

---

---

# データの世界から見るシステム開発

## 全体最適システムの効率的な実現を目指して

日揮情報システム株式会社

---

### ■ 執筆者 Profile ■



村上 浩貴

2000年4月 日揮情報システム(株)入社  
システム業務担当  
2007年8月 基盤技術サービス部  
OSS・モデリング技術 R&D 業務担当

### ■ 論文要旨 ■

IT 技術の急速な高度化のなか、ビジネスからのシステムに対する要求は複雑さを増す一方である。複雑に連携するシステムの実現には、全体最適思想が必要不可欠だが、全体最適思想をシステム開発活動に一貫して維持することは困難である。日揮情報システムでは、情報誘導型設計を根底に置く DOA(Data Oriented Approach: データ中心アプローチ)によるシステム開発に多数チャレンジし、全体最適なシステムの効率的な実現に成功してきた。データに着目し、データモデルを中核として活動するこの手法は、システムの構想計画、設計、開発、レビューなど、システム開発全般に効果的に作用する。その手法の概要と実践方法を取りまとめる。

## ■ 論文目次 ■

<b>1. はじめに</b> .....	《 3》
1. 1  当社の概要	
1. 2  システム開発手法の変遷	
1. 3  情報システムエンジニアリング	
1. 4  当社の取り組み	
<b>2. システム開発における方法論の必要性</b> .....	《 5》
2. 1  システム開発の遂行における方法論の必要性	
2. 2  全体最適システムの実現における方法論の必要性	
2. 3  当社システム開発方法論“Just-Method”	
2. 4  “データ視点”である理由	
<b>3. DOA によるシステム開発の実践</b> .....	《 8》
3. 1  データ視点のモデルドリブンシステム開発	
3. 2  開発各フェーズでのデータモデルの位置付け	
3. 3  モデルマネジメントの実践	
3. 4  DOAでのシステム開発遂行体制の提案	
3. 5  DOAの実践によって享受される効果	
<b>4. おわりに</b> .....	《 13》

## ■ 図表一覧 ■

<b>図1</b> システム開発手法の変遷 .....	《 3》
<b>図2</b> 情報システムエンジニアリング .....	《 4》
<b>図3</b> “Just-Method”フレームワーク .....	《 6》
<b>図4</b> データモデルの変遷 .....	《 7》
<b>図5</b> データモデルレイアウト .....	《 11》
<b>図6</b> DOAプロジェクト遂行体制例 .....	《 11》

# 1. はじめに

## 1. 1 当社の概要

日揮情報システム株式会社（以下、当社という）は、日本を代表する総合エンジニアリング会社である日揮株式会社（以下、日揮という）の情報システム部門が分社独立して設立された企業である。

日揮は Oracle データベースの国内初のビジネスユーザーであり、リレーショナルデータベースを早期から活用している。そういった日揮の DNA を受け継ぐ当社も、データベースへのこだわりは強い。

当社は、日揮の基幹業務を支援する大規模システムの開発、保守運用業務に加え、日揮以外にもエンジニアリング業、建設業、製造業の顧客を中心に、Project Management System（以下 PMS という）、生産管理システムなどの大規模システムの構想計画から、開発、導入までの一貫したシステムインテグレーション業務を提供するシステムインテグレータである。

## 1. 2 システム開発手法の変遷

システムプロセス、システム機能に着目したシステム開発手法である Process Oriented Approach（以下 POA という）が現在も多く実践されているものとする。POA はシステム機能分析に注視しているが、大規模なシステム開発ではシステム全体の整合性維持が難しく、プロジェクト遂行上のリスクとなっている。

1970 年代以降、システム全体のデータ構造に着目した Data Oriented Approach（以下 DOA という）が発展してきた。DOA はシステムで取り扱われるデータに注視し、システムの全体整合をデータベースのレベルで担保すると言った思想に基づくアプローチである。

