

# 情報システム開発の手法1 (データ中心のアプローチ)

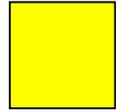
(情報システム開発論、第3回講義)

URL <http://homepage3.nifty.com/suetsuguf/>

Email [fwhy6454@mb.infoweb.ne.jp](mailto:fwhy6454@mb.infoweb.ne.jp)

作成者 末次 文雄 ©

# 復習：プロジェクト管理の内容

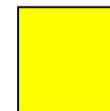


(内容)

(ツール)

	(内容)	(ツール)
組織化	<ul style="list-style-type: none"><li>・要員確保、責任分担、技術確保</li><li>・命令系統、意思決定法、評価基準</li></ul>	
文書化	<ul style="list-style-type: none"><li>・文書化のルール(構成、詳細度)</li><li>・文書の公報、その手段</li></ul>	CASE
見積り	<ul style="list-style-type: none"><li>・生産性、工数見積り(各フェーズ)</li><li>・コスト見積り、コスト依拠</li></ul>	FP法
要員計画	<ul style="list-style-type: none"><li>・必要スキル・レベル、投入時期</li><li>・教育、訓練</li></ul>	
作業計画	<ul style="list-style-type: none"><li>・全体計画(マスタープラン)</li><li>・作業の詳細化、作業計画(WBS)</li></ul>	PERT図 WBS
進捗	<ul style="list-style-type: none"><li>・納期、品質、コストの予実把握</li><li>・問題点の発見、解決策</li></ul>	

# 復習：（続き）

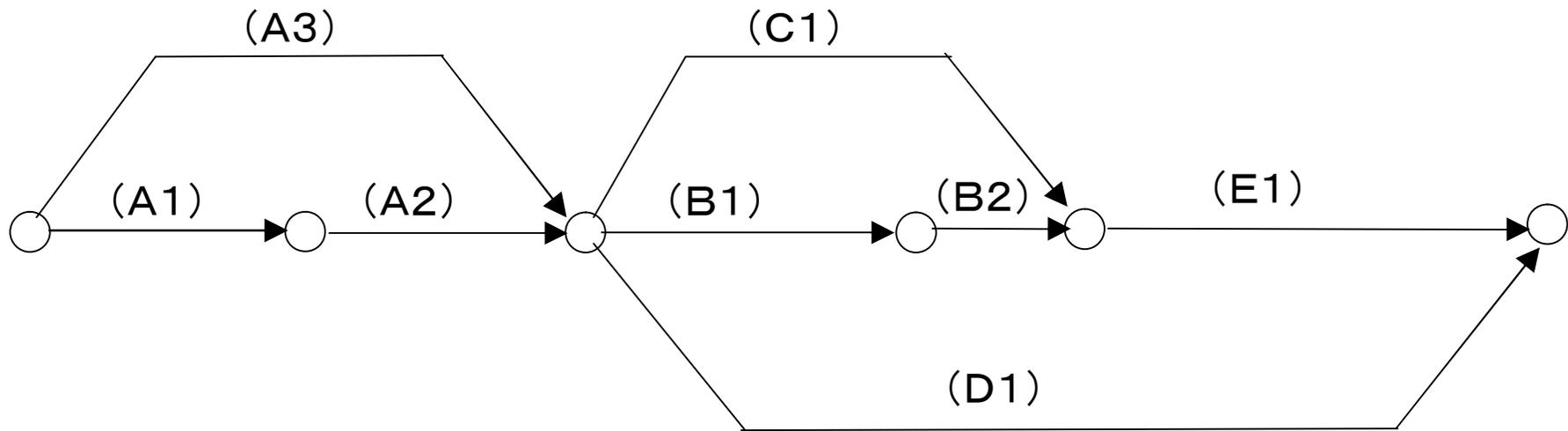
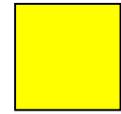


（内容）

（ツール）

品質管理	・ <b>目的に合った機能</b> 、レスポンス、信頼性、回復方法、使いやすさ、保守が容易か、レビュー時期	
コスト管理	・ 予算立案（人件費、機械費、材料費、外注費、経費、運用費） ・ 予定実績差異（費目別）	
問題管理	・ 問題点の登録、解決策評価、進捗	
変更管理	・ 変更点の把握、影響度 ・ 変更指示、公報、変更作業の進捗	
外注管理	・ 範囲、見積り取得、委託先の選定 ・ 外注要員の評価、進捗、受入・検収	
リスク管理	・ <b>リスク想定</b> 、リスクの査定、回避策	

# 復習: PERT図の例示



- ・A2はA1の後続作業
- ・(A1、A2)とA3は併行作業
- ・B1、C1、D1は、A3、A2の後続作業

# 目次(情報システム開発手法1)

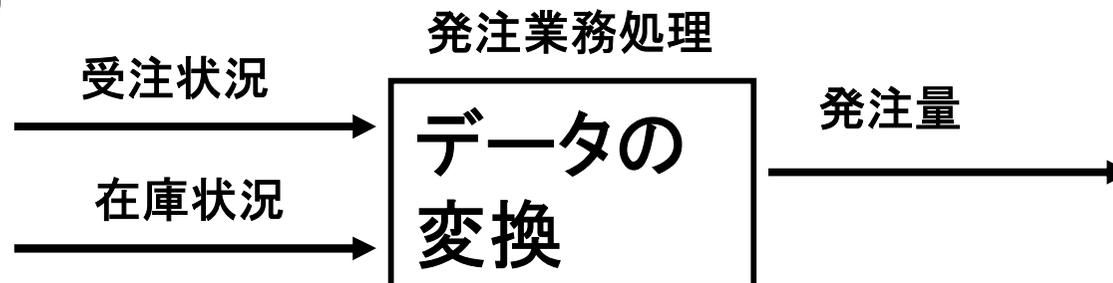
1. データ中心型アプローチとは
2. 開発手法の種類
3. 上流工程でのDOA手法
4. 開発工程とDOA手法の関係
5. DOA型開発の実際
6. まとめとレポート課題

# 1. データ中心型アプローチとは

## 1.1 DOAの考え方

- ・業務処理の流れをデータの流れとして把握し、処理は単にデータの変換に他ならないものとする。
- ・データ項目は永続性がある。(処理は変わるが)
- ・それによって、システム開発の上流工程での品質確保、開発効率の向上をねらった手法。

