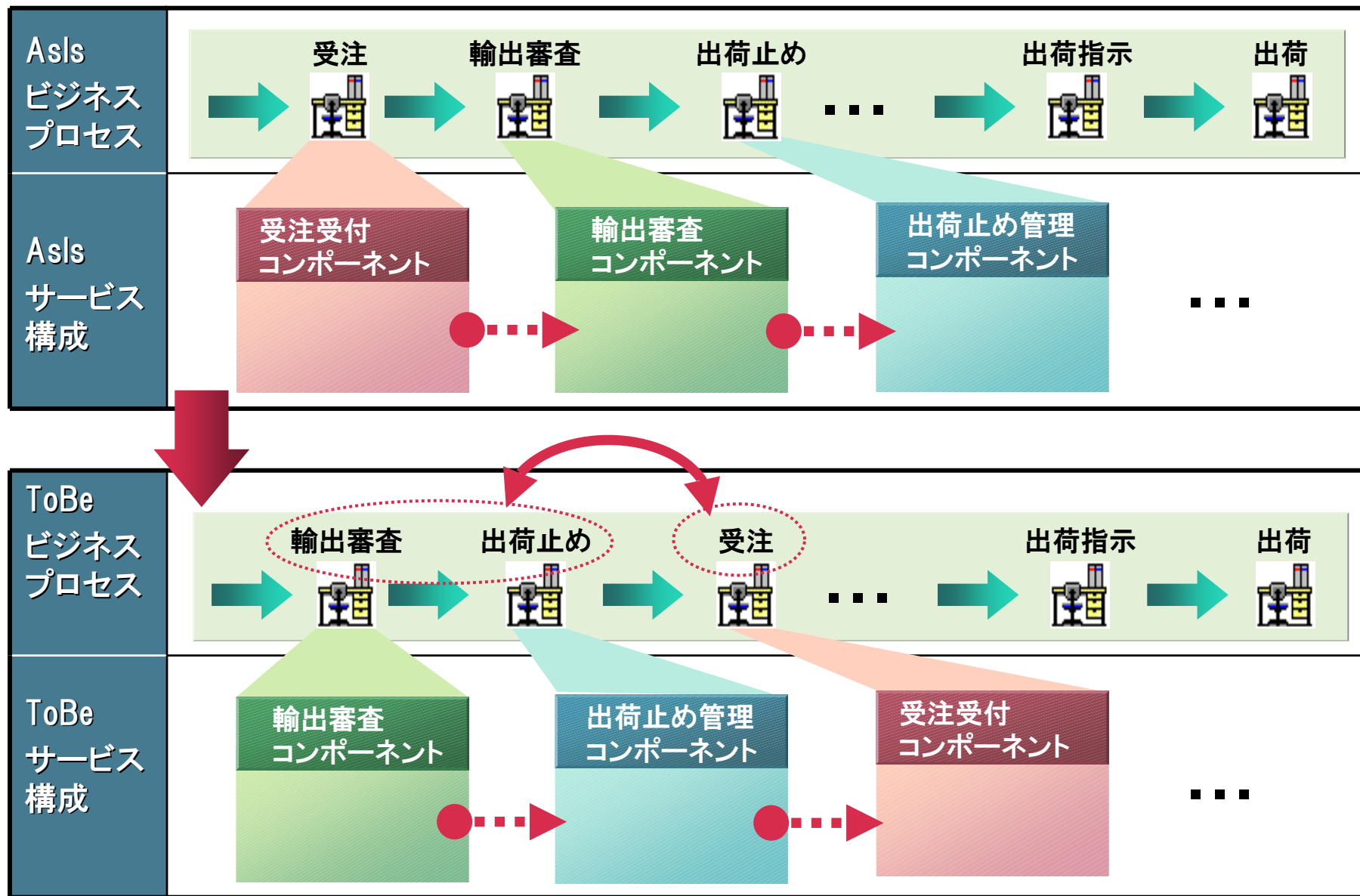
1-4 SOAとコンポーネント化の事例（サービス組立）



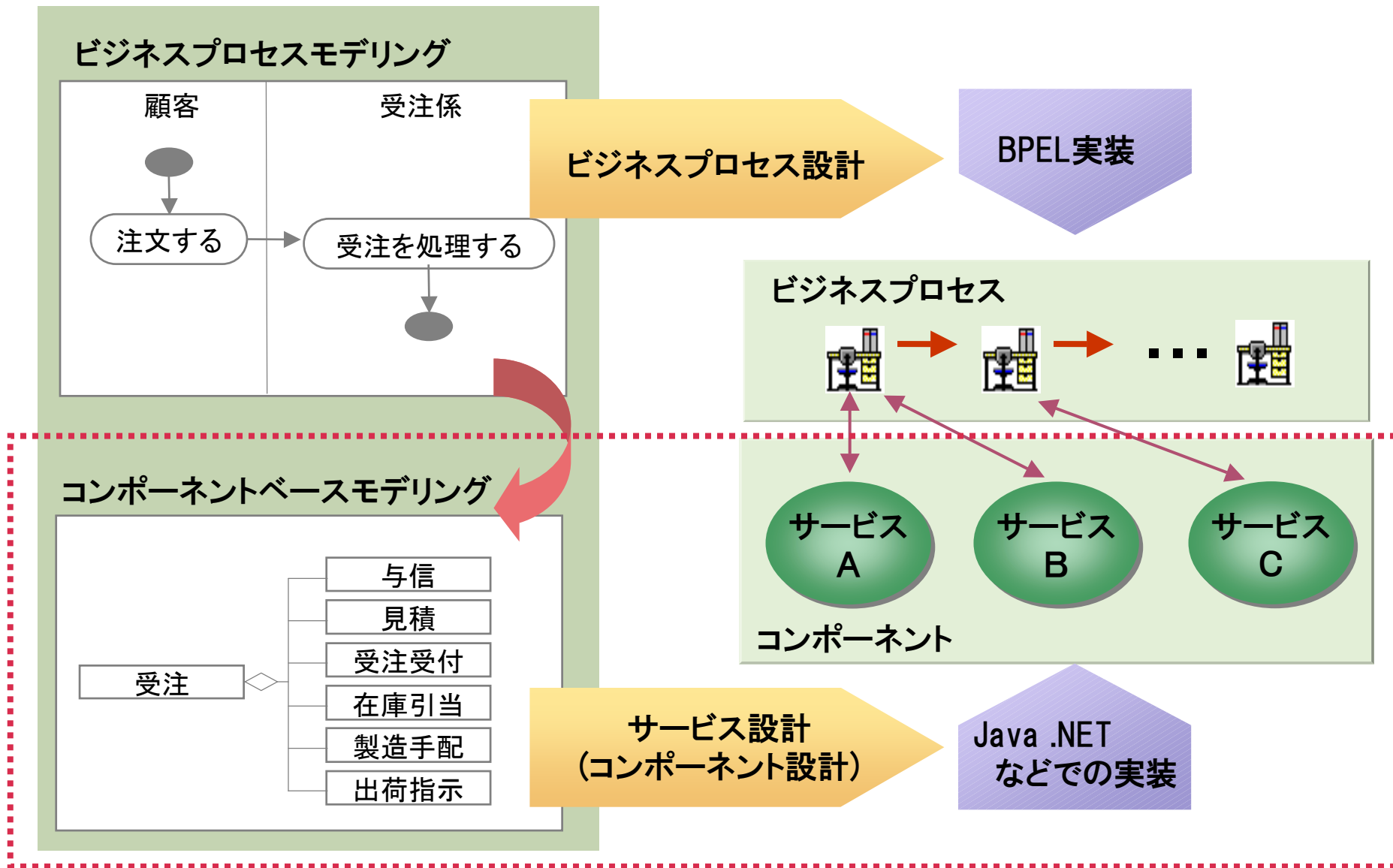
1-5 SOAとコンポーネント化（適合性事例）

業 務 処 理	SOA 適合性	コンポーネント化 のメリット	備 考
トランザクション系 共通処理	○	○	コンポーネント化を行なうことでパラメタ駆動による処理が可能な部分が多い。 さらに、顧客ごとに異なる処理を組み合わせる部分があるため、コンポーネント化を行い、サービスという単位で組み換えを行なうことで、これらの対応が容易になる。
トランザクション系 個別処理	×	×	個別部分はコンポーネント化しても、他からの流用が無い場合、効果が望めない。
照会系処理	△	○	コンポーネントのデータアクセス部分は全て共用可能。情報などを公開することを考えると、アクセス権に応じたサービスの組み換えが必要となり、これらに対応可能。
印刷系処理	○	○	コンポーネントの共用は全面的に可能。 サービスという側面で考えると、どういった印刷をするかを判断し、組み立てる構造にできるため、この部分でのメリットがある。
マスタメンテナンス	×	○	データアクセス部分は全て共用可能。 メンテナンス系は使用者が限られ、サービスとして組み替えるニーズは少ないはずであり、SOAとしての適合は対象外。

1-6 SOAとコンポーネント化の事例（サービス組替え）



1-7 SOAにおける設計アプローチ



Contents

1. SOAとコンポーネントの事例
- 2. コンポーネントベースモデリングの背景**
3. コンポーネントベースモデリングの事例
4. プロジェクト推進のポイント

2-1 参画プロジェクトでの反省

年	適用技術	対象システム
1999年～ 2000年	・Java ・UML(クラス図のみ)	他社(製造業)SCMシステム



- 反省点 ● 保守性: プログラム構造がバラバラ
● 生産性: 開発効率が従来と比べ、それほど上がらない

年	適用技術	対象システム
2001年	・Java ・UML(クラス図のみ) ・MVCフレームワーク ・コンポーネント適用	自社CRMシステム

- 保守性: プログラム構造の統一が可能
● 生産性:
設計工程 → 3割減
コーディング工程 → 1割減
テスト工程 → 2割減

- 反省点 ● もっと抜本的に低減できないのか？
● コンポーネント化はうまくいったのか？

2-2 プロジェクトでの適用コンポーネント検証

● MVCフレームワーク提供の適用コンポーネント

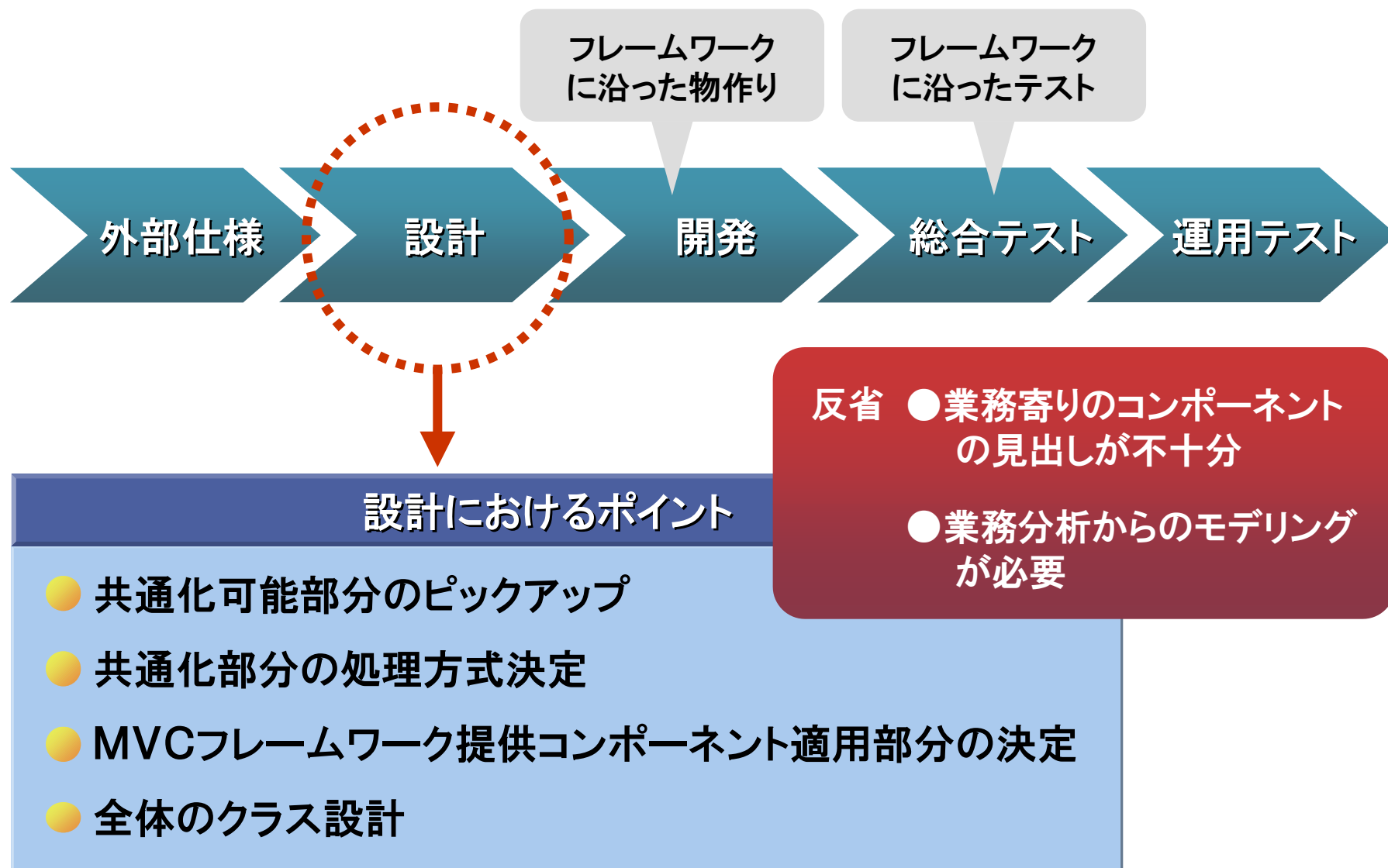
提供コンポーネント	内容
入力チェック	定義条件に基づき、ユーザの入力内容をチェック。 入力画面における文字サイズ、数字項目のチェックなどに利用
RDBアクセス	RDBをオブジェクトベースでアクセスする。
HTMLテンプレート	拡張HTMLで記述されたテンプレートファイルで動的画面生成