1980 年代以降、スモールトークに端を発し、近年 Java に代表されるオブジェクト指向言語の出現によって、再利用性の向上や、共有 IT 資産の活用容易性向上を根底に置く、オブジェクト指向分析やオブジェクト指向プログラミング（Object Oriented Analysis / Programming：以下総称して OOA という）が発展してきた。

現在、オブジェクト指向技術の発達に伴う、コンポーネント化技術の発達や、XML に代表される通信プロトコル技術の発達は目を見張るものがある。そういった技術の発達によって実現可能性の出てきた手法として、企業の内外を問わず使える情報及び IT 資産の有効活用を目指す、サービス指向アーキテクチャ（Service Oriented Architecture：以下 SOA という）が現在話題の中心になりつつある。

図 1 に示すとおり、これらの手法はそれぞれ 10 年程度の間隔をもって提起されている。

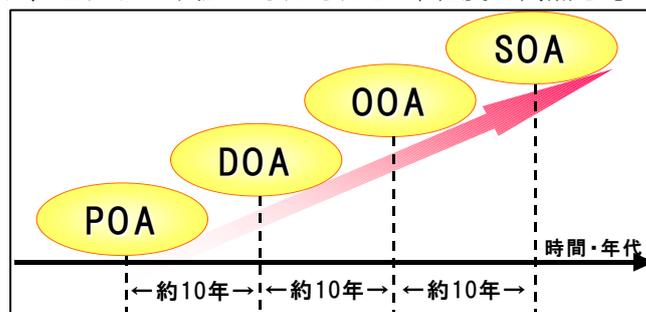


図 1 システム開発手法の変遷

### 1. 3 情報システムエンジニアリング

オブジェクト指向関連技術の進化に伴い、ビジネスユーザーからの情報システムへの期待、要求はより高度化され、複雑化してきている。Ajax に代表されるような操作性、ユーザビリティを向上する技術は、エンドユーザーに対して高い満足を提供することは明らかである。これらは“ソフトウェアエンジニアリング”の手法として、今や必須の技術となっている。

一方、インテグレートされた情報システムでは、そのシステムが取扱う情報を設計し、構築し、及び管理する“インフォメーションエンジニアリング”が不可欠である。インフォメーションエンジニアリングを実現する手法としては、未だ DOA を超えるものはないと考える。

図2に示すとおり、POA、OOA、SOA はソフトウェアエンジニアリングに注視した手法であり、DOA はインフォメーションエンジニアリングに注視した手法であると位置付けられる。これらの手法を相互補完の状態適切に連携させて活用することで、効率的な情報システムエンジニアリングが実現可能となる。