(例)



## 1. 2 DOAの経緯

「過去のシステム開発手法を集約したものである」

### ①DFD(Data Flow Diagram)

- ・‘70年代の構造化設計、分析手法で確立

### ②データの正規化(Data Normalization)

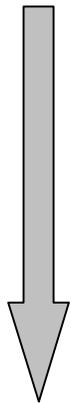
- ・1970年、コッド博士が提唱した関係データベースの基礎理論

### ③ER図(Entity-Relationship-Diagram)

- ・1976年、チェン氏が発案したデータモデル記述方法

# 1. 3 DOAが生まれた理由

## 処理中心／機能中心の開発



- ・情報処理システムが扱うデータ量やデータの種類が少なく、演算・処理コストが高かった時代。コスト最小化のために、必要最小限機能に絞り込む必要があった。

## データ中心の開発

- ・扱うデータの量・種類の増大により、処理機能の規模・複雑さが飛躍的に増大し、確実なシステム開発が困難になってきた。
- ・ディスク装置が高密度・安価になってきた。

# 1.4 DOAの特徴

(特徴)

- ・データの流れと分析に重点を置いた要件分析、設計
- ・業務工程を、データ中心の手法と表記法(DFD)で表現する
- ・設計の重点は、データ分析、データ構造検討(正規化、ER図)
- ・成果物の標準化による開発工程間のスムーズな接続
- ・変更に強いシステム構造(プログラムとデータの分離)

(効果)

- ・システム要件をシステムに漏れ無く反映
- ・上流工程での品質確保と手戻りの減少
- ・変化／ニーズへの対応しやすい保守の容易さ
- ・システム開発の生産性向上

## 2. 開発手法の種類

### 2.1 手法の種類

#### ① DFD (Data Flow Diagram)

- 適用業務における**データの流れ**を  
4つの単純な記号で表記
- 処理内容を3つの観点で、記述
  - **データフロー記述** (データフローを流れるデータ項目の記述)
  - **データストア記述** (データストアのデータ項目の記述)
  - **処理機能記述** (各プロセスの処理内容を記述)
- トップダウンで詳細化

## ②データの正規化 (Data Normalization)

- ・DB設計に先立って、  
関連性の強いデータ項目を  
正規化理論に基づいて  
グルーピングする
- ・その結果、ダブリなどの冗長さを排除
- ・非正規形 (Unnormal Form) から、  
正規形 (Normal Form) を作成する。

### ③ER図 (Entity-Relationship Diagram)

- ・適用業務が**管理すべき対象をEntity**とし、**Entity間の関連**を簡単な記号で定義する。
  - ・Entity、Entity間の関連(Relationship)は**属性(データ項目)**をもつ。
- ・二つの作成方法がある。
  - ーDFDのデータストアの正規化結果から作成
  - ー対象業務のあるべき姿から作成

#### ④データ・ディクショナリー

- ・全てのデータ項目の辞書
- ・データ項目の名称、意味、桁数、データタイプを記述する。

#### ⑤拡張DFD

- ・リアルタイムシステムの記述用
- ・制御とタイミングを記述する

#### ⑥状態遷移図

- ・システムの状態が、外部的な信号やイベントに対してどのように推移するかを記述（機械の制御、機器のランプetc.）

## ⑦複合設計／構造化設計

- ・プログラムのモジュール分割(細分化)の方法である。
- ・**モジュール分割の良否を判定する尺度**
  - ーモジュール強度(機能的強度、情動的強度・・・)
  - ーモジュール間結合度(データ結合・・・)
- ・複合分析によるモジュール分割
  - ーデータフロー中で、大きくデータ構造が変わる
  - ー共通機能を取り出す

## ⑧処理機能記述用のダイアグラム

- ーデシジョン・テーブル
- ーシュード・コード(擬似コード)
- ー構造化文
- ーN-Sチャート、IPO図

## 2. 2 各手法の使用工程

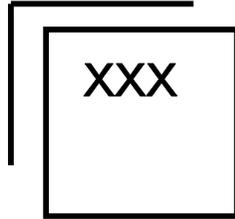
- ・DFDは、要件定義、外部設計で使用
- ・データ正規化も、主として要件定義、外部設計で使用
- ・ER図も、主として要件定義、外部設計で使用
- ・データディクショナリーも主として要件定義、外部設計で使用

- ・状態遷移図は、外部設計、内部設計で使用
- ・複合設計／構造化設計は、内部設計で使用
- ・処理機能記述用のダイアグラムは主として、内部設計で使用

# 3. 上流工程でのDOA手法

## 3.1 DFD

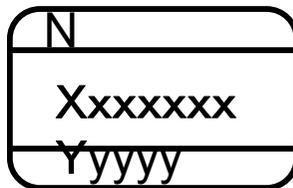
### ① DFDの表記法(4種類のみ)



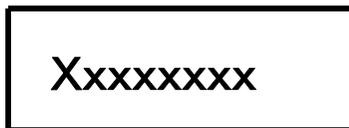
---データの発生源／行き先(External Entity)  
・組織、人、サブシステムなど



---データの流れ(Data Flow)  
・データフロー名を記入(物の流れ、処理は不可)

