

31. データベース設計システム

日本電気(株) 中央研究所 川越 恭二・真名垣昌夫

○ あらまし

この報告ではデータベースのデータ構造の設計を設計者と計算機とが対話しながら行っていく、データベース設計システムを提案している。データベースの設計の過程を大きく6ステップに分割し、各ステップでの計算機と設計者の仕事を明確化した。概念モデルとして基本オブジェクトモデルを導入し、設計者の実世界情報記述、スキーマの生成、スキーマの操作を全てこの基本オブジェクトモデルと対応づけている。このモデルは、情報として意味を持つ最小単位の集合であり、関係モデルと類似しているが、関係モデルが関数従属、多値従属により分解していくのに対し、種々の保全情報(例えば、動作特性、写像関係)により意味を考えて分解している。このデザイン法を用いる本設計システムのシステム構成を示している。

1. はじめに

データベース管理システム(DBMS)が登場して10数年、CODASYLのデータベース言語の仕様が提案されて8年以上経過し、最近では、多くの汎用DBMSが利用されて、上はレコードタイプが100近くもある大規模なデータベースシステムから、下はミニコン・オフィスコンでの分散指向のデータベースシステムまで、多くのデータベースシステムが稼動している。しかし、データベースの利点が強調されつつも、一方ではデータベースの設計にデータ、すなわち情報の正確な把握を必要とするため、設計者の負担の増大、および設計者の個人差(スキルレベル及び情報把握の視点)による設計されたスキーマの統一性の欠陥が指摘されている。また、初期設計したデータ構造にその場主義で修正を施して実世界をゆがんで写像したスキーマを設計し使用したために、システムの拡張を困難にしているシステムもある。

今や、データベース設計に求められているのは、均質な設計、矛盾のない設計そして効率的設計が得られる手段である。

従来、この様なデータベース設計システム、あるいは支援システムとして著名なもののひとつとして、IBMのデータベースデザインエイド(DBDA)が存在する¹⁾。このDBDAは、IBMのIMS専用のシステムであり、CODASYL型のDBMSあるいは将来頻りに利用されるであろうリレーショナルDBMSに対しては適用されない。また、連関(association、属性間の写像)及びその頻度からIMSにおける論理的セグメントを生成したり、設計者へのフィードバック情報が多く設計者の意志決定作業量が多大になるため、設計者の負担はあまり減少していない。

他のデータベース設計支援システムとして、スウェーデンのBubenko氏の首唱によるCADISグループが開発したCS4システムがある²⁾。このシステムは、概念レベルでのシミュレーション(infological simulation)を非常に重要視している点に特徴がある。しかし、概念モデルとしてDIAMを基本にし

て、しかも設計者に概念モデルを扱わせるために、設計者独自の視点で実世界を写像することは困難である。

本システムは、上記DBDA、CS4システム等の持つ欠点を排除し、データベースの設計における設計者の負担を減らすシステムである。設計者は設計対象に関する情報記述、及びアプリケーションモデル記述を対話的に入力していき、スキーマ格納構造までも最終的に決定する。主な特徴は、概念モデルとして基本オブジェクトモデルを提案し、それによりCODASYL、リレーショナルなどのデータモデルへの汎用性をもたせた事、そして、論理スキーマ生成・操作前に、設計者の情報記述、処理記述の内容を評価するステップを設けて、スキーマ最適化操作と独立させた事などである。

以下の内容を簡単に説明する。2.ではデータベースの設計過程を従来の方法論とを比較しながら説明する。3.では本システムの考え方で重要である、情報記述内容及び基本オブジェクトモデルの説明、そして変換アルゴリズムの概要を説明し、CODASYLスキーマを生成する方法をのべる。また、リレーショナルモデルとの簡単な比較を行なう。4.では本データベース設計システムの構成、特長をのべる。

2. データベース設計過程

データベースの設計を容易にするには、まずデータベース設計過程を明確にしなければならぬ。図1に現在考えられている設計過程をその支援ツールと共に示す。一番左の図が本システムで採用した過程である。