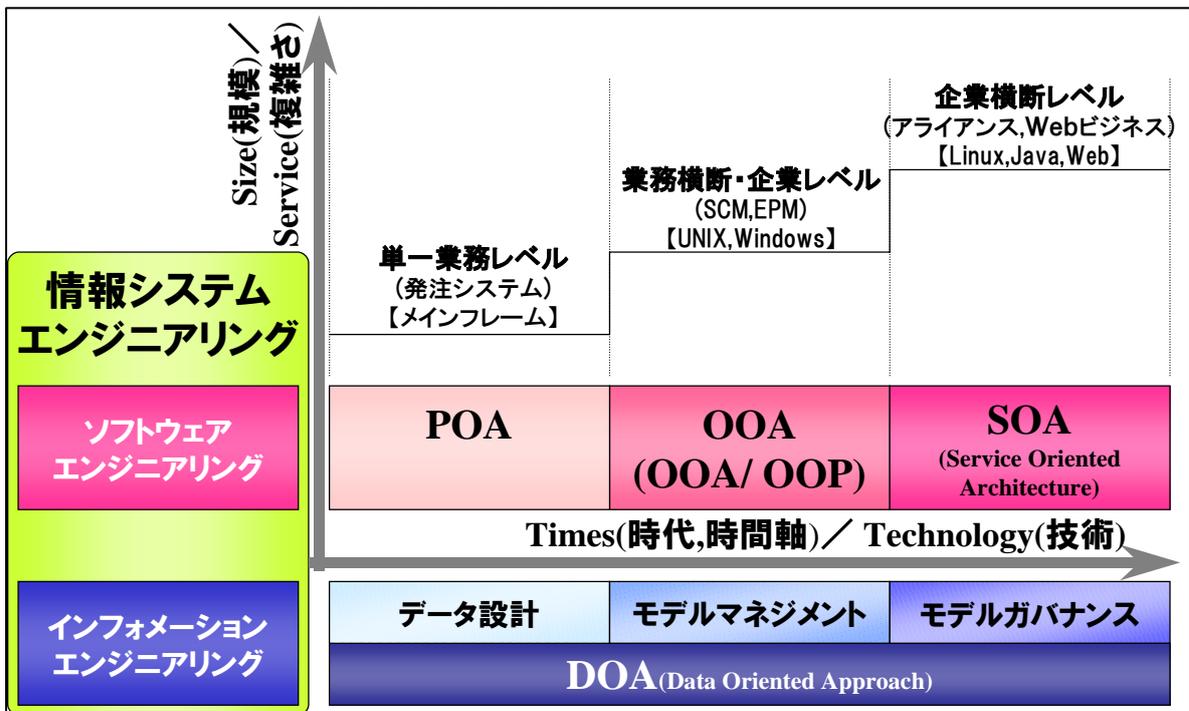


図2 情報システムエンジニアリング

### 1. 4 当社の取り組み

当社においては、DOA をベースとする開発方法論によるシステムインテグレーションを実践している。このアプローチの特徴は、対象システムが取り扱う“データ”に着目し、そのデータ構造を表現する“データモデル”をシステム開発の中核とすることで、システム化コンセプトをシステム開発プロジェクト全般に渡って維持する活動である。

当社においても、過去には「ただデータモデルを作成する」のみ、「データベースの設計図としてだけのデータモデル」と言った事例も見られた。このようなアプローチでは、各担当者単位でデータモデルが構築され、システム全体としての構造把握が薄いままにシ

システムが構築されることとなる。

個別に検討され、個々のシステム単位に関してはデータ中心でのアプローチによって確かな構造を持ったシステムが完成したとしても、全体のバランスを失ったシステムが、業務を遂行するユーザーにとって、いかに使い勝手の悪いものであるか、ひいては、使われないシステム、動かないシステムとなる可能性がいかに高いかは、想像に難くない。

特に、当社で導入事例の多い PMS など、幅広い業務の円滑な遂行を支援する類のシステムに関しては、全体最適が必要不可欠であり、システム全体をバランス良く把握することが最大の課題となっていた。

そこで、当該開発方法論の適用することにより、データモデルを用いたシステムコンセプト維持の方法や、データ中心、データモデル中心の開発ステップ、さらにはシステム開発遂行の体制における役割と権限をプロジェクトメンバーで共通認識し、開発プロジェクトに臨むことで、全体最適でユーザーに満足されるシステムを実現してきた。

以下にこの方法論の概要と、データ視点でのシステム開発実践へ向けた提言を紹介する。

## **2. システム開発における方法論の必要性**

### **2. 1 システム開発の遂行における方法論の必要性**

システム開発技術は、時代とともに進化する反面、複雑化しており、システム開発プロジェクトでは複雑に関与しあう様々な技術を計画、管理する必要性に迫られている。

人を管理し、かつ技術を管理する必要があるシステム開発プロジェクトを成功させるためには、体系的に整理された方法論は不可欠である。

体系立った方法論によって「いつ、何を、どこまでやるべきなのか？」を明確に把握し、確かな開発計画の下でのシステム開発が、Win-Win の関係でのシステム実現の第一歩である。

### **2. 2 全体最適システムの実現における方法論の必要性**

一方、システムの品質に目を向けても方法論の必要性が見えてくる。

“システム”とは、個々の機能や要素が有機的、体系的に結合された全体を指す。今日、情報システムがカバーすべき業務範囲は広がり、その機能や要素は複雑に連携している。更に、ソフトウェア構築技術の進化によってシステムが“実現できること”は拡大しており、個々機能の複雑性はより高いものとなる。