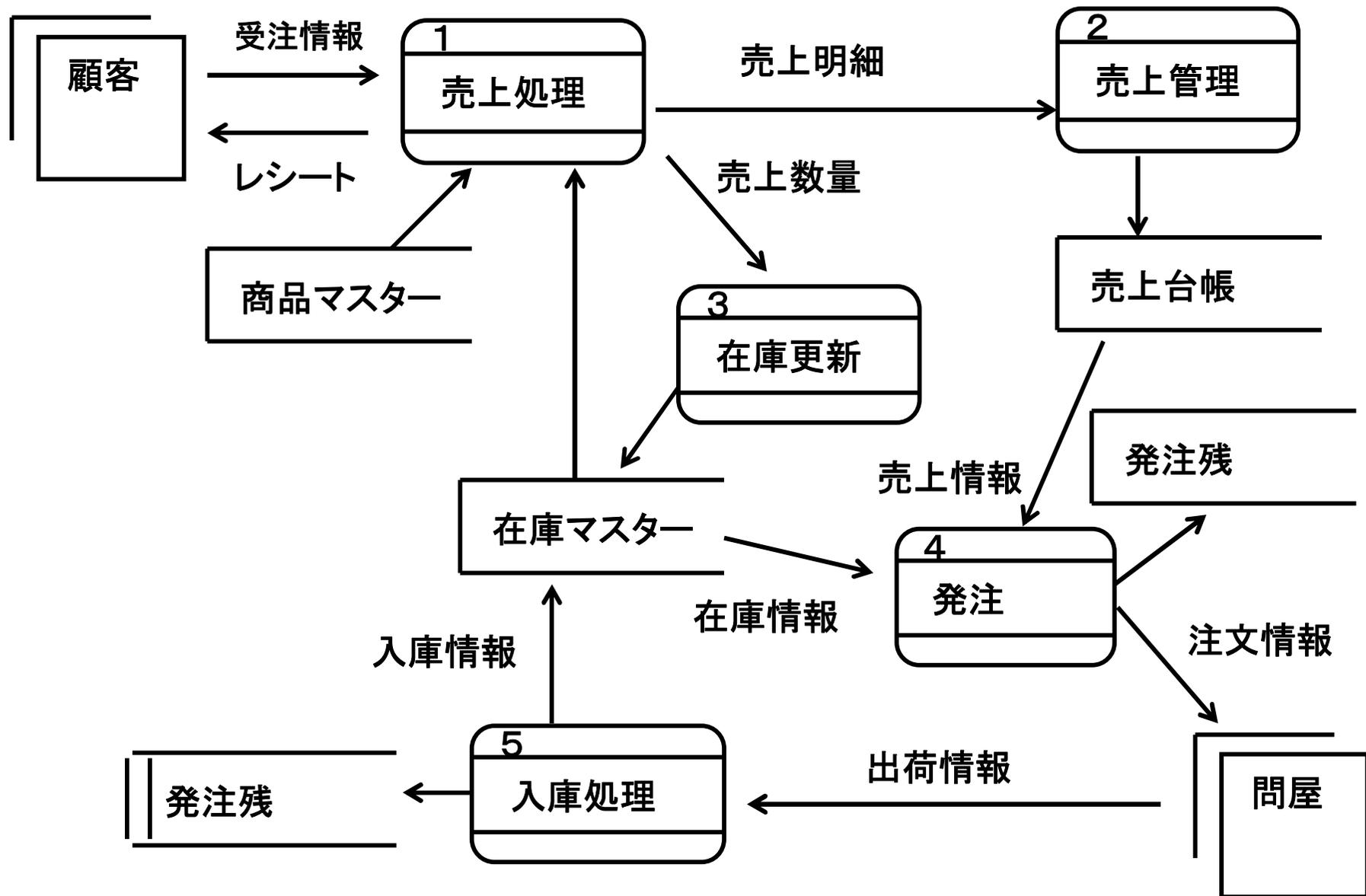


---データの処理、変換(Data Process)  
・処理名称  
・Nは識別番号、Yは担当部門名など



---データ・ストア(Data Store)  
・台帳、データベース、ファイルなどデータ蓄積

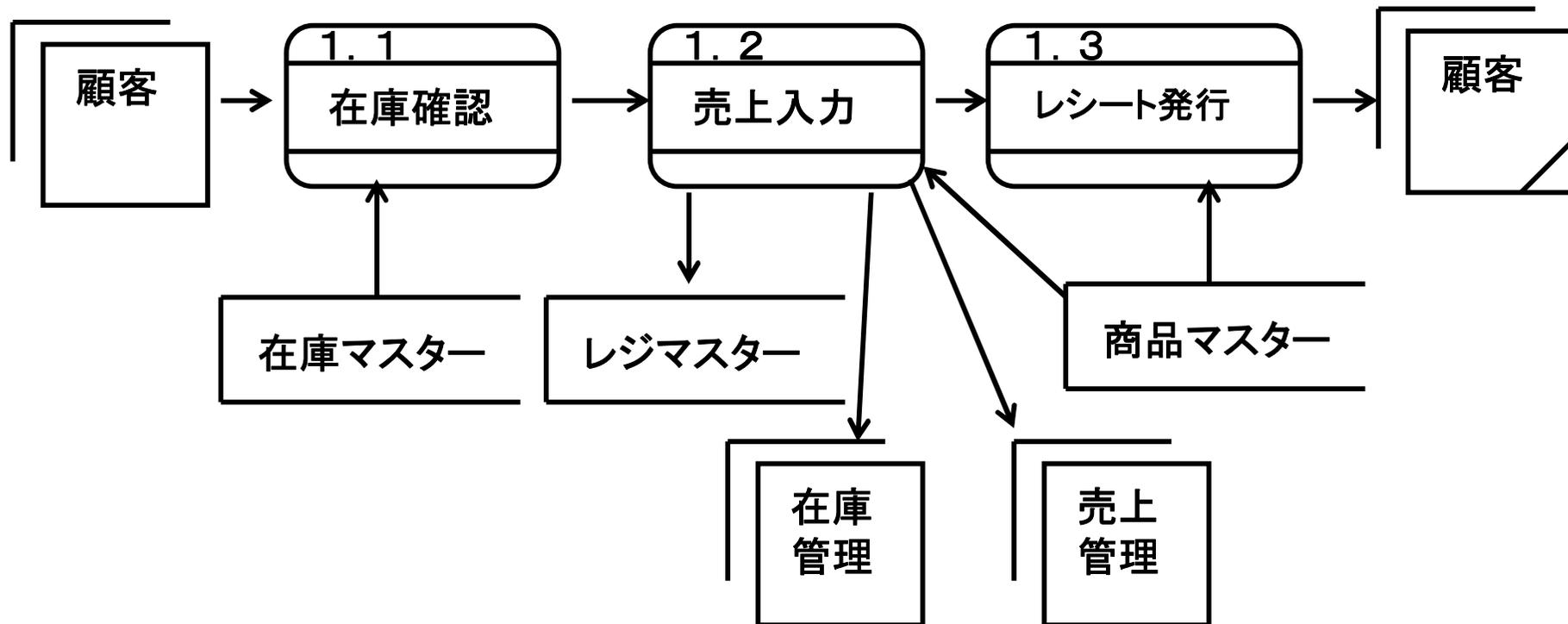
# 例示（小売業）



## ② DFDの作成方法

- 上位レベルから下方に展開する
  - 単純な対象は1レベル
  - 複雑な対象は、3レベル程度
- 第1レベルのDFDを作成
- 処理のボックスを下位レベルに展開する