データベース設計のアプローチには次の2つがある²⁾³⁾。

- (i) ユーザ要求を中心とするアプローチ
- (ii) 実世界のモデリングを中心とするアプローチ

前者は、何人ものユーザの要求を収集後、用語の矛盾をとり除き情報構造を作成する。後者は、まず対象とする実世界の情報を正確にモデル化した後、ユーザ要求に応じた情報構造に必要なならば修正する。図1において、IBMのDBDACADISのCS4は何れも(i)のアプローチを採用している。我々も、(i)のアプローチを基本としているが、次の点を強く主張したい。

- (i) 設計者に柔軟な設計アプローチである事。
- (ii) 情報記述と処理記述とを分離する事。
- (iii) キュアリのみでなくアルゴリズム的な処理の際の情報記述内容の分析が可能なる事。

(i)については、情報記述でのエンティティを、サブシステム、プロセスごとに、各設計者独自のレベルで記述することができるようにする。たとえば、エンティティひとつのマスターファイルであったり、エンティティがひとつのキュアリ対象であってもよい。

(ii)については、各々情報記述言語、アプリケーションモデル(APモデル)記述言語を設ける。前者の内容によって情報構造を決定し、後者の内容によって処理に応じた最適化を行う。

(iii)については、CS4に見られるインフォロジカルシミュレーションをアルゴリズム的な処理に対しても行ない、情報記述内容を厳密化しておく。

以下に、提案するデータベース設計過程を簡単に説明する。

データベース設計過程		IBM ¹⁾	CADISグループ ²⁾	高橋氏(富士通) ⁴⁾
情報記述	本データベース設計システム	データ要件の収集	実世界モデルの設計 ----- 情報要件の分析 ----- 情報構造の設計	概念スキーマ設計 (自然言語による) (処理内容記述)
アプリケーションモデル記述				
情報記述内容評価			インフォロジカルシミュレーション	インフォロジカルシミュレーション
スキーマ生成				論理スキーマ設計, 評価
論理スキーマ操作			データ構造の設計, 評価	アクセススキーマ設計
論理スキーマ評価				物理スキーマ設計
格納構造設計			格納構造の設計, 評価	評価
格納構造評価			(パフォーマンスシミュレーション)	アクセスバス設計 (索引付データ域)
構築			インプリメンテーション/テスト	評価
評価				ハードウェア構成設計 評価シミュレーション

図1 データベース設計過程

〔情報記述〕 設計対象のシステムに関する情報(エンティティとその属性集合及び属性に関する性質, 保全情報)をサブシステム・プロセス(業務)ごとに記述する。ここでのエンティティは一般に言われているものでなくともよい。

〔APモデル記述〕 各プロセス処理(業務)内容と処理頻度を記述する。スキーマ等は意識せずに, 情報記述のエンティティを用いた処理を記述する。

〔情報記述内容評価〕 エンティティ単位にテストデータを入力し, 出力フォーマットを定義して, 実際に望む出力が得られるか, アルゴリズムを実行するのに欠ける情報は存在しないか, 等の論理的矛盾をチェックする。

〔スキーマ生成〕 情報記述内容を用いて適当なデータモデルの基準となるスキーマを生成する。この前のステップと, このステップにおいては, 概念スキーマの存在を暗に仮定している。また, データモデルも指定する必要がある。

〔論理スキーマ操作, 評価〕 基準スキーマは, 処理, 処理効率を考慮せずに作成したものであるため, ここではデータ構造を矛盾のない範囲内で操作し, その評価を求め, 最適なスキーマを決定する。CODASYL型スキーマに対する操作として, レコード統合, 結合, 抽象化, インデックス付け, データ項目重複化, セット消去, 追加, 等がある。評価手段は, APモデル記述内容を用いて, 各プロセスごとの所要時間, 必要記憶量を非常に荒く見積る。

〔格納構造設計，評価〕データベースの実際の格納構造（ページサイズ，クラスタ，エリア）を決定する。

次の章では，本システムにおいて特に重要な考え方である。情報記述と基本オブジェクトモデルについて説明する。

3．情報記述と基本オブジェクトモデル

3-1 情報記述

設計者は，サブシステム・プロセス単位に次に示す内容を記述する。

(i) エンティティとその属性と識別子

エンティティは，通常，事実を表現し，事実を示す情報を表現するものであり，具体的事物を候補とされる。しかし，ここでは，エンティティを対象業務で扱われる情報の集まりで，設計者にとって認識しやすい単位とする。したがって，全情報をひとつのエンティティとすることもできる。属性はエンティティを構成するものである。また，エンティティを一意に識別する属性集合を記述する。