これらの機能を、効率的、効果的にインテグレートするためには、それに見合った方法論が必要となる。特に大規模システムの実現には、システムに期待されるトップダウン的なコンセプトと、ボトムアップ的な利便性の双方からのアプローチが必要になる。

## 2. 3 当社システム開発方法論“Just-Method”

当社では、上記二つの要件を満足するために、DOA をベースとする開発方法論“Just-Method”を開発した。Just-Method は、図3に示すとおり横軸にシステム開発のフェーズ、縦軸に EA (Enterprise Architecture) 視点での開発対象を展開したフレームワークをベースとする。

当該フレームワークにおいては、システム開発各フェーズでの各開発対象領域に関する主要成果物、及び主要アクションが表現される。

### (1) システム開発フェーズ (横軸)

- (1.1) 企画・計画 : システム化企画・概念設計
- (1.2) 設計 : 基本設計・詳細設計
- (1.3) 構築 : プログラミング・単体テスト
- (1.4) テスト・ロールアップ : 統合テスト・システム移行

### (2) EA 視点の開発対象 (縦軸)

- (2.1) 業務体系 : 業務・ビジネスプロセス
- (2.2) データ体系 : データ・データベース
- (2.3) アプリケーション体系 : ソフトウェア・アプリケーション
- (2.4) 技術体系 : 情報システム基盤、ミドルウェア

開発各フェーズに関する更に詳細なアクティビティは、全社標準として構築された FWBS (Functional Work Breakdown Structure) で規定され、システム開発プロジェクトの計画、管理に活用される。



図3 “Just-Method”フレームワーク

一方、品質面、特に全体最適性の維持に対しては、データ体系領域をシステム開発の主軸とすることで担保する。図4に示すとおり、データ体系領域の中心成果物であるデータモデルを構築、維持、管理する形で、システム企画からロールアップまで一貫したシステムコンセプトを担保するものである。

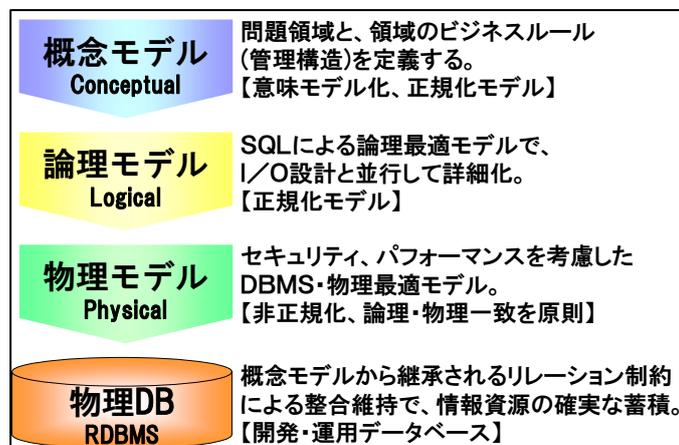


図4 データモデルの変遷

”データ視点”で遂行される DOA によるシステム設計は、情報誘導型設計コンセプトを根拠としている。情報誘導型設計とは、情報要求と、情報要求を満たすデータ、及びそのデータ構造に注視する設計思想である。

データベースに格納されているデータはそのままでは価値がなく、情報として要求され、ビジネスシーンで活用されることによって価値が向上すると言った思想に由来する。

元来、情報システムとは、データを情報としてユーザーに開示し、ユーザーの業務や意志決定の支援を主な目的とするものである。情報誘導型設計を根拠とする DOA においては、情報要求を中核に、データ構造を構築するアプローチが重要となる。

## 2. 4 “データ視点”である理由

“ビジネスプロセス”、“アプリケーション機能”、“データ”を見比べた場合、データはこれらの中で極めて静的要素であるといえる。

“ビジネスプロセス”は、昨今の SOA の流行からも見て取れるとおり、変化の柔軟性と迅速性が求められている。“アプリケーション機能”はエンドユーザーの多様な要求を受け、日々進化し、改善される。対して“データ”は、ビジネス活動のアウトプットであり、インプットであるため、ビジネスモデルの大きな変更がない限り安定的に存在する要素である。