例示 (1. 売上処理)



## ③記述書の作成

データフローの分析結果を明確にするために、  
3種類の記述書を作成する。(=構造化仕様という)

- ・ **データフロー記述** — — — — → **画面／帳票設計に使う**
  - データ項目、意味、桁数など
- ・ **データストア記述** — — — — → **データベース設計に使う**
  - データストアに含まれるデータ項目の記述
  - 先でデータベース仕様書として完成
- ・ **処理機能記述** — — — — → **プログラム設計に使う**
  - 処理ボックスの処理内容を記述
  - 通常、最下位レベルの処理ボックスが対象

## ④ 4つのDFDモデル

	現行システム	新システム
本質モデル (Essencial)	現・論理モデル	新・論理モデル
実現モデル (Implementation)	現・物理モデル	新・物理モデル

現状分析

システム設計

- ・物理モデルと論理モデルの差異は以下の通り
  - －実現手段などの要件の有無
  - －例えば、プロセスの実施者、実施部門、場所、サイクル、タイミング媒体などがある。
- ・現－論理モデルから、新－論理モデルへの変換は、発想の転換が必要で、システム要件、新規のデータ要件を加えて作成する。

## 3. 2 データの正規化

帳票、画面といった一組の「情報の製品」から適正な単位のデータの組として「データの部品」を作成する作業である。

- ・メリット: データの更新、追加、削除が弾力的になり、データ間の整合を保ちやすい。

## ①第1正規形

- ・これ以上分解できないデータ項目のみで構成。
- ・繰り返しのデータ項目を持たせない。

## ②第2正規形

- ・キーの一部に従属しているデータ項目を分離。
- ・キー以外の全てのデータ項目がキーに従属

## ③第3正規形

- ・キー以外のデータ項目間の従属関係を分離。

## 3.3 ER図

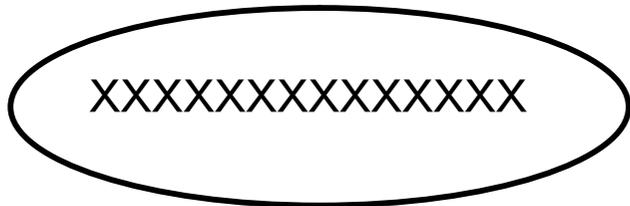
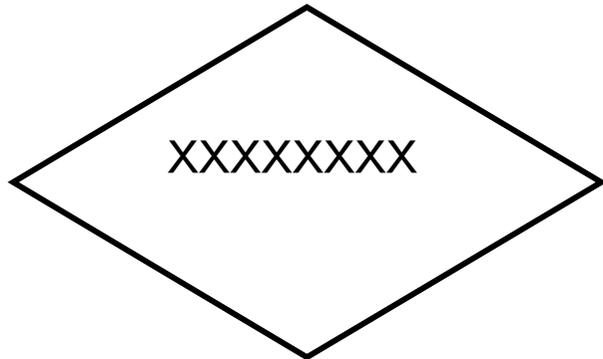
### ・特徴

