

構造化分析手法とデータベース設計技法の統合方式の提案

6J-6

黒木 宏明 山本 修一郎  
NTTソフトウェア研究所

はじめに

大規模ソフトウェア開発では業務設計とデータベース設計が別グループで独立して開発を進めることが多い。さらに、業務設計とデータベース設計では開発技法が異なっており、互いに仕様を検証することが困難であった。本稿では構造化分析手法とデータベース設計技法それぞれで作成される設計情報の対応関係を整理し、それに基づく分析作業モデルを提案するとともに相互の変換方法について明確にする。

構造化分析とデータベース設計

(1)構造化分析手法(SA手法)

構造化分析手法<sup>1)</sup>に基づいて、機能、データ項目の階層的構造、データ項目の論理的关系などを記述する。一般に構造化分析を用いた業務設計は以下の手順で行う。

- 1)機能分析 業務を構成する機能およびデータの全体像を整理する。主にデータフロー図(DFD)、実体関連図(ERD)を使用する。
- 2)機能設計 各機能の概要を明確にするとともにデータの構造および属性を明確にする。主にデータフロー図、実体関連図、データ構造図(DSD)を用いる。
- 3)詳細設計 業務をシステム化するために必要なモジュールおよび処理手順を明確にする。主にモジュール構造図(SC)を用いる。

(2)データベース設計手法(DB設計手法)

データベース設計手法<sup>2)</sup>に基づいてシステムで使用するデータベースの設計を以下の手順で行う。

- 1)概念設計 業務で必要とするテーブル全体像を整理し、テーブルの正規化を行う。主に実体関連図、テーブルの列情報(列名、型名、精度、説明等)を記述するテーブル定義票、システムで実現する業務の一覧を表わす業務分析票を用いる。
- 2)論理設計 システムの性能条件を考慮してテーブルの構造を再整理する。テーブルのアクセス回数およびアクセス順序を記述する業務テーブル関係票を用いる。

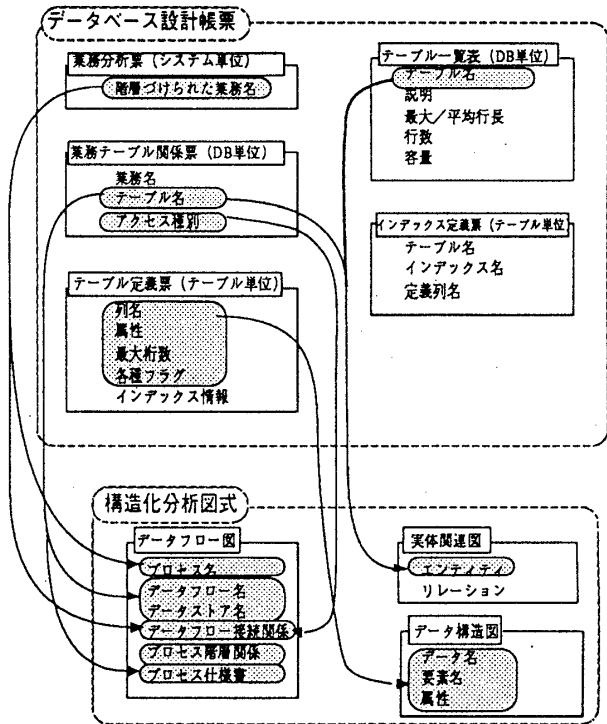


図1 データベース設計帳票と構造化図式の設計情報の対応関係

3)物理設計 具体的なデータの格納方法を決定する。

(3)両手法の関係

両手法で用いられるドキュメントの構成要素を整理すると図1のような対応関係がある。

- 1)SA手法もデータベースの概念設計が行え、DB設計手法と重複がある。
- 2)DB設計手法でも性能診断を行うためには業務分析結果を必要としている。この業務分析結果はSA手法で分析される。

このため、両手法の統合によりソフト開発作業を効率化することができる。

統合作業モデル

システム開発の特徴を以下に示す。

- 1)大規模システムでは扱うDBおよび処理も複雑である。
- 2)それぞれ開発手法および設計ノウハウが異なる。このため、業務設計、DB設計はそれぞれの設計者が

独立して行う。

しかし、両手法間の設計情報には関連があるため、無関係な開発は行えない。従来の手法では、開発作業中での整合性の検証手順が不明確であったため、手戻りが大きく作業効率が悪かった。また、同一設計情報を重複して分析/投入することになり、作業効率や設計品質の低下の原因となっていた。