● 開発コンポーネント(代表例)

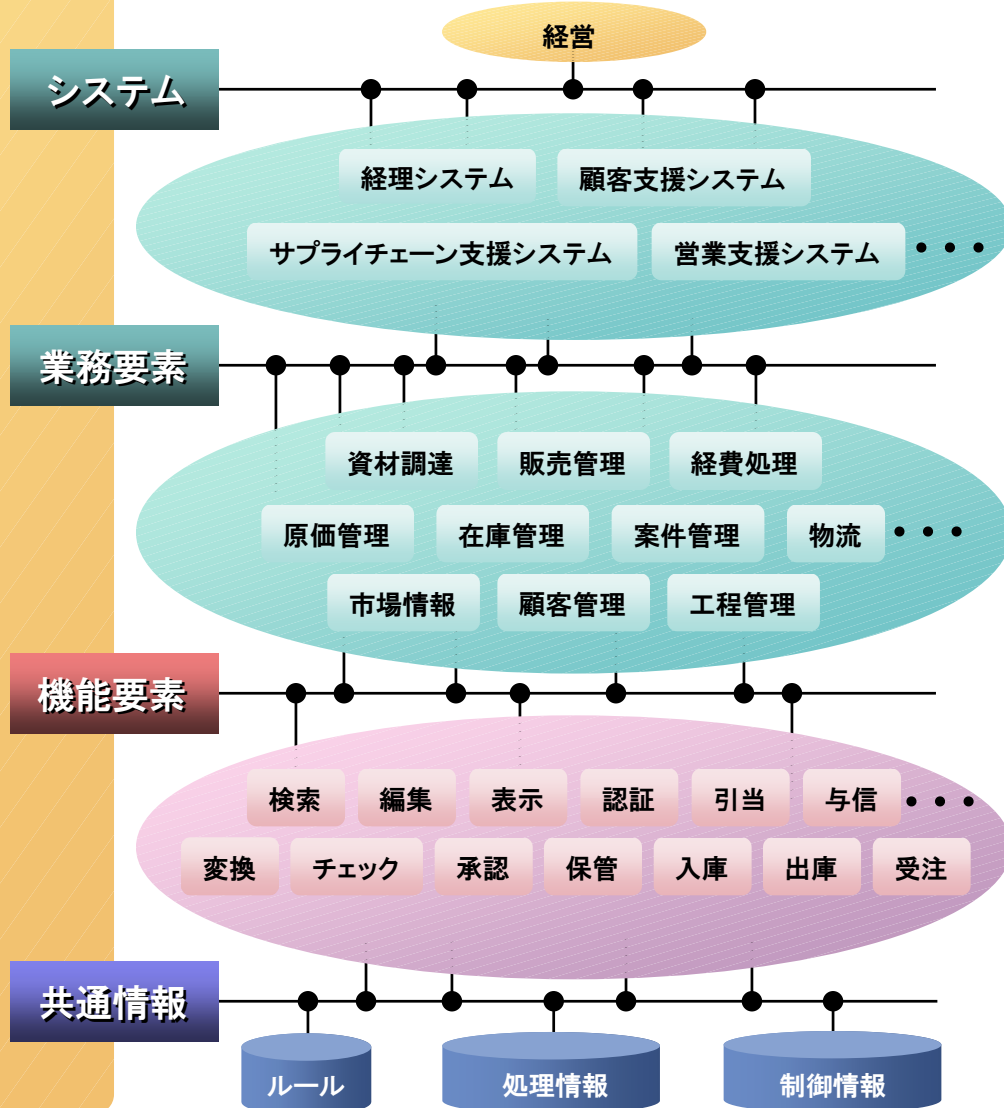
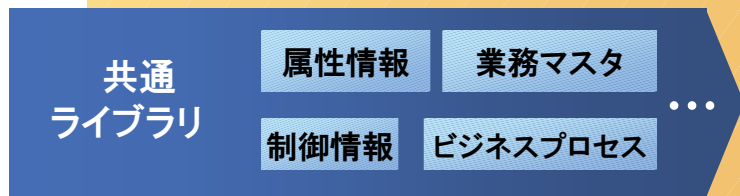
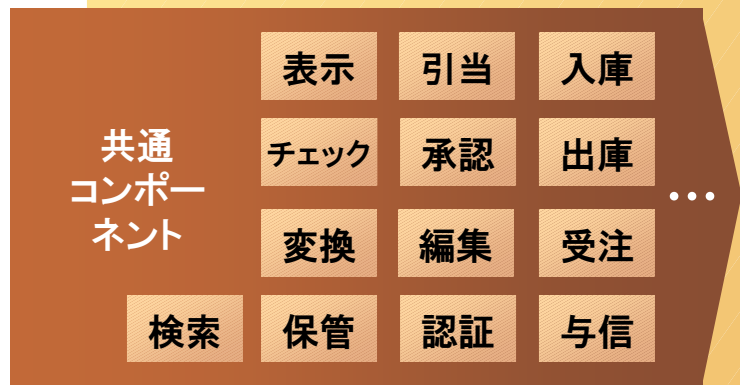
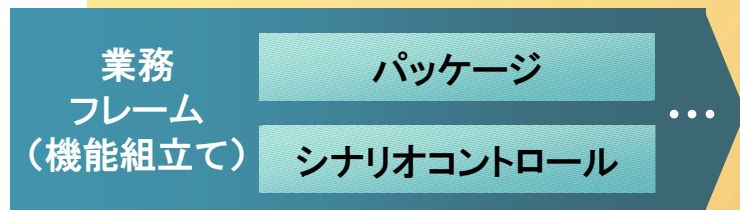
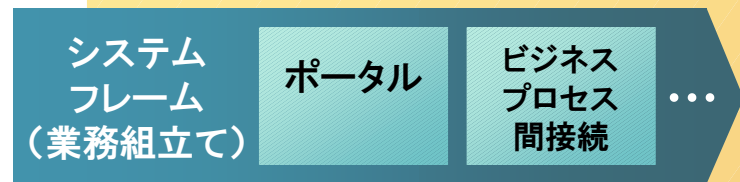
開発コンポーネント	内容
ユーザ認証	ディレクトリサーバ(LDAP)からのユーザ照会と情報取得
HiRDBアクセス	cPersistenceのHiRDBアクセスラッパ
ファイル操作	WebAPIに対し、ローカルファイルのアップロード・ダウンロード、 業務用ロジックへのファイルインタフェースの提供
入力データ変換	コード変換、日付変換など業務AP開発に必要な操作の提供
リストボックス展開	リストボックスへの外部パラメタからのデータ展開