- データ正規化の結果を元にして、
- データ本来の意味とデータ間の関連を把握し、
- データ固有の構造と特性を把握する。

### ・メリット

- 管理すべきデータを構造的に理解できる
- 独立性が高く冗長性が無いデータを保有
- 高品質で安定したシステムおよびDBの構築が可能となる。

# ① ER図の表記法



## ----エンティティ（実体）

- ・管理対象となるもの
- ・DBの候補
- ・厳密には1件ずつをエンティティといい、集合はエンティティセットという

↑  
通常、エンティティと略称

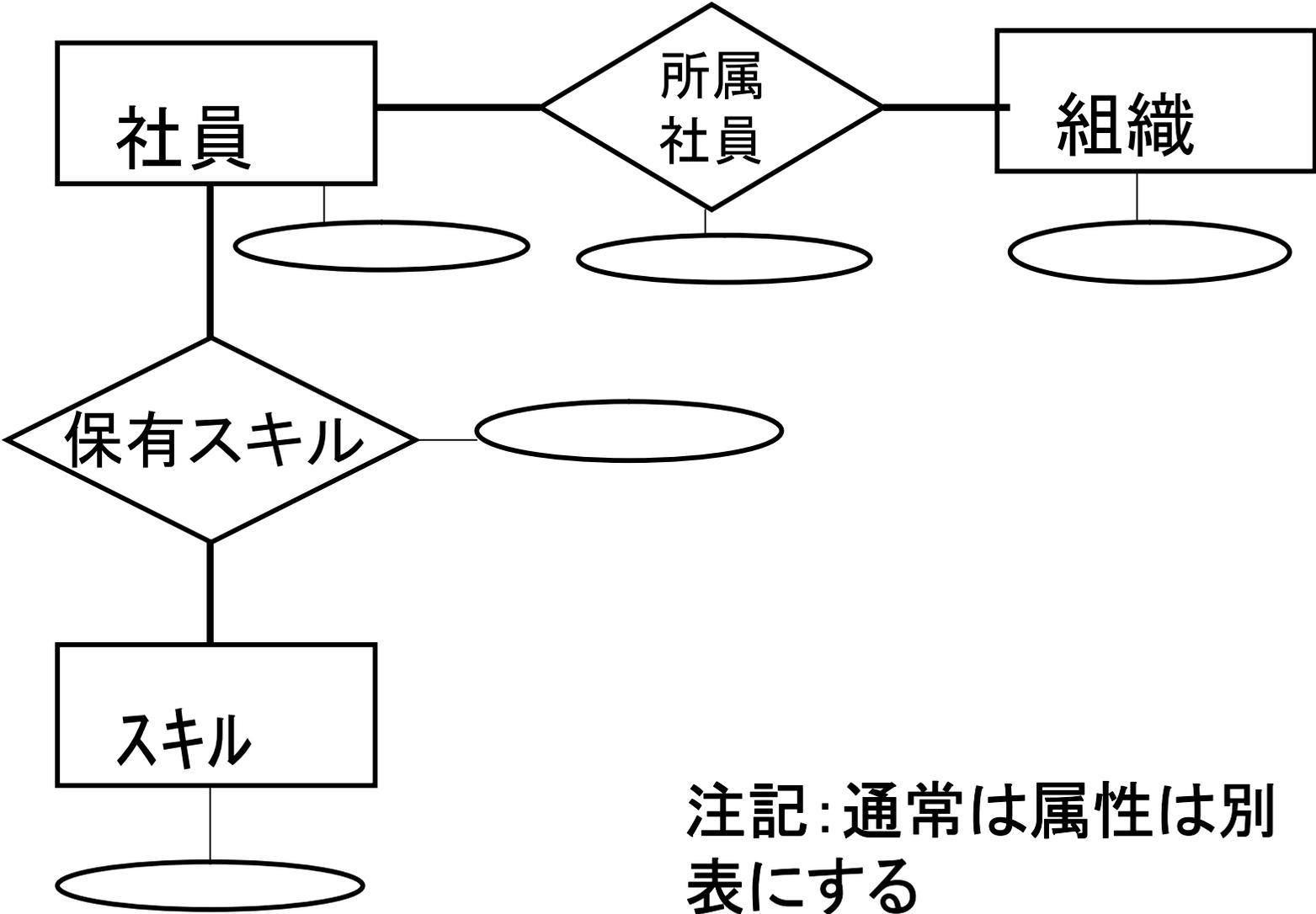
## ----関連

- ・エンティティ同士の間にある。
- ・これもDBの候補

## ----属性（データ項目）

- ・エンティティ、および関連に含まれるデータ項目である

例示



注記: 通常は属性は別表にする

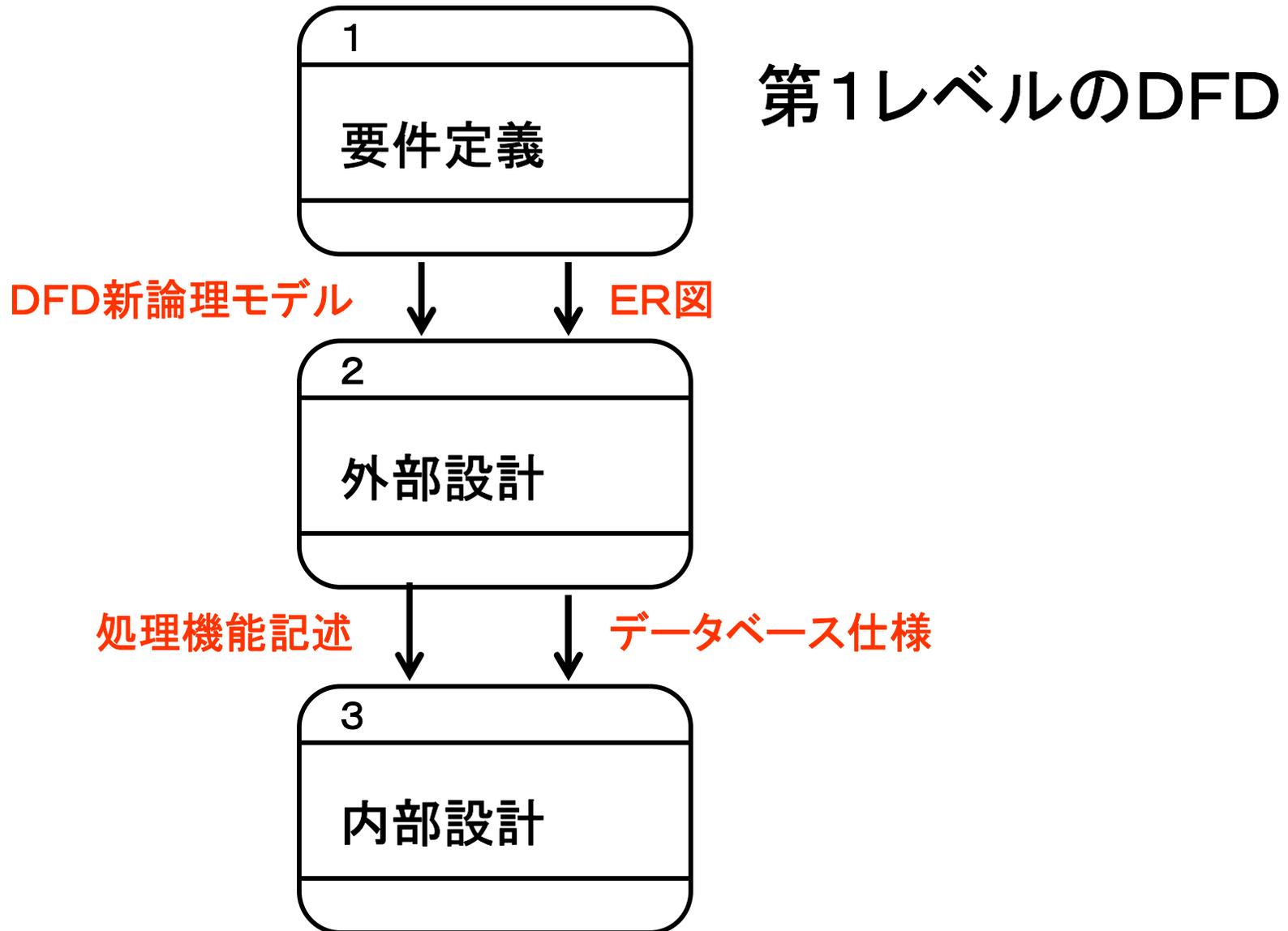
## ② ER図の作成手順

1. システム化対象の中で、データ正規化結果を元に、エンティティを決める。
  2. エンティティ同士の関連 (Relationship) を見出す。
  3. その関連の対応関係を見出す。  
(1:1      1:M      M:M)
  4. 属性 (データ項目) を漏れなくまとめる。
- これで、論理DB図が完成。