(ii) 属性特性

属性には，物理的に存在する事物（M属性），イベントを表わす属性（e属性），及び概念的な属性（c属性）が存在する。エンティティを構成する各属性について，この特性を記述する。

(iii) 属性定義域

属性にはその表わしている情報の内容に応じてデータの定義域が存在する。例えば，小学生の年齢（S.A.G.E）は，一般に $6 \leq S.A.G.E \leq 12$ である。このように，属性の意味を含めた定義域を記述する。意味を表わす単語は，システム既定義，ユーザ定義が可能である。

(iv) 写像関係

属性間の写像関係を，タイプを I（自然数，固定），B（0あるいは1），M（複数）の3つに分割して記述。また，エンティティ名に関しての記述であってもよい。

(v) 動作特性

動作特性には順特性及び逆特性の2種類が存在し，動作とは生成（C），削除（D），更新（U）を意味する。順特性とは，ある属性にある動作を施した時必ず他のある属性にある動作を施す必要のあることであり，逆特性とは，ある属性にある動作を施すには前もって他のある属性にある動作を施す必要のあることを意味する。

(vi) 誘導特性

誘導特性とは，ある属性の値が他の属性集合の値から，あるオペレーションで決定される特質である。

(vii) アグリゲート構造

アグリゲート構造とは，いくつかの属性集合の直積で表現されるアグリゲート属性とそれら属性集合との関係を表現するものである。

(viii) 存在構造

属性には、それ自体で意味の存在するものもあるが、他の属性を付加してはじめて意味のもつ、すなわち他の属性に従属した属性が存在する。この関係を存在構造と呼ぶ。

(X) 属性存在特性

ひとつのエンティティ内で、ある属性が他のある属性の値がある値（あるいは範囲）にある時、意味を持つ場合に、この性質を属性存在特性と呼ぶ。これは、エンティティの具体値集合のある部分集合に固有の属性が存在する場合に記述される。

(X) 抽象関係

抽象関係は、定義域記述で厳密に定義される場合があるが、ロール関係のような関係も存在するため、ここでは、特に重要な抽象関係をその条件があればそれと併に記述する。

(XI) イベント記述

実際の物理的実体に関する属性でなく、いくつかの実体がある動作によって結合された時に生じた属性がある。この動作をイベントと呼ぶが、このイベントとこれら実体を表現する属性集合及びイベントに従属する属性を記述する。

以上の記述によって情報を表現するが、この記述の例を各々、図2に示す。各記述は各々独立していない。これは、設計者が実世界を認識した形に近い表現で記述させるためである。したがって、設計者が記述した内容によって、スキーマが生成され、誤りがあれば、この記述に追加、修正をしていけばよい。

3-2 基本オブジェクトモデルとその生成アルゴリズム

3-1で説明した情報を用いて、エンティティを概念モデルである基本オブジェクトに分解する。基本オブジェクトは、リレーショナルモデルにおけるリレーションと同様にいくつかの属性の組が表現される。そして、保全情報をできる限り埋め込みながら、しかも情報を失わない範囲でエンティティ集合を分解したところの情報が意味を持つ最小の単位である。