システム機能的な
コンポーネントが中心

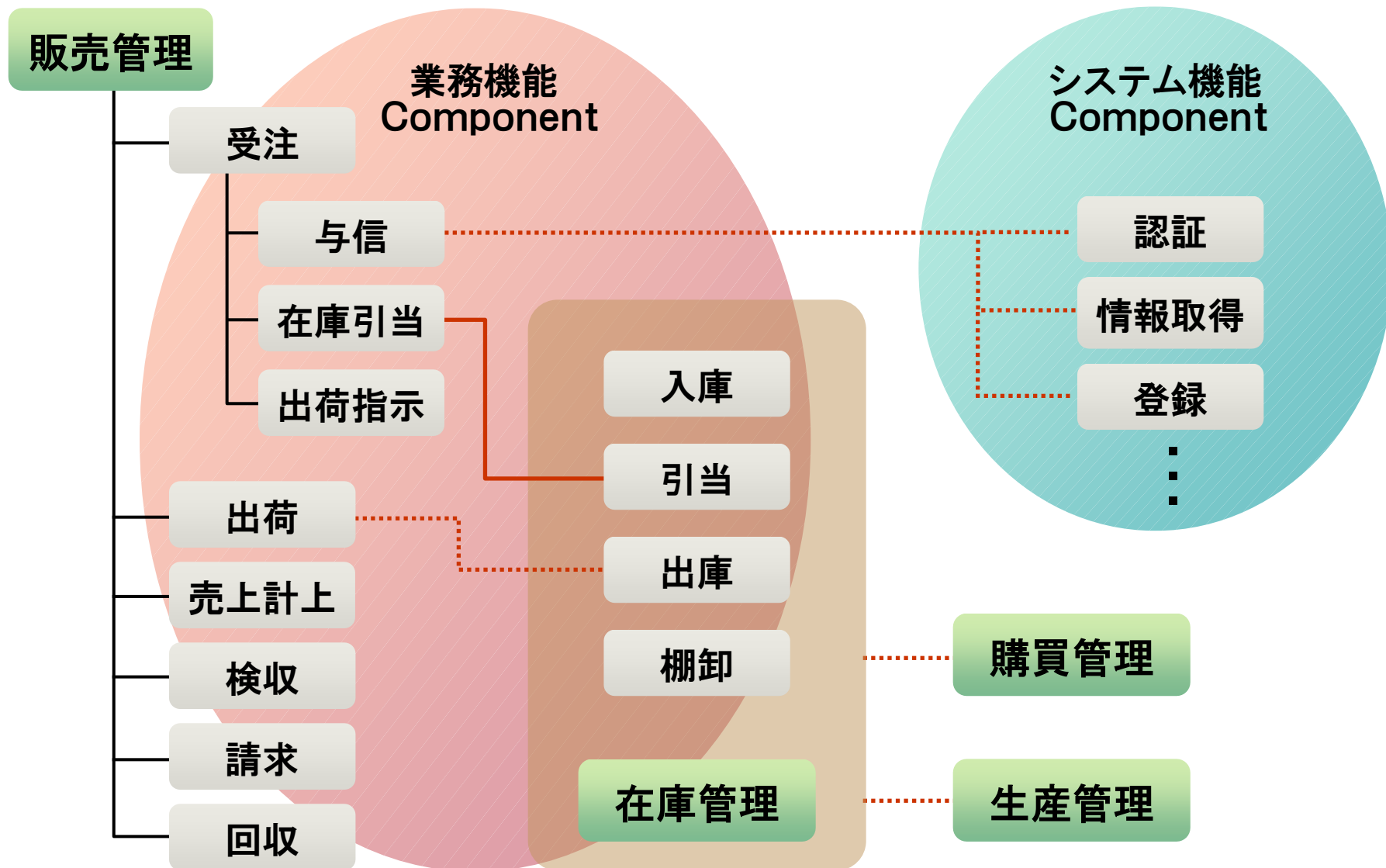
2-3 プロジェクトでの開発プロセス検証



2-4 コンポーネント化対象エリアの検討



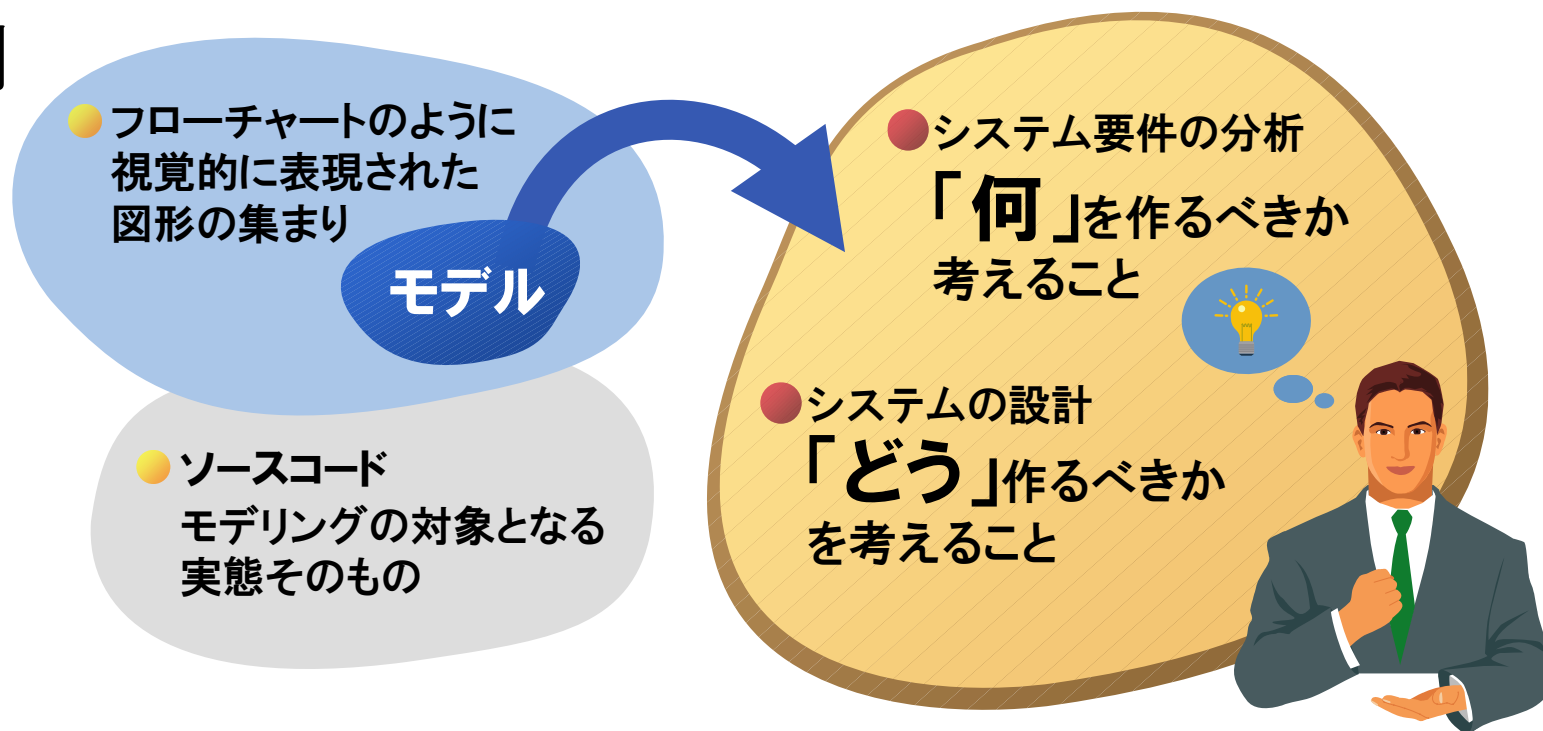
2-5 コンポーネントの粒度



2-6 モデリングとは

モデリングとは	モデルを作ること
モデルとは	情報システムなどの開発において 複雑な構造を抽象化し、単純化したもの

例



MDA(Model-Driven Architecture): モデル駆動型アーキテクチャ

- モデリング主導のシステム開発を実現する参照アーキテクチャ
- OMG*（国際標準化団体）が普及活動中

UML(Unified Modeling Language): 統一モデリング言語

- 分析／モデリングの際に利用する「言語」
- 1997年にOMGが標準化

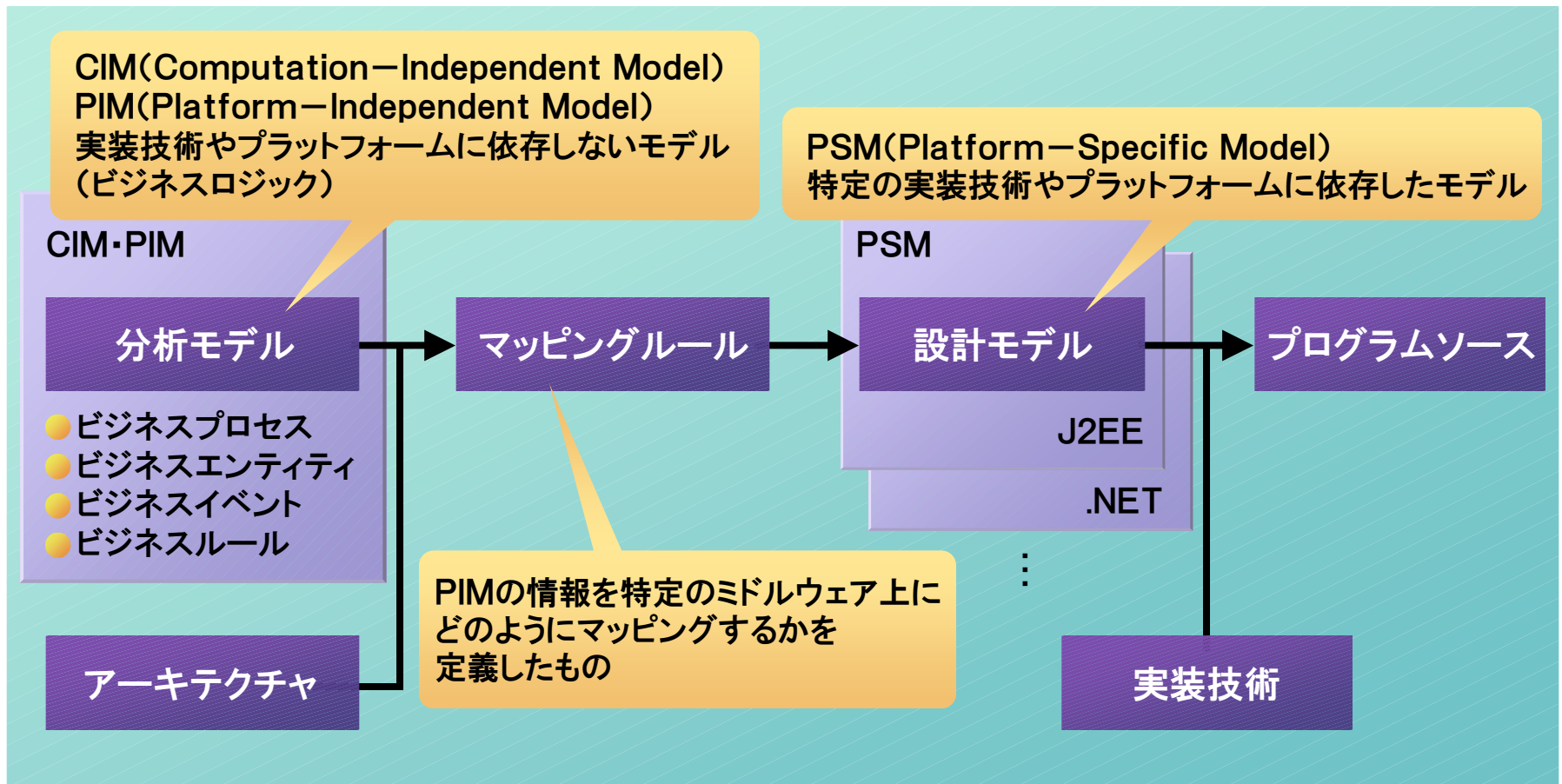
*OMG(Object Management Group):

オブジェクト指向技術の標準化、普及をすすめるため、1989年に設立された標準化団体

2-8 MDAによる開発アプローチ

● MDA (Model-Driven Architecture)

業務モデルとITモデルを分離し、それぞれ独立したモデルで定義し、プラットフォーム決定時、業務モデルを設計モデルにマッピングする体系。



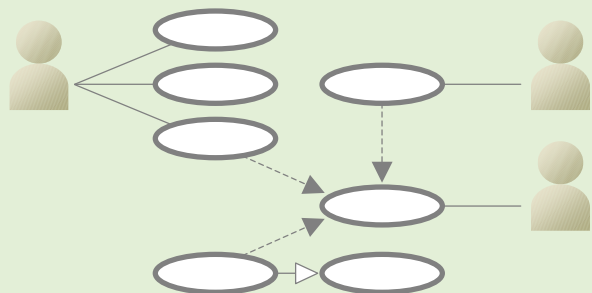
2-9 UMLの種類 (UML2.0の例)

UML種別	概 要
アクティビティ図	業務フローなどアクションの実行の流れを表現できる
ユースケース図	アクター(システム外のユーザや他のシステムなど)との間での外部アクションとその関係を表現できる
クラス図	クラス(型)が持つ属性と、その操作の関係を表現できる
オブジェクト図	クラス図を導き出すための補助的なものであり、クラス図と異なるのはインスタンス(クラスからの生成物)を表現する
ステートマシン図	特にクラスの中で状態がある場合、その状態遷移を表現する
シーケンス図	複数のオブジェクト間のメッセージの流れを時間軸中心で表現できる
パッケージ図	クラスなどをグルーピングして表現できる
複合構造図	全体一部分構造をもつクラスの実行時の構造を示す
コミュニケーション図	インスタンス間の相互作用を構造中心に表現する
インタラクション・ オーバービュー図	条件によって異なる動作をするシーケンス図を、アクティビティ図の中に含めることで表現する
タイミング図	インスタンス間の状態遷移や相互作用を時間制約付きで表現する
コンポーネント図	実装コンポーネントとその関係を表現できる
配置図	ハードウェアとその間の関係を表現できる