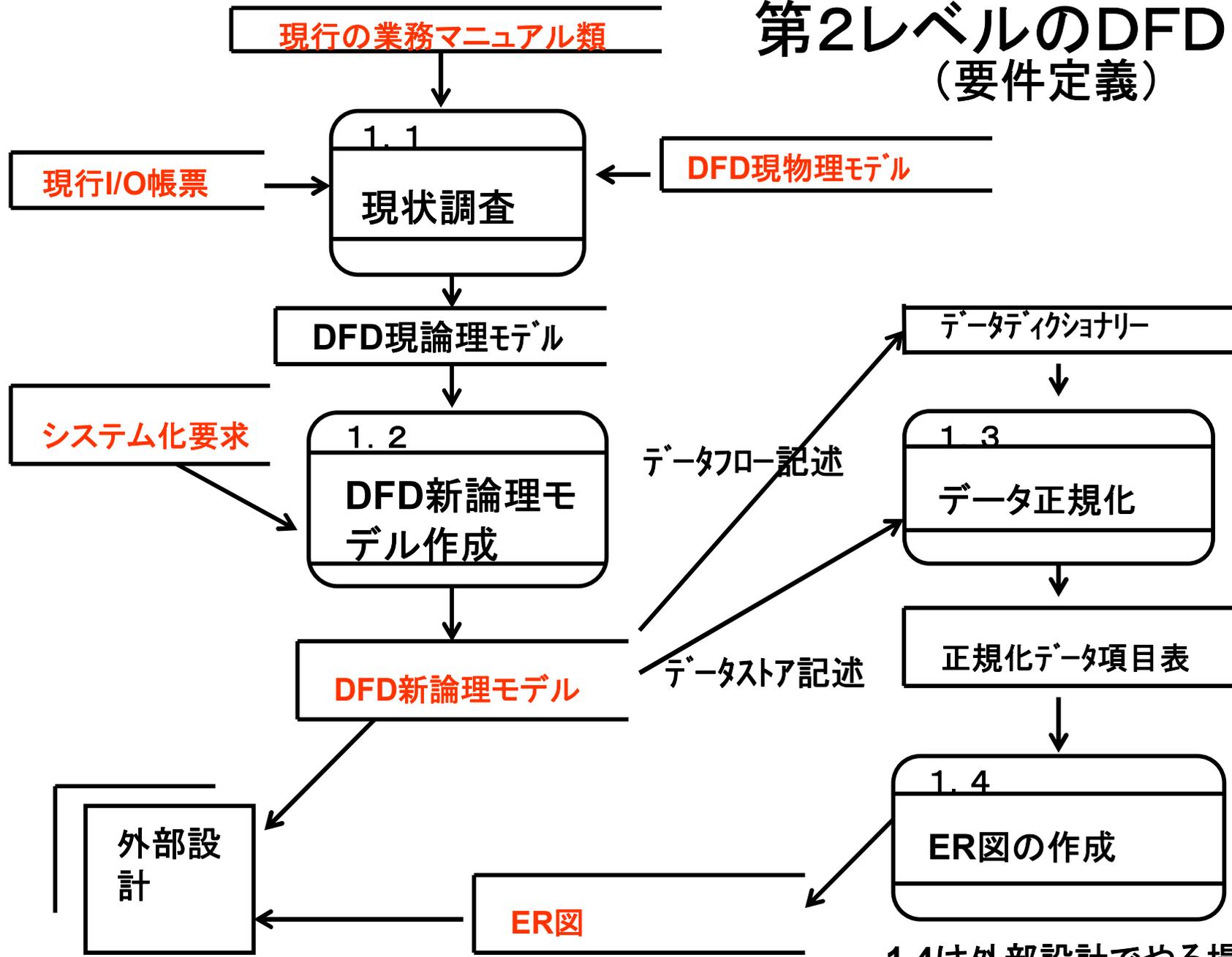
〔他に、汎化／特化、集約／分解の概念もある〕

# 4. 開発工程とDOA手法の関係

DOAを使用した実際の開発工程を、DFD形式で、以下に示す

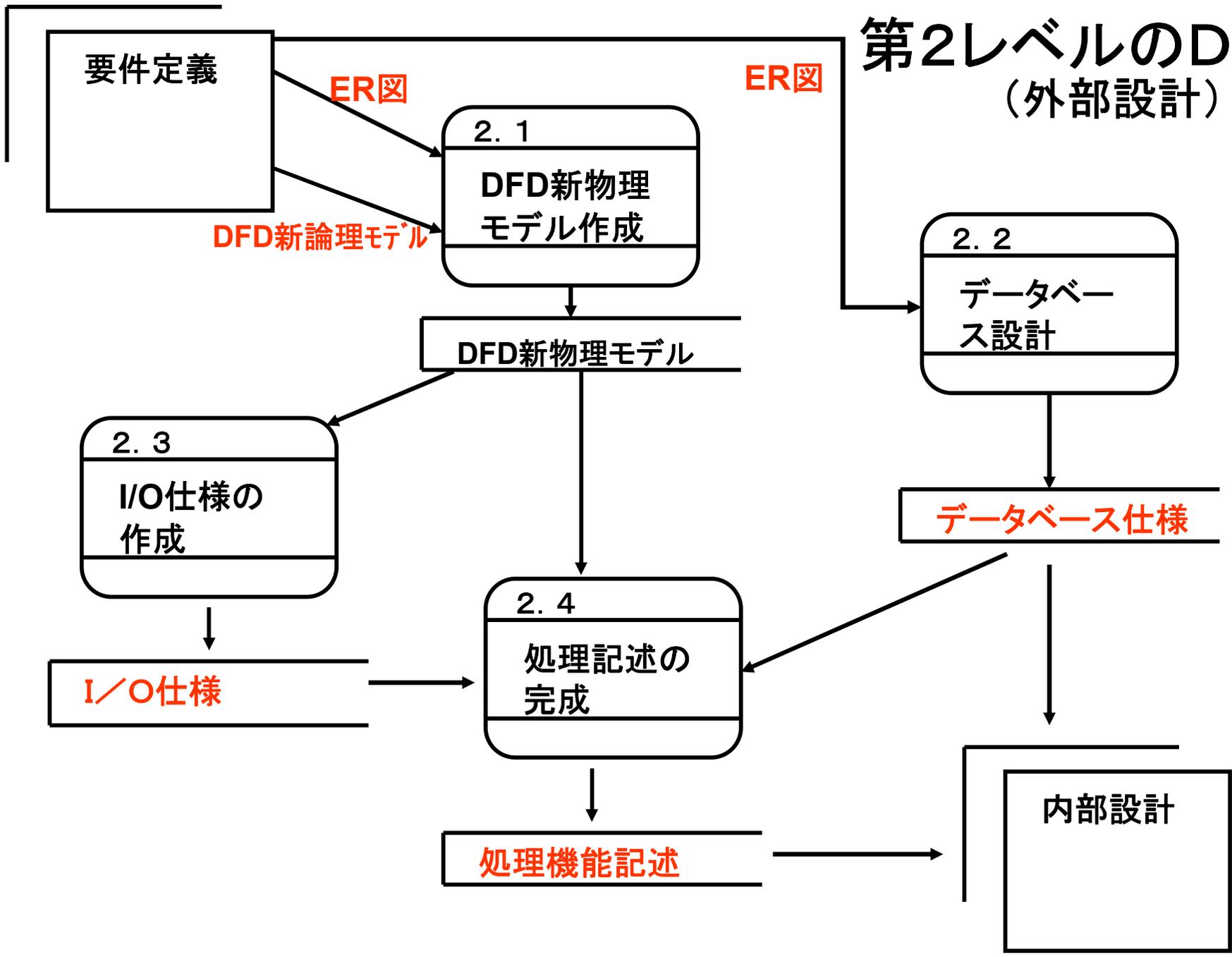


# 第2レベルのDFD (要件定義)

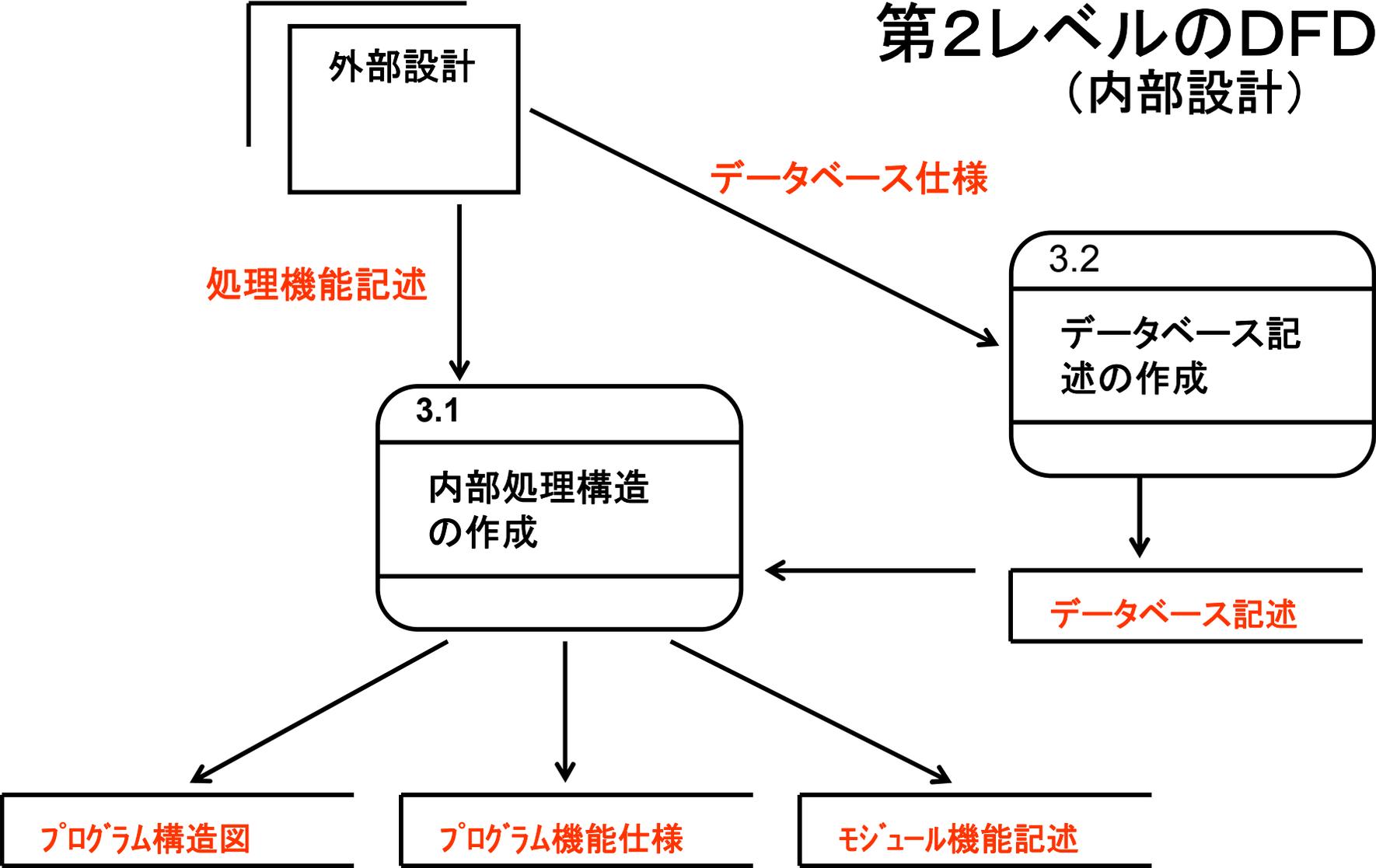


1.4は外部設計でやる場合あり。