以下に基本オブジェクト生成の概要を記述する。

A. エンティティの統合

A-1: 属性存在特性を有するエンティティはこの属性により分割。

A-2: 同じ属性集合あるいは識別子が同じ属性集合を持つエンティティは統合し、ひとつのエンティティとする。

B. 基本オブジェクト(BO)分解

B-1: 属性特性で物理属性、イベント属性のものをBOの代表属性として基本オブジェクトを生成。

B-2: 代表属性でないもので代表属性と存在構造、写像、動作特性で関係のある属性を、その代表属性に関するBOの付随属性とする。

B-3: イベントの識別子を全て含むエンティティがあれば、イベントを表わすBOを生成する。

B-4: エンティティ内に物理属性、イベント属性がなければそれ自体をイベントBOとする。

B-5: どのBOにも入りこまなかつた属性は、その所属するエ

```

BEGIN ICL
-----
SUBSYSTEM   カ*クワイ-サフ*システム
PROCESS    カ*クワイ-トウロク
ENTITY     カ*クワイ-エンティティ (KEY カ*クワイ-メイ)
-----
ATTRIBUTE
;          カ*クワイ-メイ   =シ*ンメイ, M
;          シ*コウシヨ      =シ*ユウシヨ,C,EX-カ*クワイ-メイ
;          タンシ*ヨウヒ*   =キンカ*ツヒ*,C,EX-カ*クワイ-メイ
;          カ*クワ*ーメイ   =ナマエ,M,1;M-カ*クワイ-メイ,EX-カ*クワイ-メイ
;          ホシヨウエン     =シ*ンメイ,N,1:1-カ*クワイ-メイ,EX=カ*クワイ-メイ
;          ホシヨウエン-シ*コウシヨ =シ*コウシヨ,C,EX-ホシヨウエン
-----
PROCESS    クラフ*ートウロク
ENTITY     クラフ*ーエンティティ (KEY カ*クワイ-メイ,クラフ*ーコート*)
-----
ATTRIBUTE
;          カ*クワイ-メイ,
;          クラフ*ーコート* =コート*,M:M-カ*クワイ-メイ
;          クラフ*ーメイ   =ナマエ,E,G-クラフ*ーコート*
;          ニコウフ*ーヒ*   =キンカ*ツヒ*
;          クラフ*ーソウセツヒ* =キンカ*ツヒ*,C,EX-クラフ*ーコート*
-----
PROCESS    クラス*ートウロク
ENTITY     クラス*ーエンティティ
-----
ATTRIBUTE
;          クラス*ーメイ
;          カ*クワイ-メイ =シ*ンメイ,M,M:M-クラス*ーメイ
;          キヨウシ*ーメイ
;          キヨウシ*ツ*ン*
-----
END
-----

```

図 2. 情報記述言語例

- E - 1 (学生名, 住所, 誕生日, 学部名, 保証人, 保証人住所)
- E - 2 (クラブ名, 学生名, 入部日, クラブコード, クラブ創設日)
- E - 3 (クラス名, 学生名, 教室 No)

↓ [生成]

- | | | |
|--------------------------|---|--------------------------|
| B 0 1 (学生名, 住所, 誕生日) | , | B 0 8 (学生名, 保証人名) |
| B 0 2 (学部名) | , | B 0 9 (クラブ名, 学生名, 入部日) |
| B 0 3 (保証人名, 保証人住所) | , | B 0 1 0 (学生名, 学部名) |
| B 0 4 (クラブ名, コード, 創設日) | , | B 0 1 1 (クラス名, 学生名) |
| B 0 5 (クラス名) | , | B 0 1 2 (クラス名, 教師名) |
| B 0 6 (教師名) | , | B 0 1 3 (クラス名, 教室 No) |
| B 0 7 (教室 No) | | |

↓ [統合]

B 0 5 ~ 7 の消去

↓ [基準スキーマ生成]