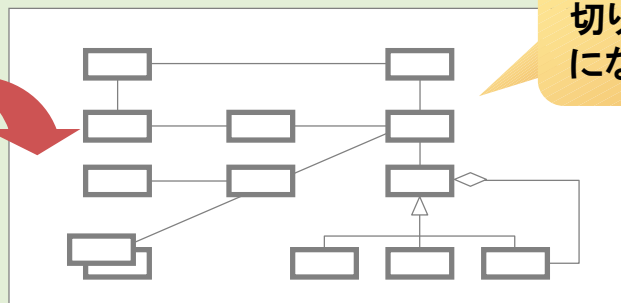
2-10 コンポーネントベースモデリングの特徴

一般的な手法

膨大なユースケース



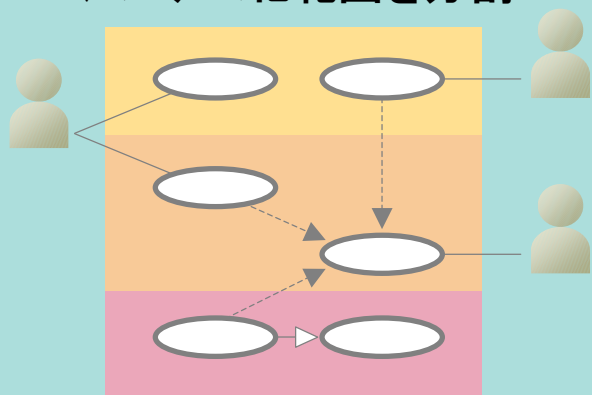
膨大なクラスと複雑な関連



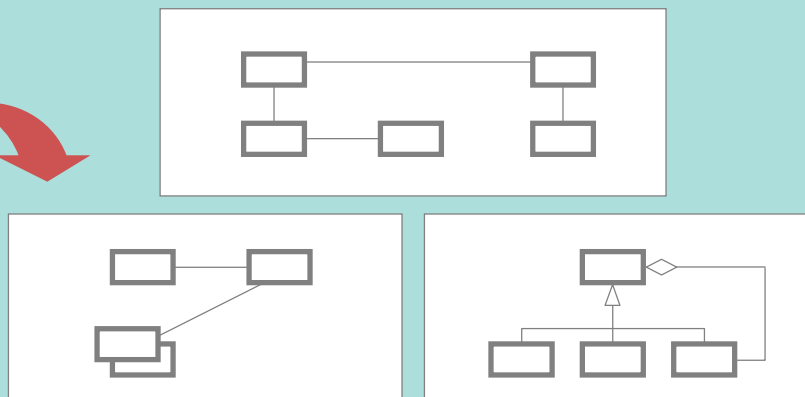
コンポーネントの切り出しが困難になる

コンポーネントベース

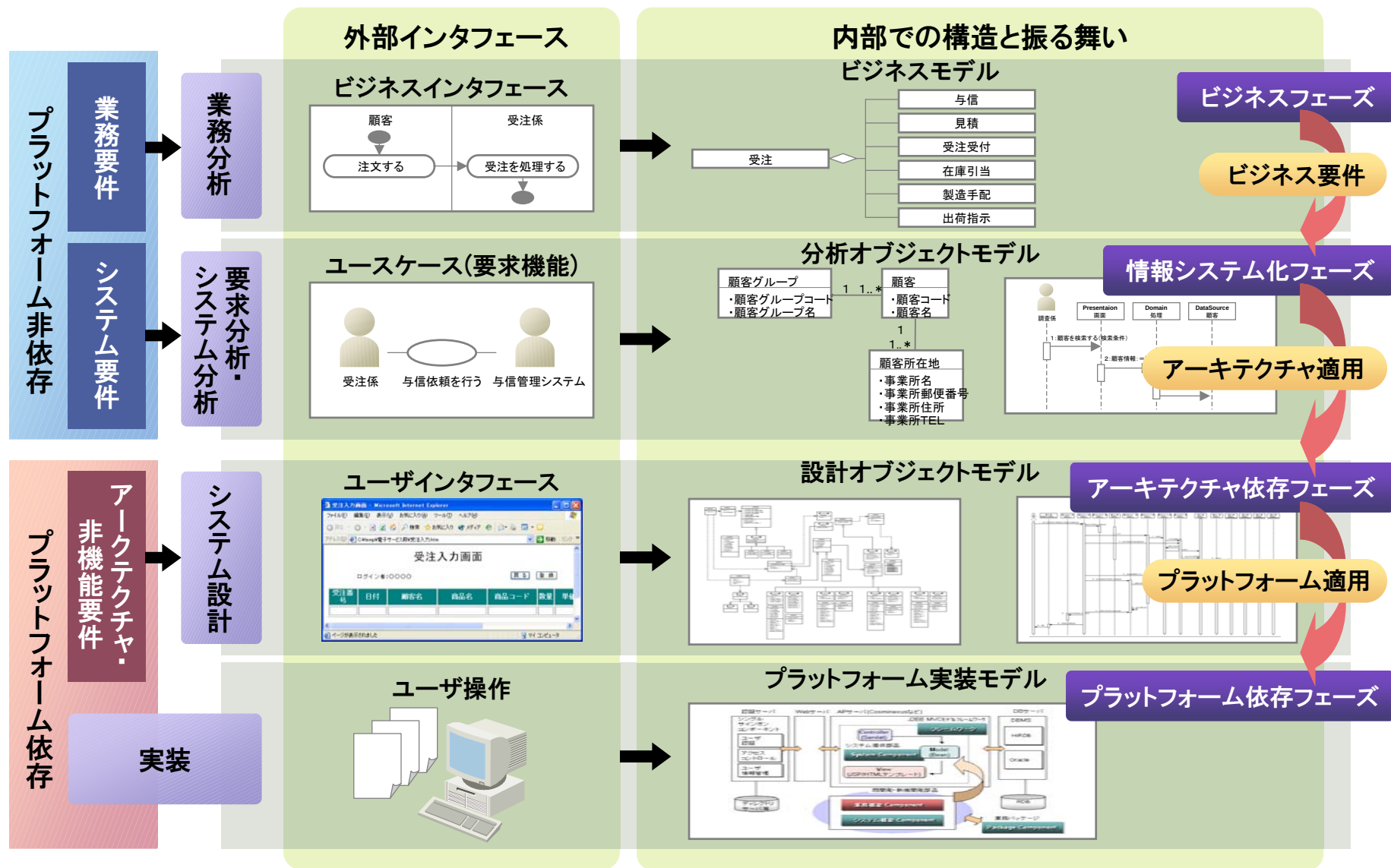
モデリングの段階からシステム化範囲を分割



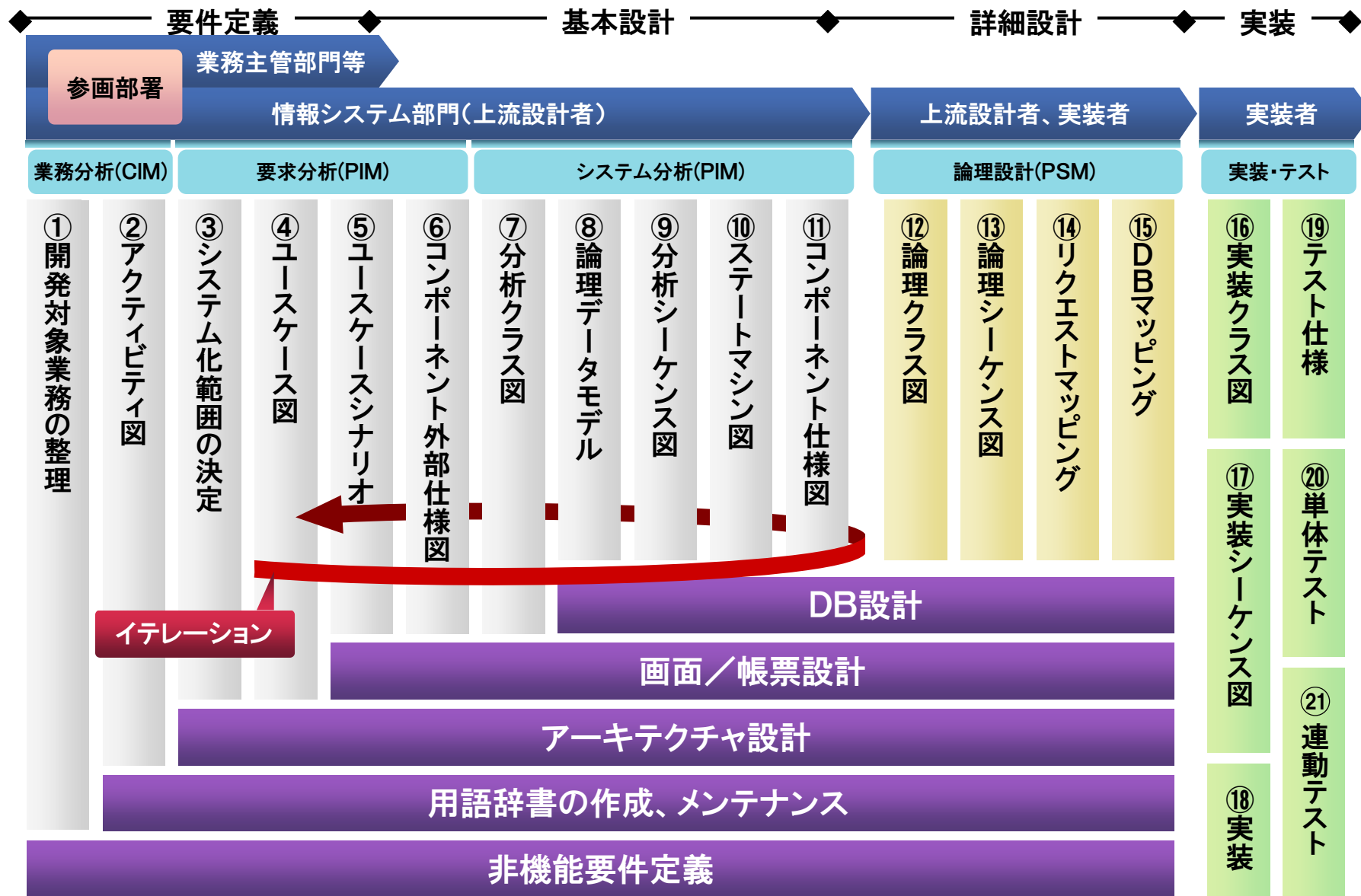
複数のシンプルなクラス図