# 第2レベルのDFD (外部設計)



# 第2レベルのDFD (内部設計)



# 5. DOA型開発の実際

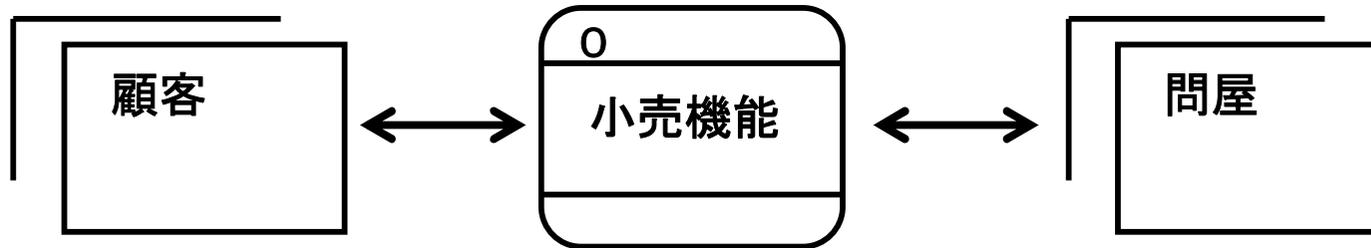
- ① 要件定義での**現状調査**が第1のキーポイント
  - ー対象業務の現状を良く知る
  - ー帳票／画面、業務マニュアルを収集・分析
  - ー各層にヒヤリング（トップ、ミドル、担当者）
  - ーヒヤリング対象者との信頼関係を作る
  - ー同業者、異業種から情報収集
  - ー業務のあるべき姿を勉強（本、雑誌・・・）
  - ー関係者に発表し、レビューを受ける