また、情報システム、インフラ基盤など、企業には IT 資産が多数存在するが、中でもデータは最大の資産である。データは、蓄積することに意味があるのではなく、蓄積されたデータを有効活用することに意味がある。そのため、活用可能なデータを蓄積するための方法と構造を確保することは、情報システム構築に対して最大のミッションとなるのである。

なお、データ構造の構築にあたっては、想定される情報要求を加味することが肝要となる。対象業務全般のデータ構造を明らかにすることで、システムの全容を明確に把握し、情報要求を加味することでシステム機能の難易度を把握する。これによって、システム全体としての難易度、複雑さが明確となり、誤った IT 投資やシステム開発プロジェクトの混乱予防に大きな効果が期待できる。

### **3. DOA によるシステム開発の実践**

#### **3. 1 データ視点のモデルドリブンシステム開発**

DOA によるシステム開発においては、当該システムの取扱うデータ構造を可視化し、顧客を含めた開発プロジェクトのステークホルダー間で共有することが重要となる。データ構造の可視化にはデータモデル (ER 図) を用いる。DOA はデータ視点のモデル駆動型アプローチ (Model Driven Approach : MDA) である。

システム開発に向けた計画段階では、ユーザー要求は曖昧な場合が多く、曖昧さを残したままでの要件定義は、システム開発終盤に発覚するような大きな問題を内在させやすくする。

初期の段階で曖昧なユーザー要求を情報要求として捉え、データモデルで構造化し、できる限りの曖昧さを排除することが、開発プロジェクトを成功へと導くのである。

先述のとおり、データモデルの構築にあたっては、トップダウン的なビジネス要求、ビジネスルールを維持する“高い視点”と、ボトムアップ的な操作性、利便性要求をバランスよく設計する“広い視野”が必要となる。

特にトップダウン的に設定されるビジネスルールに関しては、システム開発の計画段階から、テスト、ロールアップに至るまで、全般に渡って確実に維持されなければならない。DOA においては、開発のフェーズにあわせた 3 タイプのデータモデルで、ビジネスルールの維持に取り組むこととなる。

#### **3. 2 開発各フェーズでのデータモデルの位置付け**

Just-Method フレームワークにおいては、以下のとおりデータモデルが構築される。

- |            |            |
|------------|------------|
| (1) 計画フェーズ | : 概念データモデル |
| (2) 設計フェーズ | : 論理データモデル |
| (3) 構築フェーズ | : 物理データモデル |

各データモデルは、概念から論理、論理から物理と、その構造を維持、拡張させながら構築されていく。以下にそれぞれのデータモデルの位置付け及び果たすべき役割を紹介する。

### 3. 2. 1 概念データモデル

#### (1)位置付け

問題領域とその領域のビジネスルール（管理構造）を定義する

#### (2)見られる特徴

- (2.1) 分析対象範囲内外の境界線が明確
- (2.2) 下記役割を果たせる程度の粒度
- (2.3) サブタイプによる情報種類の識別と種類別の制約が表現可能
- (2.4) 理解容易性を考慮した日本語でのオブジェクト命名

#### (3)果たすべき役割

- (3.1) 対象業務全体の情報構造（ビジネスルール）を表現
- (3.2) システム化対象範囲内外の情報連携を明確化
- (3.3) 収集必須な重要データ項目の網羅性を向上

### 3. 2. 2 論理データモデル

#### (1)位置付け

SQLによる論理最適モデルで、I/O設計と並行して詳細化する

#### (2)見られる特徴

- (2.1) システム化対象範囲内外の境界線が明確
- (2.2) SQL論理最適構造
- (2.3) 管理容易性及び物理データベースを意識したデータモデルの分割
- (2.4) 理解容易性を考慮した日本語でのオブジェクト命名