B 0 1, 3, 8, 1 0 の結合, B 0 9 から関係レコードの生成,
B 0 1 2, 1 3 の結合, など

図 3 基本オブジェクト生成例

から生成した全ての B O 内に入れる。

B - 6 : 代表属性間と写族, 存在構造, 動作特性の関係があれば, 関係を表わす B O を生成する。

C . 基本オブジェクトの統合

C - 1 : 代表属性間に関係を表わす B O があり, 各 B O に共通の付随属性があれば, 関係 B O に移す。

C - 2 : 二つの B O で同義のものは一つの B O に統合する。

C - 3 : 代表属性のみからなる B O を関係を表わす B O に埋め込む。

C - 4 : 写像が冗長なものであれば, 動作特性を犯さない範囲で削除する。

C - 5 : 抽象関係にある B O については, 具象 B O 集合に共通な属性は抽象 B O に移す。

図 - 3 にこの生成マルゴリズムの簡単な例を示す。

3 - 3 . リレーショナルモデルとの比較

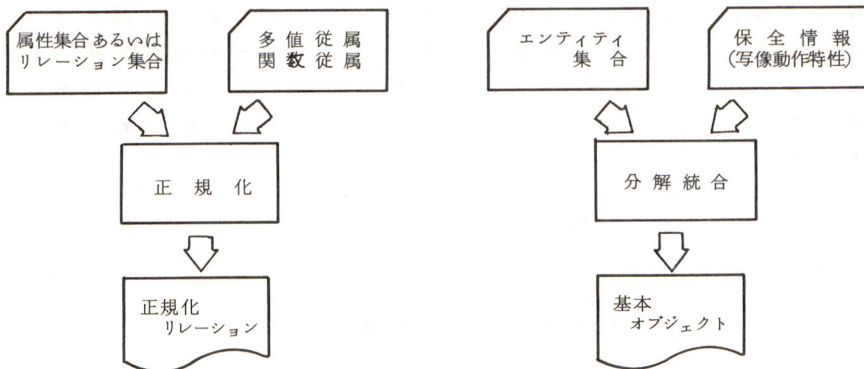
リレーショナルモデルは, 本モデルと同様に, 基本的考え方として保全 (インテグリティ) を表の形でできるだけ埋め込みながら分解 (正規化) していく。しかし, 基本オブジェクトはそのアプローチが異なる。違いを簡単にのべる。

(i) 導出方法 : (図 - 4) 本システムでは意味的な記述が明解である。

(ii) キーの扱い : リレーショナルモデルではリレーション中にキーの存在を仮定し, 明確にする。基本オブジェクトでは, 基本オブジェクト内の全キー属性の存在にかかわらずキー記述を別にする。代表属性で基本オブジェクトを代表させる。

(iii) 関数従属性 : 基本オブジェクトモデルでは, 関数従属性は写像, 動作特性, 存在構造, 誘導特性, に埋め込まれていると考えられる。また, 関数従属性の意味を考慮しているので, 2つの属性間に複数の関数従属性を扱うことができる。

(iv) モデルの意味 : 基本オブジェクトは代表属性, 関係, イベントで明確に意味が保存されている。



(a) リレーショナルモデル

(b) 基本オブジェクトモデル

図 4. 基本オブジェクトモデルとリレーショナルモデル

3-4 CODASYL型スキーマの生成

リレーショナルモデルからCODASYL型スキーマを生成するには、リレーションをエンティティと関係の2つのタイプに分け、基本的には前者をレコードにし、後者をセットとするのである⁶⁾が、写像及びリレーションの意味が厳密でないため困難である。しかし、基本オブジェクトにおいては、その生成形態、写像関係から、次の様にして(基準)スキーマを生成することができる。

D. 基本スキーマの生成

- D-1: 各BOに識別子を追加する。
- D-2: 代表属性を持つBOは、最も抽象的なBOのみレコードとし、その他のBOはレコードとしない。しかし、属性は全てレコードのデータ項目とする。
- D-3: イベントを表わすBOは、その識別属性をレコードとし、更にイベントレコード、及びセットを生成する。
- D-4: 関係BOは、その代表属性のレコード化、写像に応じて、セットを生成するか、レコードを生成するか、あるレコード内に埋め込むか、を行なう。ここで、抽象属性をもつ代表属性については、それを抽象属性でおきかえる。