② 要件定義でのシステム化要求の分析が  
第2のキーポイント

- まず全ての要求を良く聴く(複数回)
- 要求を鵜呑みにせず、対案を用意する
- 個々の問題解決策を検討(問一原一解)
- 全体像を描きながら解決策を検討する
- 要求の実現優先度を聞きだし明確にする
- システムでの実現性を平行して検討する
- 実現には、データ整備が前提となる  
ことをユーザーに理解してもらう

### ③DFD作成時のポイント

- ・レベル0を作って、**対象の本来目的**を考えるクセをつける。

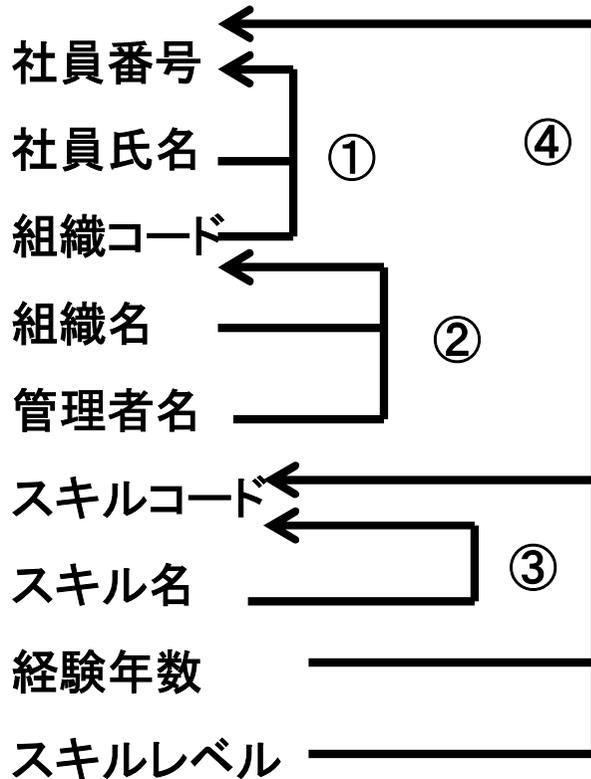


- ・1つの階層では、煩雑さを避け、機能ボックスの数は、5つ～7つ位にとどめる。
- ・下方展開のレベルは、以下を参照
  - データ項目が全て出てくるレベル
  - 帳票／画面が全て出てくるレベル
  - データ処理機能記述が出来るレベル

## ④ データ正規化の簡便な方法

- ・データ項目間の従属性に着目し、
- ・全てのデータ項目間の従属関係を洗出す
- ・**従属関係にあるデータ項目のグループを独立させる。**

社員スキル管理表＝



①社員＝社員番号＋社員氏名＋組織コード

②組織＝組織コード＋組織名＋管理者名

③スキル＝スキルコード＋スキル名

④保有スキル＝社員番号＋スキルコード＋  
経験年数＋スキルレベル

## ⑤ ER図の作成ポイント

- ポイントはエンティティを何にするかである。
- エンティティになりうるもの

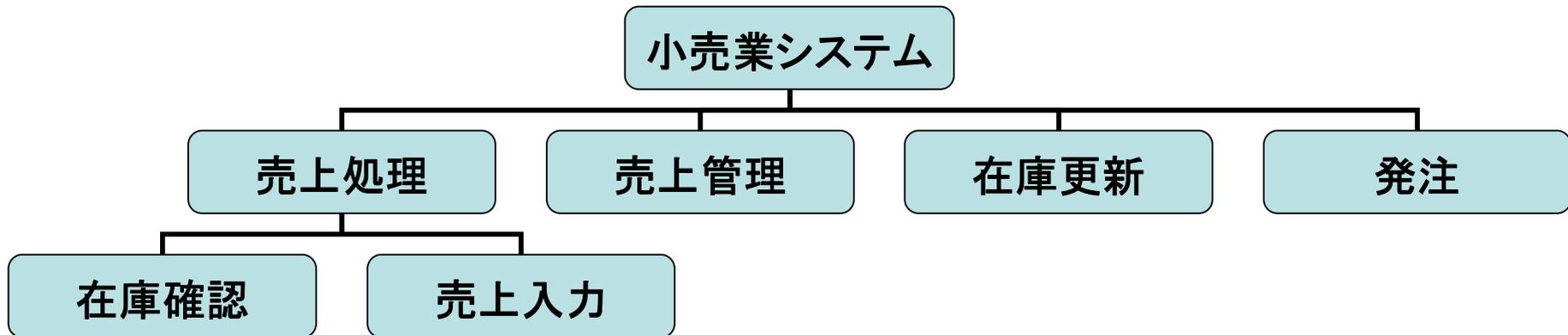
—人	: 社員、顧客、株主、学生、先生…
—組織	: 自社組織(XX課、XX係)、特約店…
—場所	: 本店、支店、販売地区、工場、倉庫、 店舗、配送拠点…
—物	: 商品、製品、部品、材料、設備、 機器、金、伝票、帳簿…
—事象	: 受注、発注、在庫、入金、売掛…
—概念	: 業務、スキル、目標、法規…

- 要するに、企業、組織にとって関心があり、**継続的に管理の対象になるもの**である。  
(=OOでいうクラスに近似)

# ⑦ 処理機能の漏れ発見方法

展開型の図表を作成

- ビジュアルで一瞥性があり、分りやすい
- **VTOC** と言う (Visual Table of Contents)



## 6. まとめとレポート課題

- 重要項目

- データ中心によるシステム開発の特徴
- DFDの概要
- データ正規化、ER図の概要

- レポート課題 (A4x1、2枚)

- ① DOA手法のメリットを述べなさい。
- ② 身近で簡単なことを、DFDで記述してみなさい。  
(買物、電話をかける、発表、レポート作成……)

期限      次回の授業開始時点

提出      レポート用紙またはメール