#### (3)果たすべき役割

- (3.1) 概念データモデルから継承される情報構造（ビジネスルール）の維持
- (3.2) システムの取扱うすべてのエンティティ、属性を表現
- (3.3) 正規化の維持、冗長化の防止
- (3.4) 論理名称の統一による物理名称（テーブル名、カラム名）統一性の維持

### 3. 2. 3 物理データモデル

#### (1)位置付け

セキュリティ、パフォーマンスを考慮したデータベース物理最適モデル

#### (2)見られる特徴

- (2.1) 論理データモデルとの構造一致が原則
- (2.2) 物理データベースの図面
- (2.3) 対象データベースシステムに対応したデータタイプによる表現
- (2.4) データベースの安定性を考慮した英語略称でのオブジェクト命名

#### (3)果たすべき役割

- (3.1) 論理データモデルから継承される情報構造（ビジネスルール）をデータベースへ実装
- (3.2) 物理データベースの構築に必要な設定情報の管理
- (3.3) 迅速確実なデータベース構築、メンテナンスの支援（ツールの活用）

### 3.3 モデルマネジメントの実践

先述のとおり、データモデルはシステム開発全般を通して、ステークホルダー間で共有され、維持、メンテナンスされていく。特に設計フェーズ、構築フェーズに関しては、当該開発に関与するメンバが増加し、データモデルのメンテナンス頻度も高くなる。

データモデルメンテナンスの遅延はプロジェクトメンバのデータモデル離れを招き、システム開発を危険な方向へ導きかねない。しかしながら、迅速性のために乱雑に描かれたデータモデルはプロジェクトメンバのデータモデルを見るモチベーションを削ぎ、同様にモデル離れを引き起こす。いかに効率的に、かつきめ細かにデータモデルを管理し、維持していくかが大きな課題となるのである。

メンテナンス頻度の多さに加え、データモデルにはエンティティ、属性に始まり、各オブジェクトの定義情報やデータタイプ、リレーション、各種制約といった膨大な情報が詰め込まれることになる。これらを効率的に管理するためにはツールの活用が不可避である。

データモデル管理ツールとしては、ERwin や ER/Studio がデファクトスタンダードになっている。これらツールは、データベース構築のための Data Definition Language (以下 DDL という) の自動生成機能や、データベースとデータモデル、データモデルと別のデータモデルの間での変更点を精査しマージする機能などを有する。

また、特徴的な機能として、論理名称と物理名称の変換機能がある。日本語、もしくはフルスペルの英語で命名された論理データモデル上のエンティティや属性の名称を、用語辞書に登録されたマッピング情報を用いることで、略語の英語名称に自動的に変換することが可能となる。

用語の統一は、プロジェクトマネジメント上の重要な管理課題の一つである。一つのシステム、データモデルの中で、同じ事象を異なる言葉で表現したり、異なる事象を似通った紛らわしい言葉で表現することはシステム開発に対して、些末な問題でありながら多くの勘違いや混乱を生じさせる。

特に、顧客業務を完全に把握できていないような初期段階におけるカタコトの会話の中で発生した小さな勘違いが、システム構築フェーズで様々な影響を与えることは、知られた課題の一つである。用語集を作成し、用語集から逸脱しないようにオブジェクトを命名することが肝要である。

しかしながら、用語集を当初から完全に整理することは困難であり、システム開発の進む中で用語集の追加、見直しは避けられない。

ただし、無秩序な用語集の編集は、無駄な用語や冗長な用語の増加につながるため、用語集の追加、見直しに際しては、「同一の事象を表す異なる用語は登録されていないか?」「同じ用語が既に異なる意味合いで登録されていないか?」といった視点で、監視する必要がある。

ツールの活用により効率化を図ると同時に、データモデルのきめ細かな管理が必要となる。特にエンティティやリレーションのレイアウトはデータモデルの見やすさを決める重要な要素となる。