本稿では以上の問題点を解決するため、統合化された業務設計手順とデータベース設計手順を示す。

**step1**

実現するシステムの業務を分析し、機能、データフローを抽出し、DFDを作成する。(業務設計者)

帳票等によりDBの概念モデルを作成しERDを作成する。(DB設計者)

テーブル間の関係を整理し、データフロー図のテーブルアクセス関係とERDのテーブル関係と矛盾がないか検証する。(両者)

**step2**

テーブル間の関係に基づいてDFDをリファインしプロセス仕様を詳細化する。(業務設計者)

システムの性能条件に基づいてテーブル構造を整理する。DBの性能診断を行うために業務分析票および業務テーブル関係票を作成する。(DB設計者)

プロセス仕様書から列アクセス関係を抽出し業務テーブル関係票から得られる列アクセス関係と矛盾がないか検証する。(両者)

**step3**

インデックス設計を行い効率的なテーブル構成を行う。(DB設計者)

テーブル構成の見直しに基づいてデータストアおよびデータフローを修正する。(業務設計者)

対応関係にあるデータフロー図のプロセス仕様と列アクセス一覧との間で矛盾がないかチェックする。

(両者)

**統合機能**

社内ツールとして構造化分析手法を支援するツールSoft DA[3]、データベース設計手法を支援するツールDBpromptがある。それぞれ独立のツールとして開発を進めていたが、開発作業効率を改善するため、統合作業モデルにしたがって統合するための機能を開発した。これにより構造化分析図式とデータベース分析帳票を構成する設計情報を相互に変換することにより、設計情報の二重投入や転記ミスによる設計品質の低下を少なくすることが可能となる。

(1)ERDからのテーブル一覧表作成機能

ERDのエンティティをテーブル一覧表のテーブル名に対応づける。

(2)DSDからのテーブル定義書作成機能

ERDに出現するエンティティのデータ構造(DSDで記

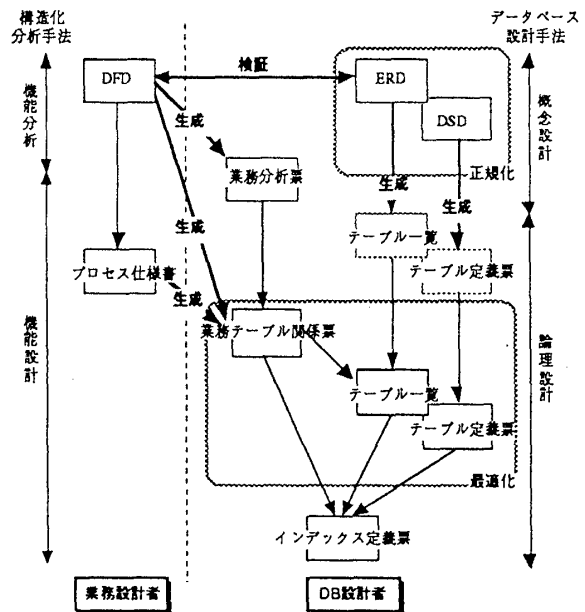


図2 統合された業務設計手順とデータベース設計手順

述)をテーブル定義書に反映する。

**step1** テーブル名をデータ構造名とする

**step2** 列名を下位のデータ構造とする

**step3** 列の属性をデータ構造の属性とする

(3)DFD/プロセス仕様書からのテーブル関係票作成機能

DFDおよびプロセス仕様書から得られる業務とテーブルとの関係を業務テーブル関係票に反映する。

**step1** プロセスの階層関係から業務情報を抽出する

**step2** データストアをテーブル情報とする

**step3** プロセスとデータストア間のデータフローをテーブルアクセス関係に対応づける

**step4** プロセス仕様書からテーブルアクセス情報を業務テーブル関係票に反映する

**おわりに**

本手法により構造化分析手法とデータベース設計技法それぞれで作成される設計情報の検証を可能とする分析作業モデルを提案するとともに相互の変換方法について明確にした。現在、本手法に基づくツールを実現し評価している段階である。

今後は設計情報の変更に伴う差分抽出および更新機能および整合性検証方法を検討する。

**参考文献**

[1]DeMarco,T.:Structured Analysis and System Specification, Yourdon Press(1978)  
 [2]NTT:データベース概念設計、データベース論理設計、阿部写真(1993)  
 [3]磯田、黒木:統合化CASEシステムSoftDAの機能,コンピュータソフトウェアVol.10,No.2(1993)