4. データベース設計システム

本データベース設計システムは、2.3でのべた考え方を基本としている。

図-5に、本システムの概要を示す。

図-6に、情報記述の論理的矛盾をチェックするためと、アプリケーションの処理時間及び記憶容量を見積るために、設計者が入力するアプリケーション記述の例を示す。

図-7に、本システムから生成されるスキーマの表示例を示す。

図-8は、本システムのシステム構成である。

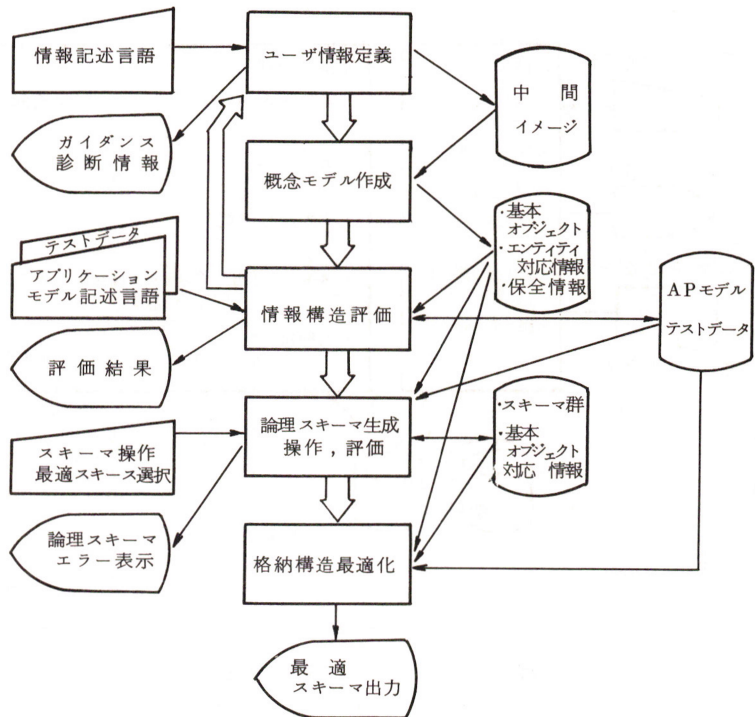


図-5 データベース設計システム概要

SUBSYSTEM カ"クセイ-サフ"システム
 PROCESS カ"クセイ-トウロク
 ENTITY カ"クセイ-エンティテイ"
 .OCCURS 500
 .PREC.(A) ネ
 SUB-PRO カ"クセイ-トウロク"
 .SEQ.NO 1
 .SELECT カ"クセイ-メイ (KEY)
 ,(ADD)
 タンシ"ヨウヒ" (ADD)
 カ"クフ"ーメイ" (ADD)
 SUB-PRO ホシヨウニン"トウロク"
 .SEQ.NO 2
 .SELECT カ"クセイ-メイ (KEY)
 ホシヨウニン"ーメイ" (ADD)
 ホシヨウニン"ーシ" ヲウ
 ,(ADD)
 OUTPUT-FORMAT OF-1

```

** スキーマ **
VER.= 2
DATE = 78-11-01
AUTHOR = KAWAGOE

*SCHEMA NAME IS カ"ツクウ"ースキーマ
*RECORD NAME IS カ"クセイ-レコード"

02 カ"クセイ-NO (KEY) PIC 9(6)
02 カ"クセイ-メイ PIC X(15)
02 カ"クフ"ーメイ PIC X(14)
02 タンシ"ヨウヒ" PIC 9(4)
03 タンシ"ヨウケ" PIC 9(2)
03 タンシ"ヨウニツキ" PIC 9(2)
03 タンシ"ヨウヒ" PIC 9(2)
02 シ"ユウシヨ PIC X(30)
02 テ"ンワ"バンコウ PIC 9(15)

*SET NAME IS カ"クセイ-クラス"ーセット

OWNER クラス"レコード"
MEMBER カ"クセイ-レコード"
MANDATORY AUTOMATIC
  