図5に示すように、エンティティはデータ発生の時系列やデータの粒度を意識した左上から右下へ向かうレイアウトが効果的である。また、リレーションはパイプライン（リレーションの束）を作り、データモデル内をちらかさないことが、見やすいデータモデルを構築するコツである。

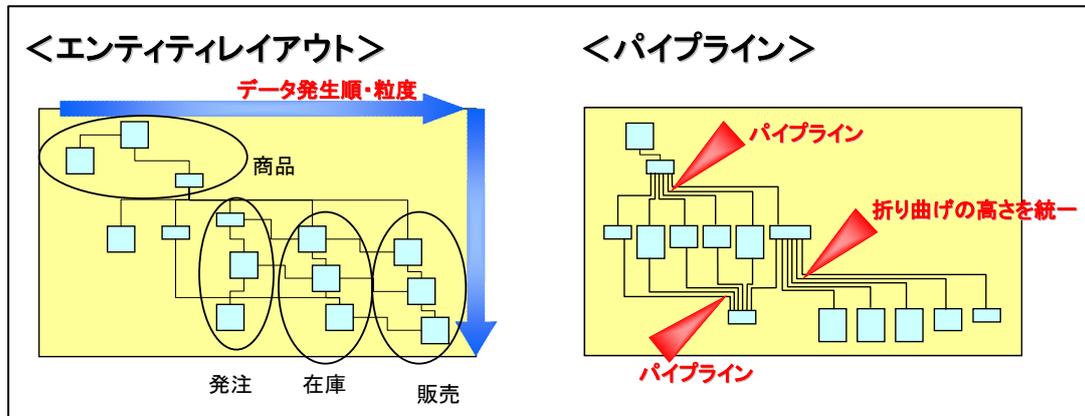


図5 データモデルレイアウト

### 3. 5 DOAでのシステム開発遂行体制の提案

データモデルの一貫性を実現するためには、プロジェクト遂行体制への留意が不可欠である。経験的に、図6に示す体制が有効であると考えられる。Data Manager（以下DMという）、Data Administrator（以下DAという）と呼ばれる役職がプロジェクトチームにおいて、相当の権限を有するところに特徴がある。