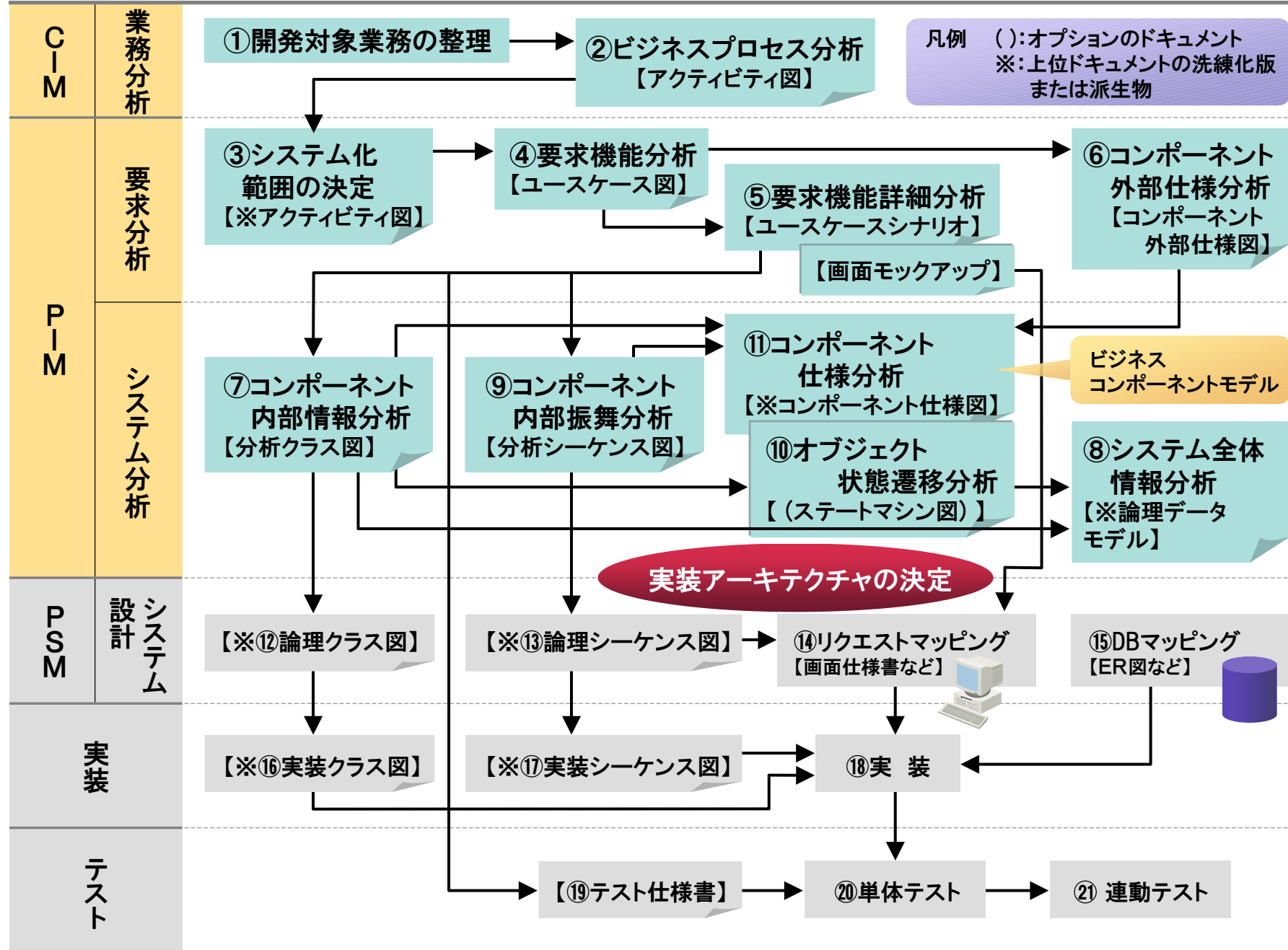


2-11 コンポーネントベースモデリングの全体像



2-12 コンポーネントベースモデリング開発プロセス

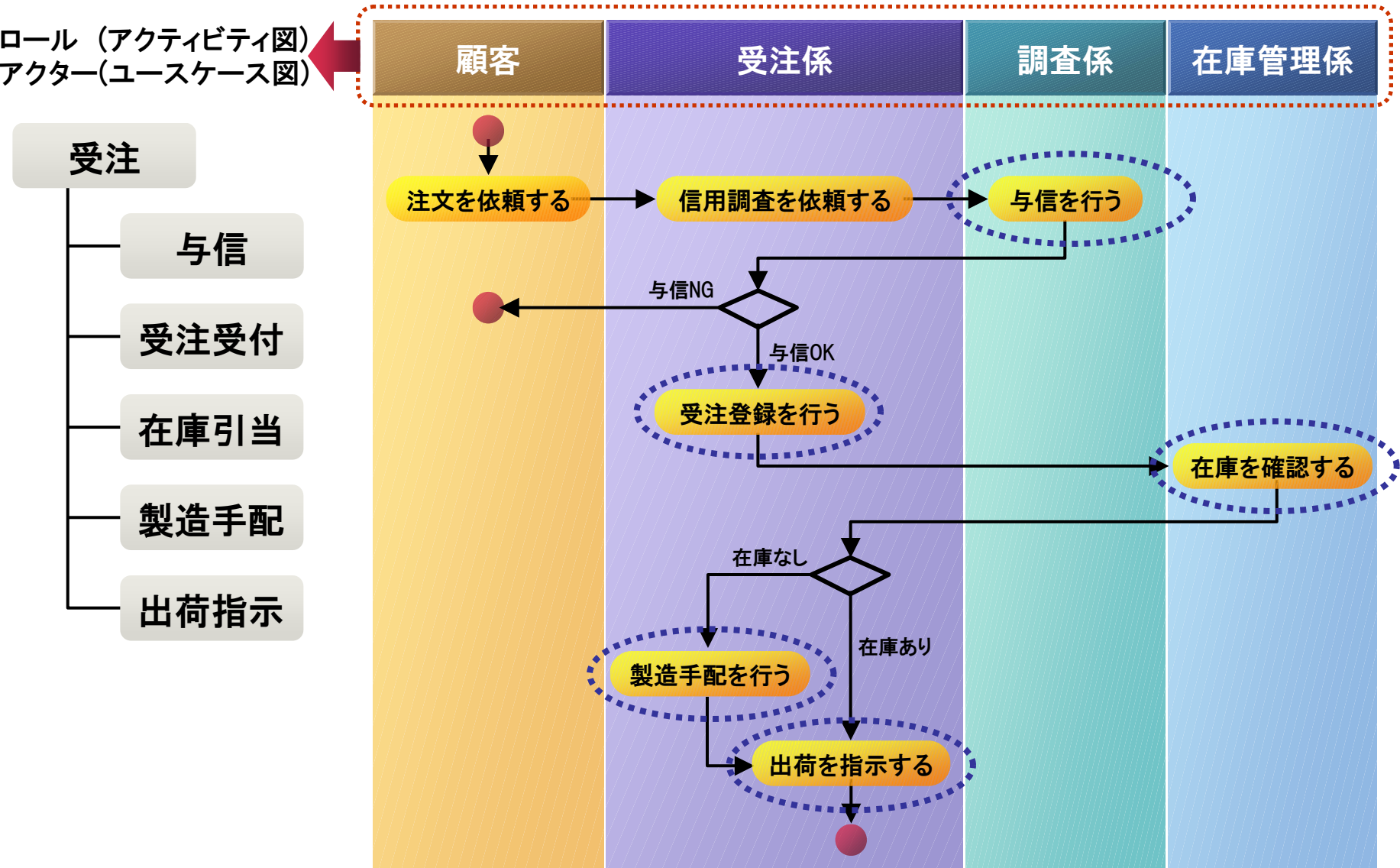


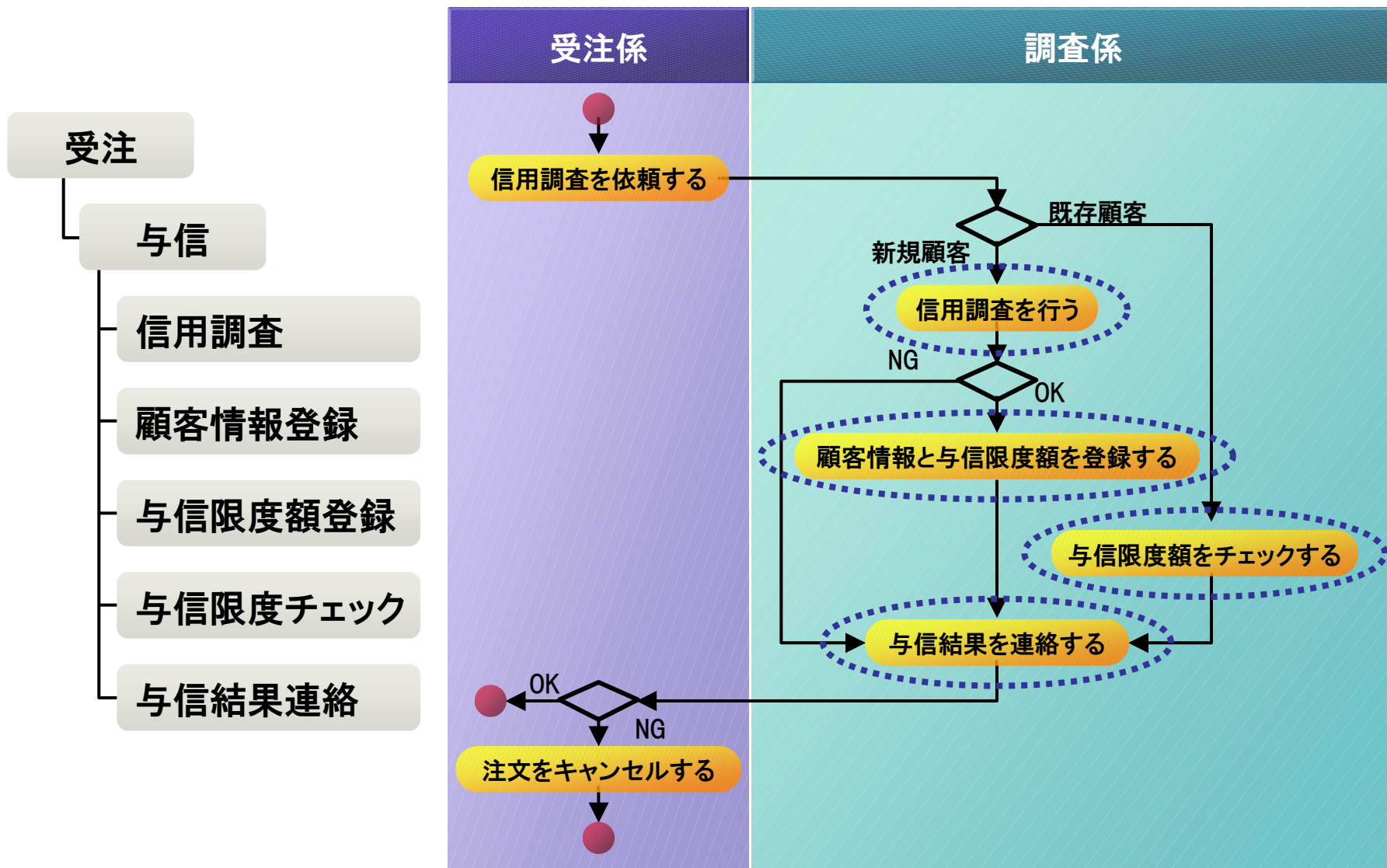


Contents

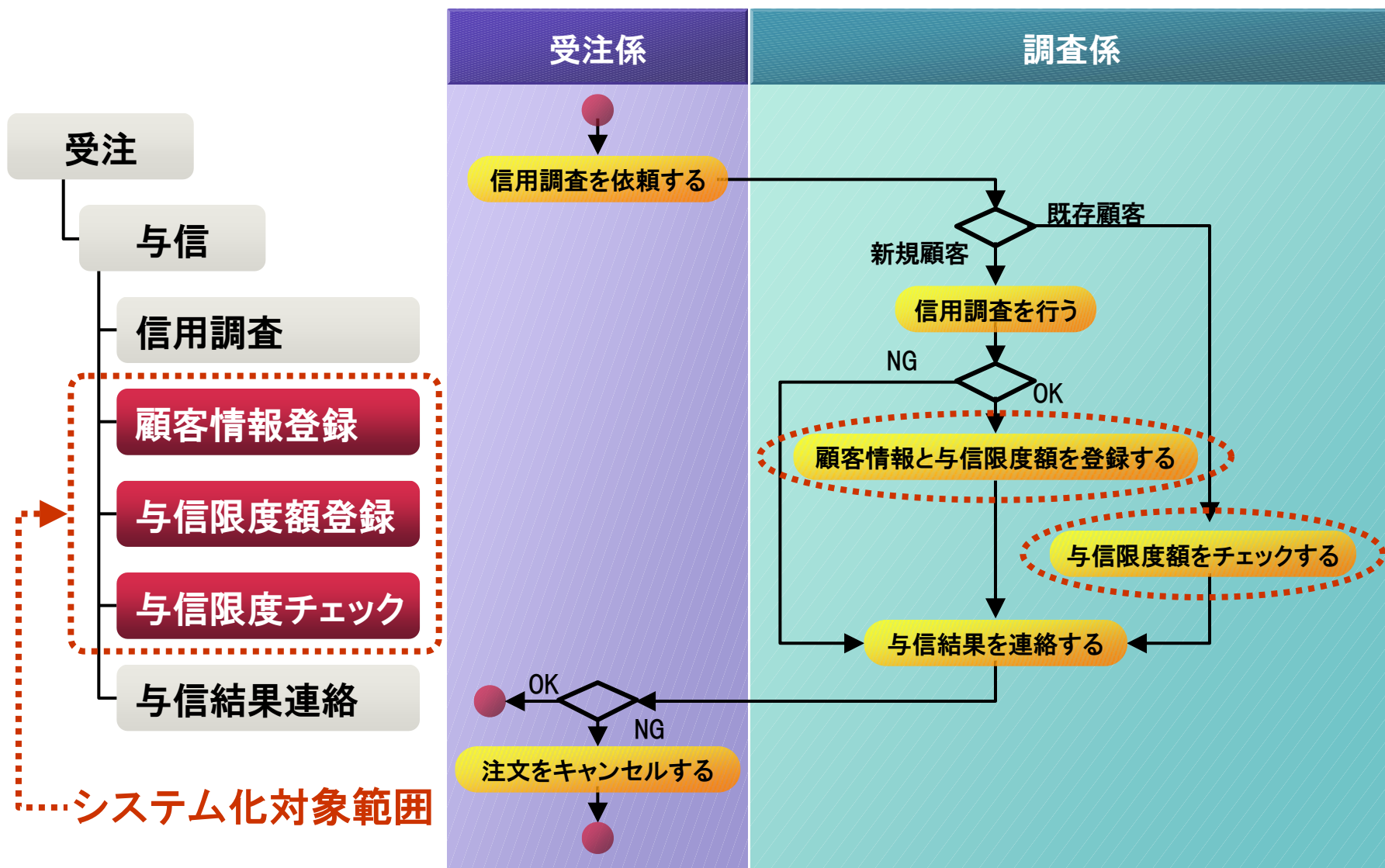
1. SOAとコンポーネントの事例
2. コンポーネントベースモデリングの背景
- 3. コンポーネントベースモデリングの事例**
4. プロジェクト推進のポイント

ロール (アクティビティ図)
アクター(ユースケース図)