```

図 6 A P D L 例

図 7 スキーマ出力例

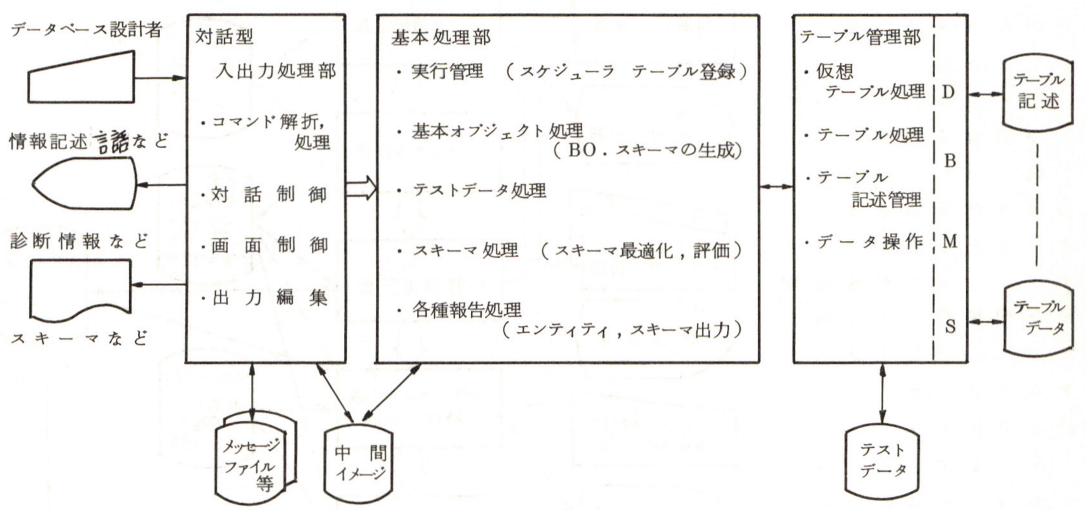


図 - 8 データベース設計システムのシステム構成

本システムの特徴は次の通りである。

- 1) 設計者^がデータベースを対話的に設計していくシステム
- 2) 情報記述から格納構造決定まで扱う総合システム
- 3) 概念モデルとして基本オブジェクトモデルの導入
- 4) データモデル (CODASYL, リレーショナル) への汎用性
- 5) 設計者の視点での情報記述のシミュレーションを行い, 論理的矛盾を検出
- 6) ドキュメントとして本システム内のデータが利用できる。
- 7) 既存のファイルシステムからデータベースシステムへの変換において, ファイル記述からスキーマの生成を行なうツールとして利用可能。入力エンティティとスキーマとの対応が基本オブジェクトを中間にして保たれているので, データの変換等にも利用可能であろうが本システムでは対象外である。

5. おわりに

本システムは, データベースのスキーマを設計するデータベース設計システムの実験システムである。

本報告で提案した基本オブジェクトモデルは, リレーショナルモデルが理論的であるの^に比べ, やや発見的 (heuristic) なアプローチであるが, 設計者に一面的な記述をさせるよりは, 種々の面からの記述によって生成されるため, 柔軟性を要求される設計には適用性が高いと考えられる。本報告では, 紙面の都合上十分なアルゴリズムの記述ができなかったが, 実際には対話処理を重視したシステムとなっている。

今後, 本システムを適用したデータベース設計を数多く行なうことによって, 設計者とのインターフェース及びアルゴリズムの現実への適用性を増加させていきたいと考えている。

最後に, 本システムを開発するに当たって, 多くの貴重な助言をくださった関係各位に感謝致します。

参 考 文 献

- [1] I B M : Data Base Design Aid, Ddsigners' Guide, Product Product, Program No. 5748-XX4, GH20-1627
- [2] Berild, S. and Nachmens, S. : CS4-A Tool For Data Base Design by Infological Simulatim, Proc on VLDB, '77, Part II pp85-94
- [3] Bubenko, J.A.Jr. : Validity and Verification Aspect of Information Modelling, Proc. on VLDB, '77, pp556-565
- [4] 高橋 : データベース設計過程とデータベース設計支援ツール, 情報処理 D B M S 研究会資料, 8-2, 1978
- [5] Chen, P.P. and Yao, S.B. : Design and Performance Tools for Data Base Systems, Proc, on VLDB, '77 pp3-15
- [6] Aurdal, E., and Splvberg, A. : A Multilevel Procedure for Design of File Organization, NCC '77, pp509-521
- [7] Smith, J.M. and Swith, D.C.P. ; Data Base Abstractions : Aggregation and Generalization, ACM TODS, Vol.2, 1977

本 PDF ファイルは 1979 年発行の「第 20 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場 (=情報処理学会電子図書館) で公開されているにも拘らず、古い報告集には公開されていないものが少からずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者 (論文を執筆された故人の相続人) を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者検索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思えます。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長 (tsuji@math.s.chiba-u.ac.jp) までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>