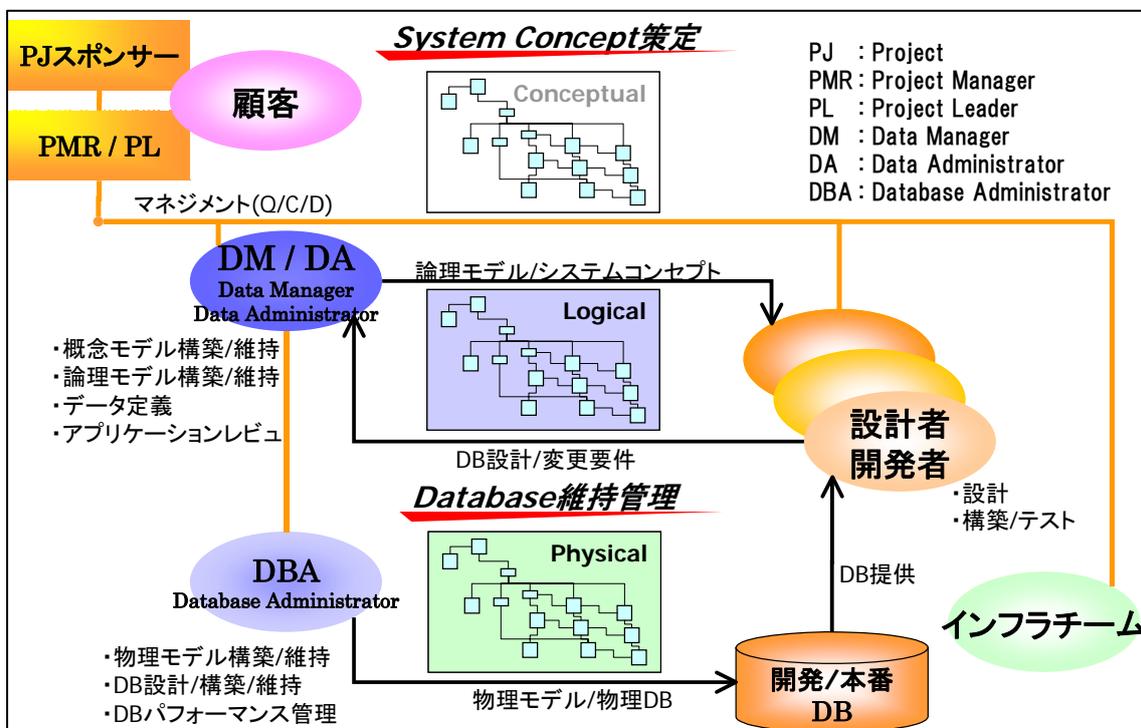


図6 DOAプロジェクト遂行体制例

DM、DA は、データモデルを通してプロジェクト全体のガバナンスの一躍を担う。実際のプロジェクトでは、DM、DA が一手にデータモデルをメンテナンスする方式や、個別サブシステムごとにデータモデルを構築し、DM、DA がレビュー、統合するなど、様々な方式が考えられる。いずれの場合も DM、DA は顧客からの要求、システムコンセプトを常に意識し、設計者の検討内容の妥当性や全体最適性を維持する役割を担う。

また、開発者に対して遅延なくデータベースを提供することは、開発の停滞を防ぐという意味で必要最低限の条件となる。DM、DA と Database Administrator (DBA) は密接に連携し、データモデルの変更管理と、データモデルの変更に対するデータベースへの変更適用状況の管理を緻密に行う必要がある。

データベースの変更はアプリケーションの設計、及び構築に直接的に影響し、プロジェクトコストにインパクトを与えることになる。経験ある DM、DA は、データモデルの変更管理の中で、その変更がシステム全体に対してどれほどの影響を与えるものかを把握し、プロジェクトマネージャーへの警鐘を鳴らす役割をも担うべき位置付けとなる。

### **3. 4 DOA の実践によって享受される効果**

DOA は、システムの基本要素である“データ”のレベルで全体最適を維持することで、システムの全体最適性を確保する。また、データモデルという目に見える形でシステム化コンセプト、ビジネスルールをデザインし、それをシステム開発全般に浸透させることができ、結果的に「使える」「使ってもらえる」システムの実現に大きく貢献する。

情報誘導型設計においては、データモデルの構築にともなってシステム全体像の把握を容易にする。“情報（システム）＝ 管理されたデータ（データベース） × 処理（アプリケーション）”といった考え方は、システムの広さと複雑さを洗い出すために非常に効率的な手法であり、的確なコスト見積りを実現することができる。

また、プロジェクトチームとしてデータモデルを共有することで、プロジェクトにおける変化や変更をつぶさに捉え、対策、代替策の検討に早期に着手することができる。これは、プロジェクトの失敗を防止することに貢献する。

的確なコストでシステム実現することは、顧客とシステムインテグレータ双方にとって Win-Win となる必須条件であり、DOA はその実現を強力に支援する手法である。

## 4. おわりに

DOA は過去の手法と言う声も多い。しかし、データは企業の重要な資産であり、実際に米国では MDM (Master Data Management) や CDI (Customer Data Integration) など、データ統合やデータ管理が重要なテーマとなっている。一方、新しいアプローチとして期待される SOA でも、改めてデータソース管理の課題が注目されている。

今後、データ統合、Web サービスの活用と言った潮流は強まるものと考えられる。そのためには、再利用性を意識した強力なモデルマネジメント手法の確立が重要であり、更に、SOA を代表とするソフトウェアエンジニアリング技術と DOA との効果的連携の確立が重要となる。

当社においても、顧客資産価値最大化を目指し、再利用性向上に向けたデータモデルリポジトリツールを用いたデータモデルライフサイクル管理や、先進技術を効果的に活用するための OOA、SOA と DOA の連携手法確立に取り組んでおり、チャンスがあれば、実践状況を再度報告させていただきたいと考えている。

以上