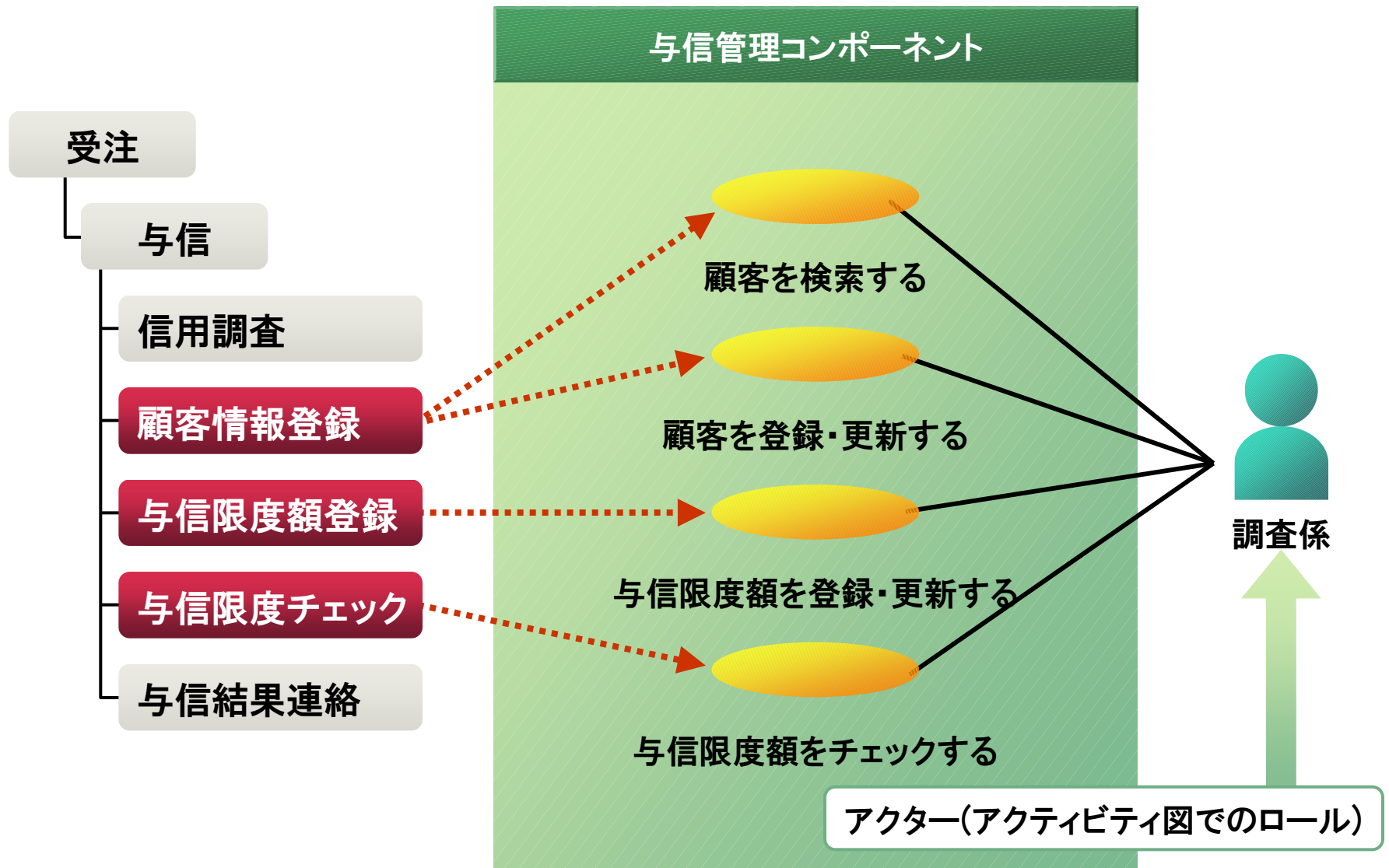


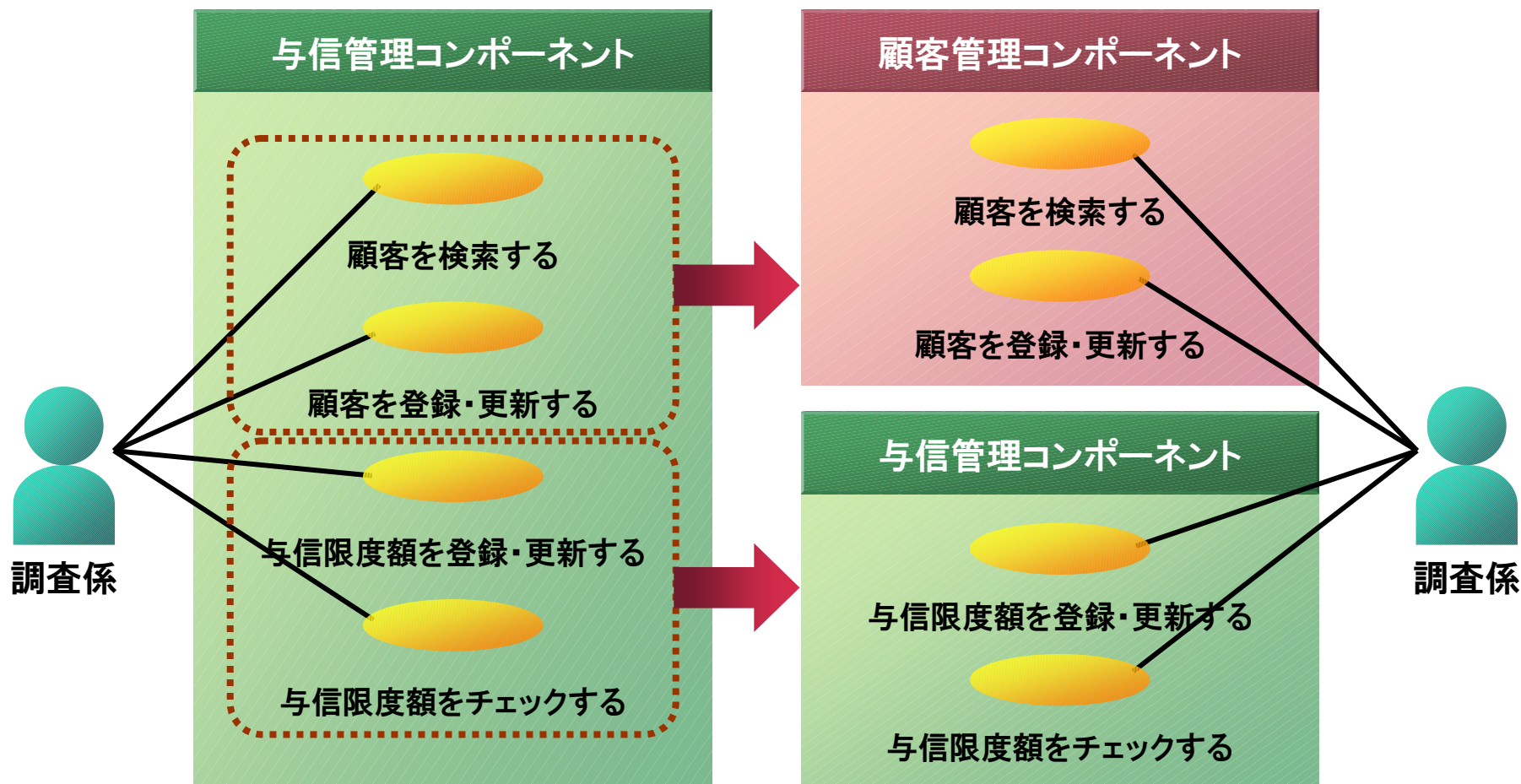


3-3 システム化対象範囲の決定



3-4 ユースケース図





ユースケースでの共通事項を見出し、
必要があればコンポーネント分割を行う

3-7

分析クラス図

● ユースケースシナリオ

[illegible]

● 画面モックアップ

顧客情報

顧客Gr

顧客名

住所

TEL

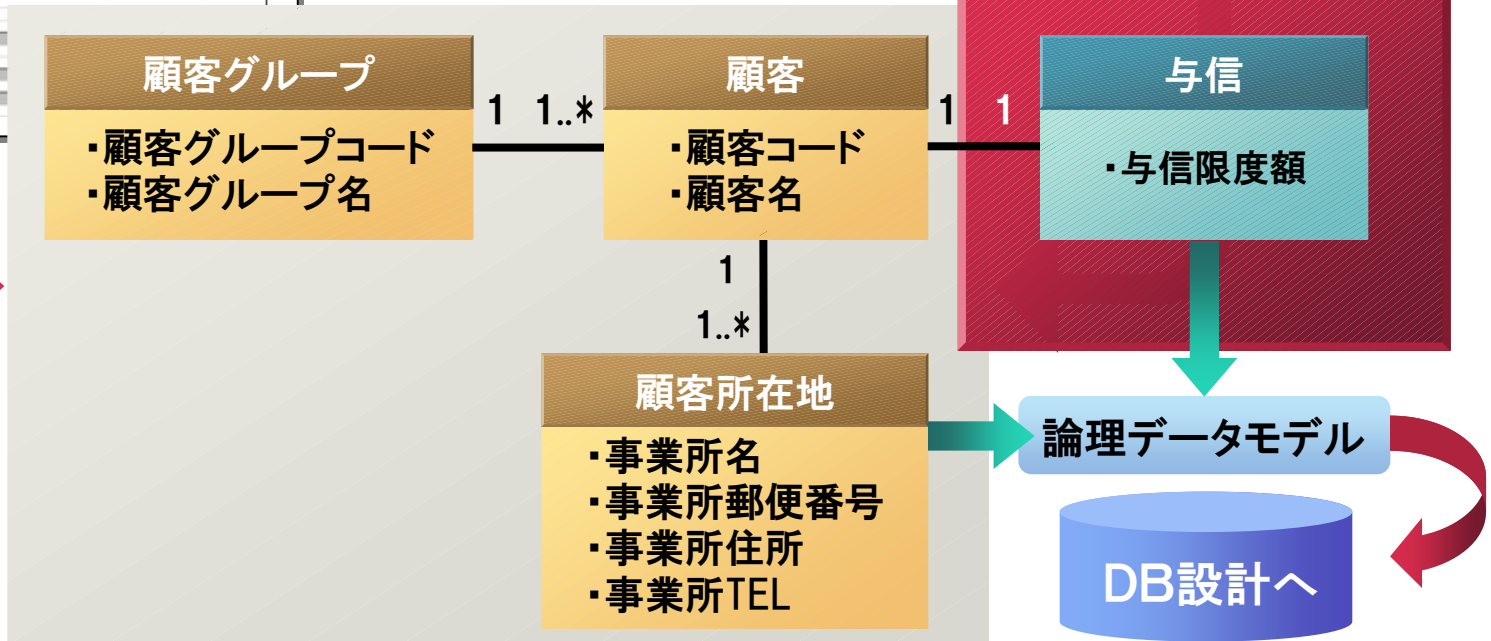
登録

削除

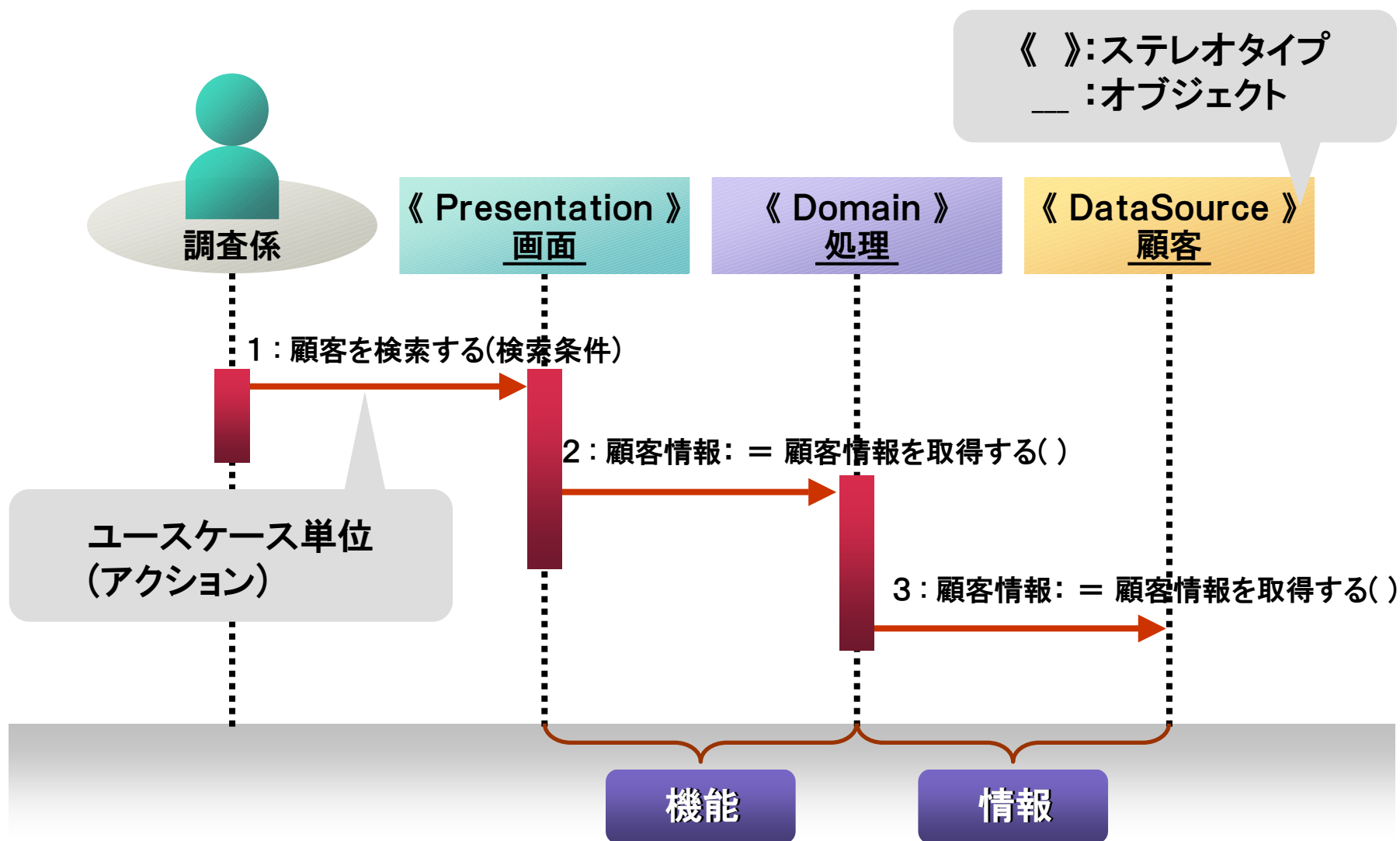
検索

検索

● 分析クラス図

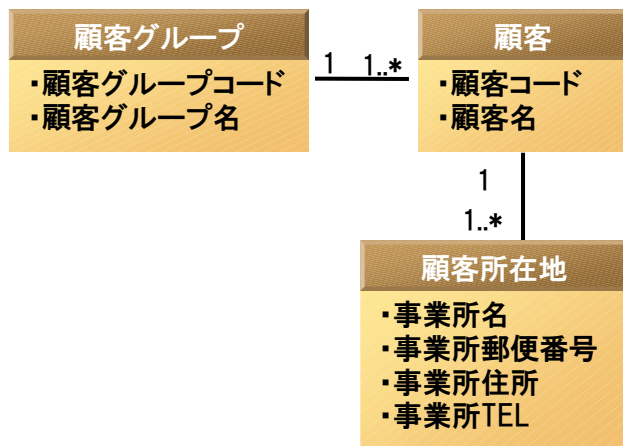


3-8 分析シーケンス図

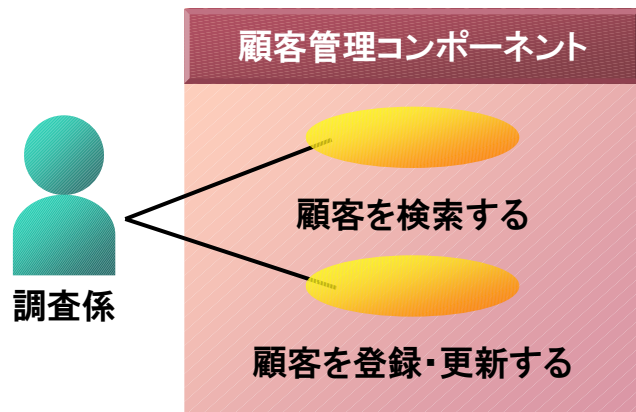


3-9 コンポーネント仕様図 (パッケージ図)

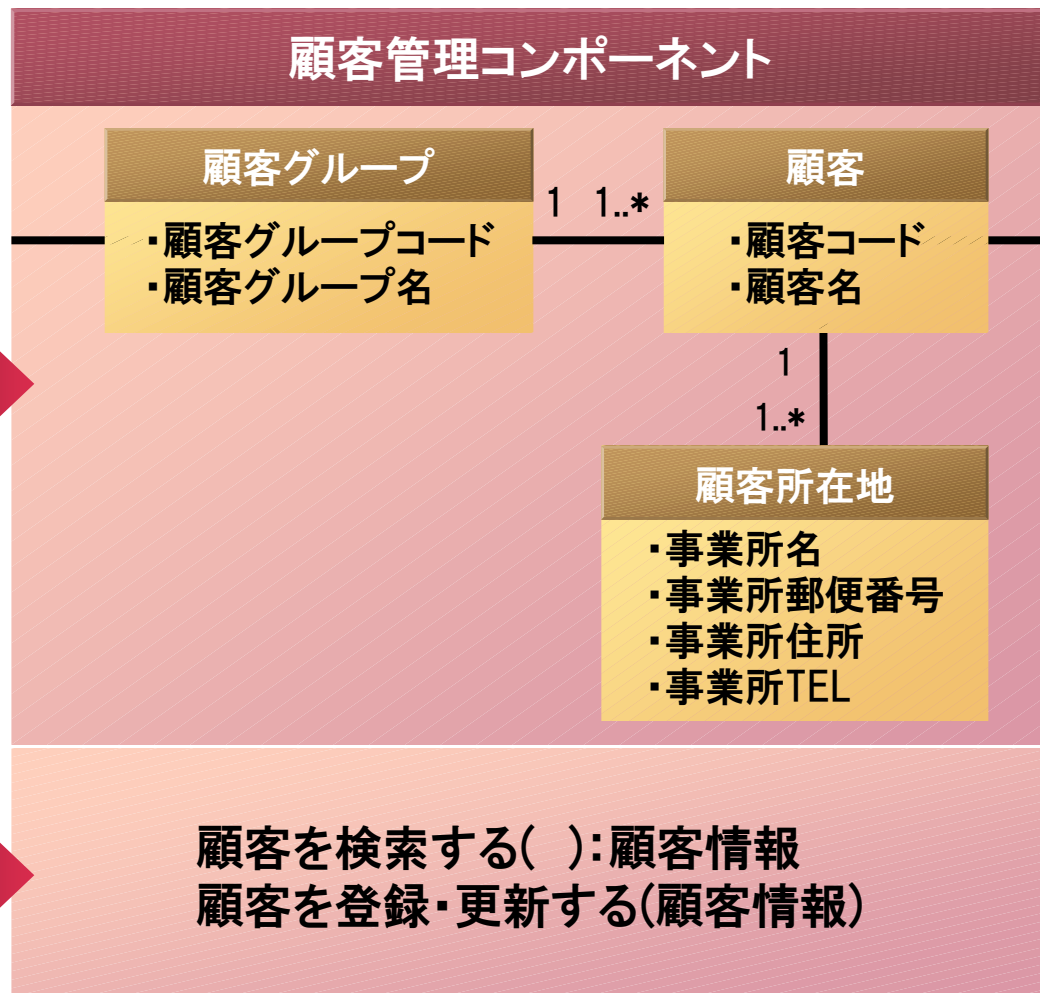
● 分析クラス図(内部仕様)



● ユースケース図(外部仕様)



● コンポーネント仕様



3-10 コンポーネント構成仕様書

コンポーネント構成仕様書

システム名称: 販売管理システム

ステップ数 ●: 既存流用 ○: 自動生成 ▲: 今回開発分流用 △: 新規開発

No.	業務	システム	業務機能コンポーネント										システム機能コンポーネント																			
			No.	ユースケース	シナリオNo	シナリオ				ステップ数				実装ステップ数				No.	ユースケース	シナリオNo	シナリオ				ステップ数				実装ステップ数			
						a	b	c	d	▲	△	a	b	c	▲	△	a				b	c	▲	△	a	b	c	▲	△			
1	与信	A-IM	2.1	信用調査を行う 与信限度額を照会する 与信結果を入力する 与信結果を連絡する	A-IM-001 A-IM-002 A-IM-003 A-IM-004												1.1	顧客情報を検索する	A-IM-001-01						2							
2	受注受付	B-IM	2.1	受注内容を入力する	B-IM-001							10						2.1.1	見積情報を検索する	D-IM-001-03					2							
																		2.1.2	顧客情報を検索する	A-IM-001-01					2							
																		2.1.3	受注情報を登録する	B-IM-001-01					1							
3	在庫引当	C-IM	4.1	在庫を照会する	D-IM-001																											
			4.2	在庫を引き当てる	D-IM-002																											
			4.3	引当結果を連絡する	D-IM-003																											
4	見積	E-IM	4.1	見積内容を入力する	D-IM-001							6						2.1.1	顧客情報を検索する	A-IM-001-01					2							
																		2.1.2	見積額を算出する	B-IM-001-01					2							
																		2.1.3	見積情報を登録する	B-IM-001-02					1							
																		2.1.4	見積情報を検索する	B-IM-001-03					2							
			4.2	見積額を算出する	D-IM-002																											
						0	0	16	0	0	0	0	0	0	0	0	ステップ数 小計						0	0	16	0	0	0	0	0	0	
								16										① 総ステップ数						0	0	16	0	0	0	0	0	0
								16										② 新規開発ステップ数						0	0	16	0	0	0	0	0	0
								0.0										③ 新規開発分での自動生成率						0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
								0.0										④ 総ステップ中の既存・今回開発流用率						0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
								0.0										⑤ 総ステップ中の今回開発分流用率						0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
								0.0										⑥ 総ステップ中の既存・今回開発流用率 (a+c)/①×100						0.0	0.0	57.1	0.0	0.0	0.0	0.0	0.0	0.0

業務機能コンポーネント
レベルのシナリオ

システム機能コンポーネント
レベルのシナリオ

業務機能コンポーネント
レベルのシナリオ

システム機能コンポーネント
レベルのシナリオ

業務機能コンポーネント+システム機能コンポーネント

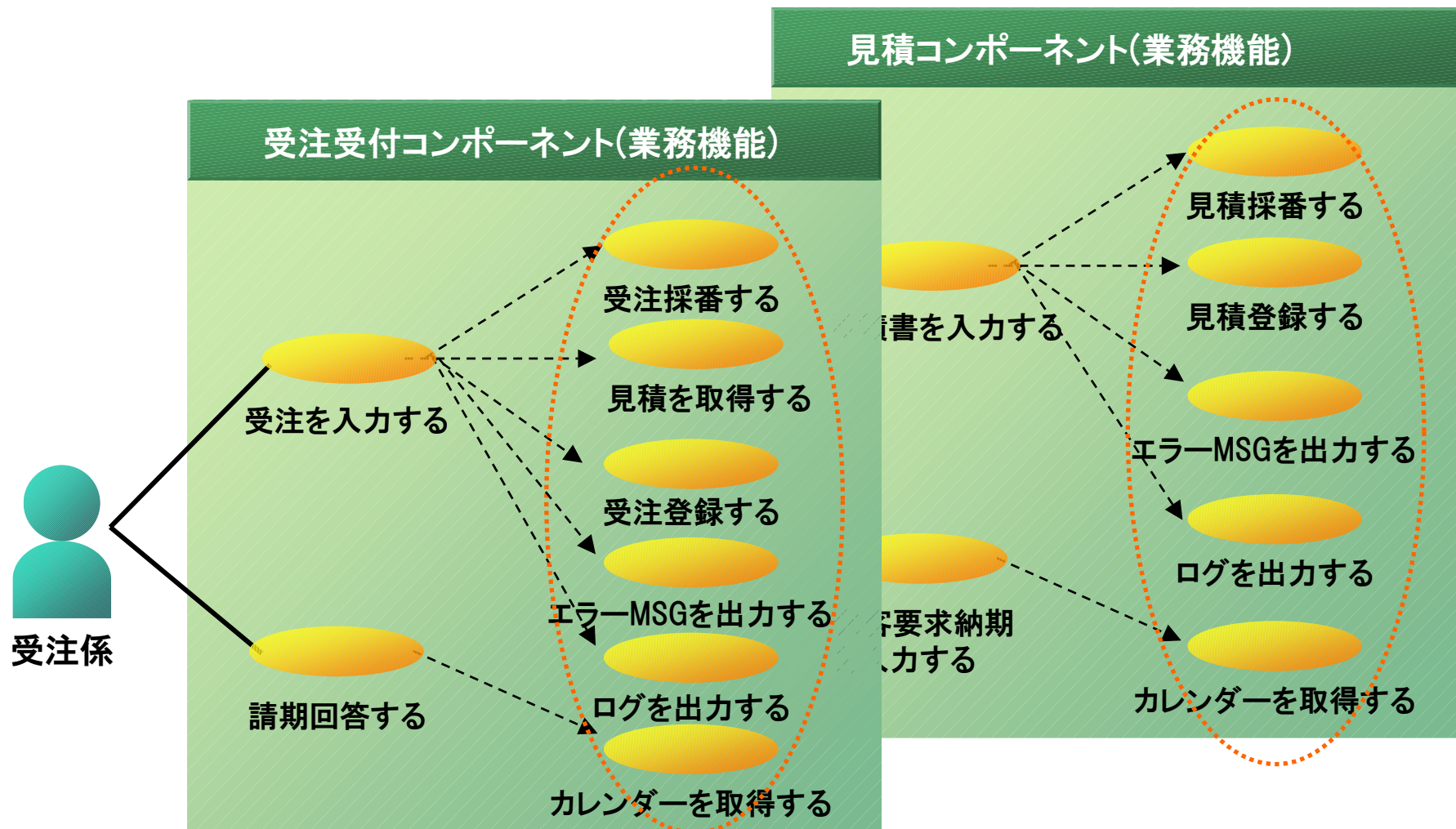
	シナリオ				実装ステップ数			
	a	b	c	d	a	b	c	d
ステップ数 小計	0	0	6	24	0	0	0	0
① 総ステップ数 合計	a+b+c+d				30			
② 新規開発ステップ数 合計	b+d				24			
③ 新規開発分での自動生成率	b/②×100				0.0			
④ 総ステップ中の既存流用率	a/①×100				0.0			
⑤ 総ステップ中の今回開発分流用率	c/①×100				20.0			
⑥ 総ステップ中の既存・今回開発流用率	(a+c)/①×100				20.0			

Excelの計算機能による
自動計算

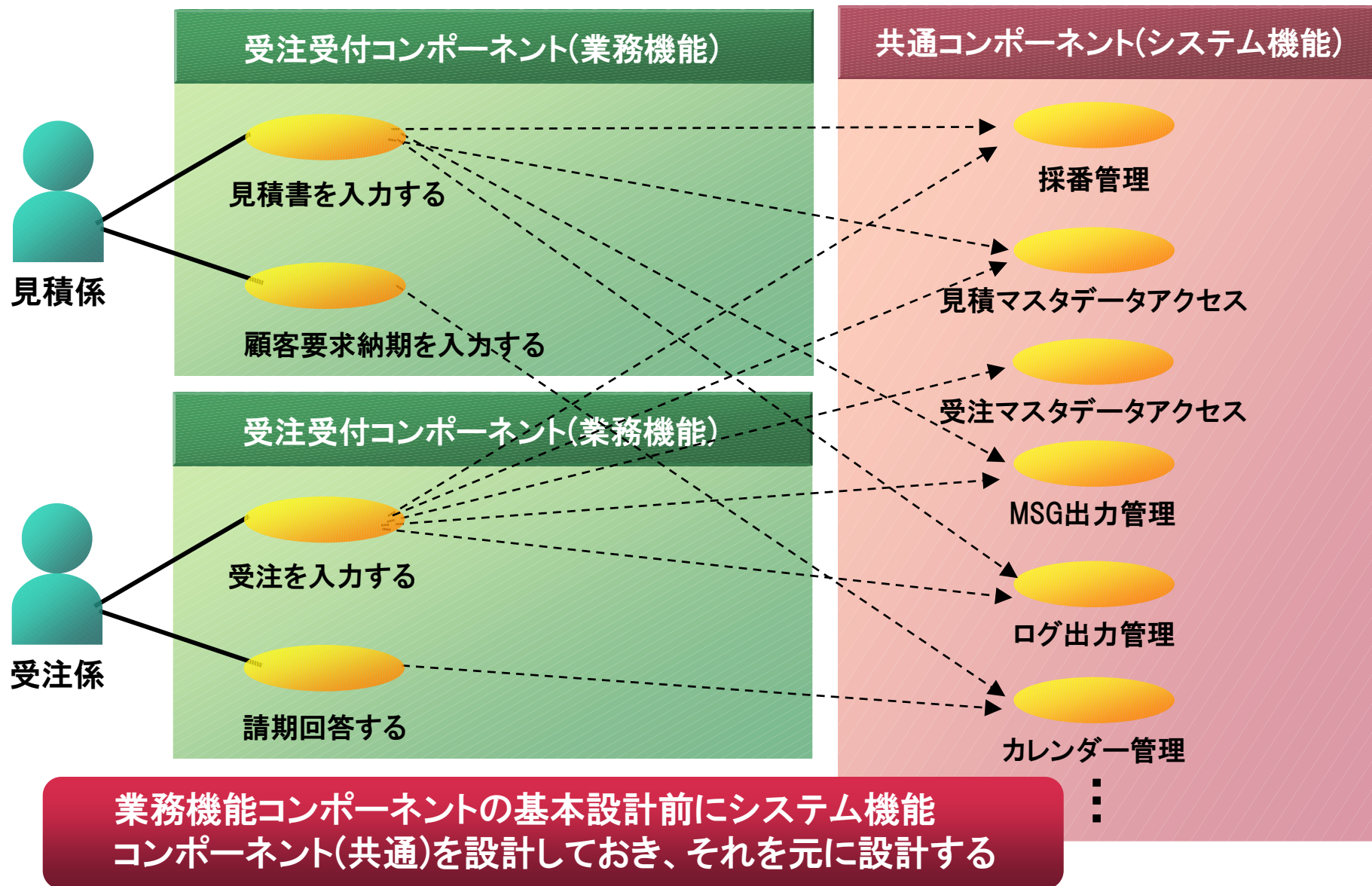
Contents

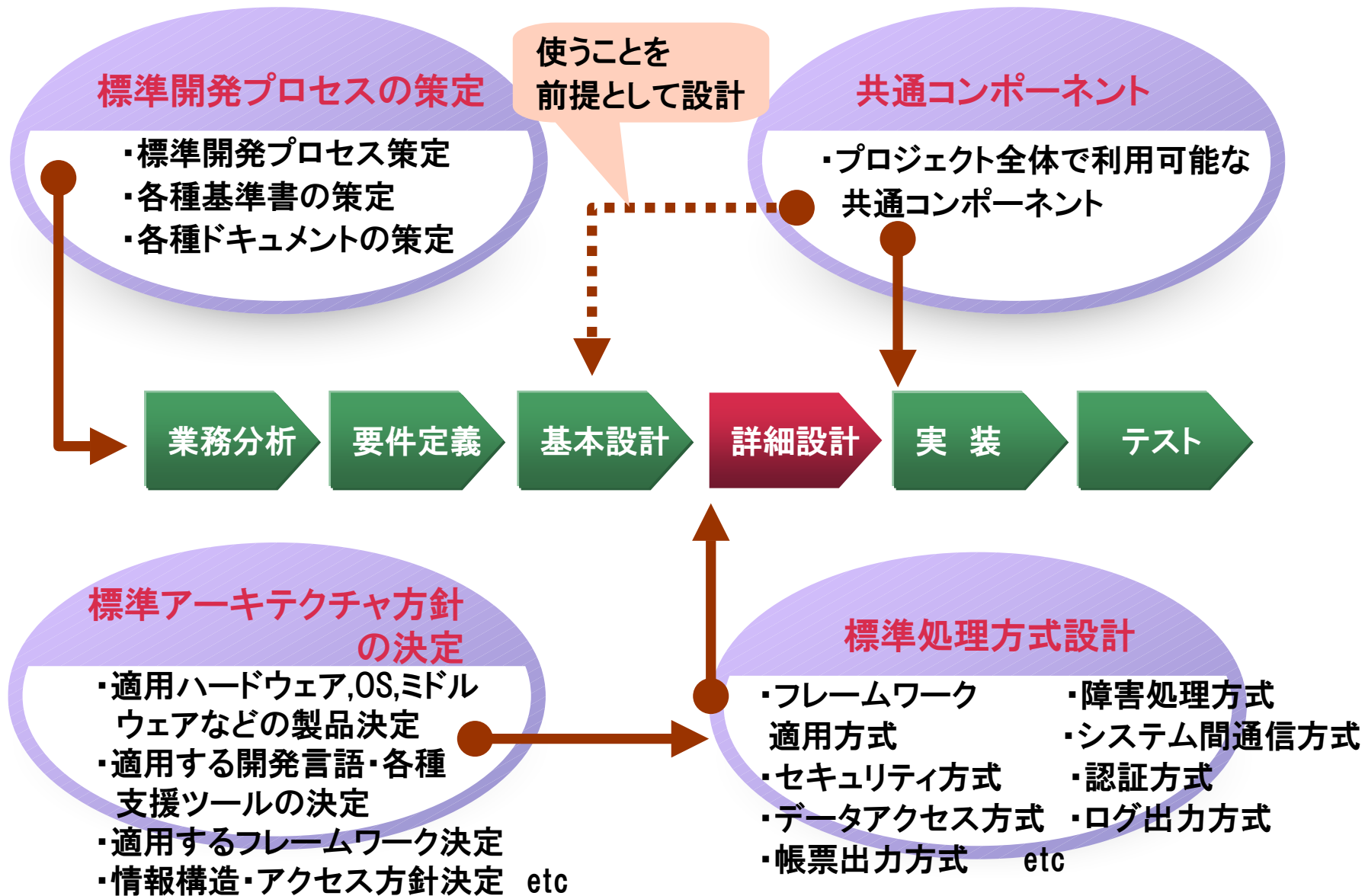
1. SOAとコンポーネントの事例
2. コンポーネントベースモデリングの背景
3. コンポーネントベースモデリングの事例
- 4. プロジェクト推進のポイント**

4-1 システム開発における共通処理機能の重複



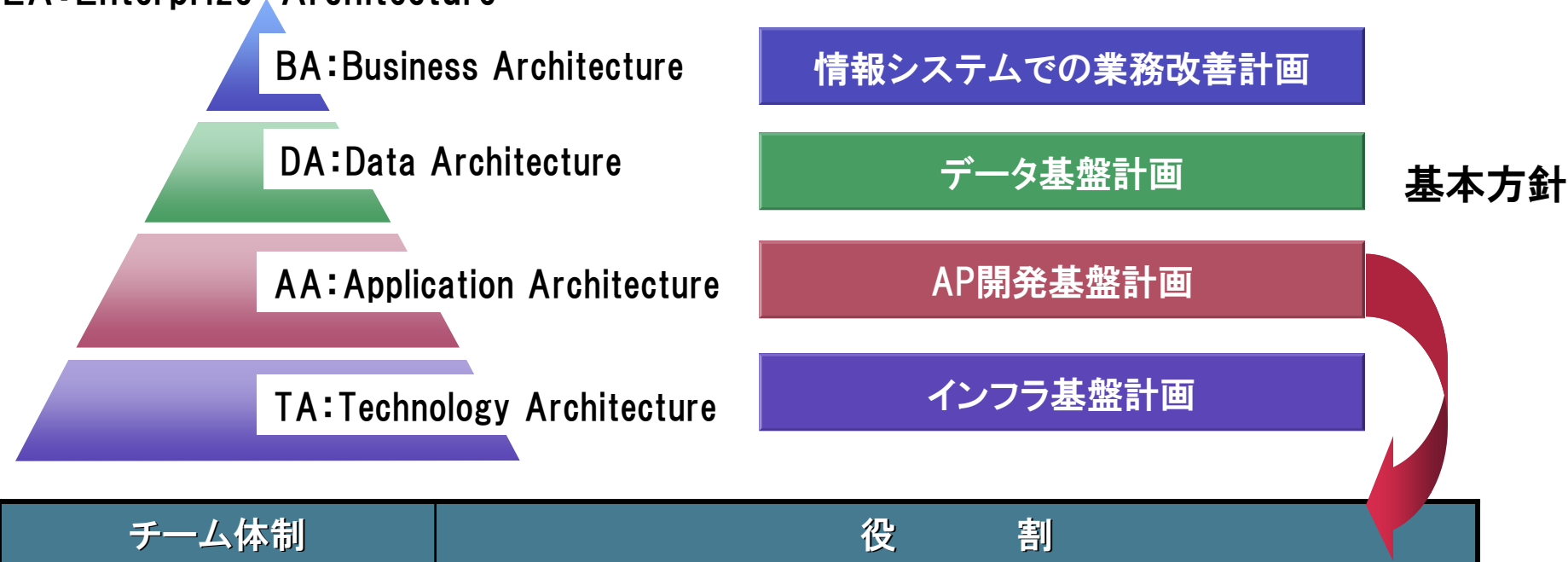
業務機能コンポーネント基本設計時、システム機能コンポーネント(共通)が決まっていないと、重複して作りこんでしまう危険性が高い





4-4 プロジェクト推進体制の事例

EA:Enterprise Architecture



チーム体制	役 割
標準化チーム	標準開発プロセスや各種基準、ドキュメントの標準化、各種アーキテクチャ標準や共通処理方式方針などを決定する
共通コンポーネントチーム	特にシステム機能コンポーネントの提供と保守、管理、APチームへのデザインレビューへの参画
共通マスタチーム	共通マスタの設計とデータアクセスコンポーネントの提供
運用管理チーム	インフラ環境の構築と各種APの運用管理方式決定
AP開発チーム(複数)	各種APの開発

コンポーネント化においては

- 業務分析からのモデリングが必要
- そのための一貫した分析・開発プロセスと、開発支援などの統一が必要
- MDAベースではモデルそのものの再利用化も可能となる
- 作成したコンポーネントを効率よく管理・運用するための体制と仕掛けが必須