

A3-2 「汎用データベース操作システムの機能解析」紹介

鈴木道夫（電力中央研究所）

1. CODASYL システムズ委員会調査報告書

CODASYL のシステムズ委員会は、1969年5月に「汎用データベース操作システムの調査」を発表した。これはデータベース操作システムの機能を列挙し、既存の9種のシステムごとに章をたて、列挙した機能の詳細を記述した調査報告である。そして1971年5月に第2次の調査報告「汎用データベース操作システムの機能解析」を発表した。本稿は第2次報告書の内容を紹介するものである。

第2次報告書では、第1表に示す10種のシステムをとりあげ、前報告書と異なって機能ごとに章をたて、各機能についてこの10種のシステムを比較する方法をとっている。全体として、第1次報告書よりくわしくかつ釣合いのとれた記述になるよう努力している。この委員会の活動はしかし既存のシステムの機能を完全に記述することが最終目標ではなく、「統合されたデータベースシステムの共通言語機能の仕様を開発すること」にある。この調査報告はそのための手段である。

第2次報告書は序論と全10章よりなる。序論は報告書本文の序論であるとともに独立した正式文書として雑誌などに発表されることを目的に書かれたものである。一方本文は、1.概要、2.データ構造、3.データ定義、4.問合せ機能、5.更新、6.生成機能、7.プログラマ機能、8.データ管理機能、9.記憶構造、10.操作環境より構成されている。第1表に示した各システムの概略は第1章で述べられている。

第1表 CODASYL システムズ委員会 第2次報告書の10種のシステム

独立言語方式のシステム

システム名		作成者	最初のシステムが利用できるようになった日付
通称	名称		
GIS	Generalized Information System	IBM	1969年9月
MARK IV	MARK IV	Informatics Inc.	1968年(全システム)
NIPS/FFS	NMCS Information Processing System	DCA, IBM, National Military Command System	1968年7月
TDMS	Time-shared Data Management System	SDC (System Development Corporation)	1969年
UL/1	User Language/1	RCA	1969年

親言語方式のシステム

システム名		開発母体	最初に発表された日付	親言語
通称	名称			
COBOL	コボル開発報告 (COBOL Journal of Development)	CODASYL	1970年4月 (仕様のみ)	
DBTG	データベース 作業委員会による 提案	CODASYL プログラム言語 委員会	1971年4月 (仕様のみ)	COBOL
IDS	Integrated Data Store	Honeywell Information Systems (当時のGE)	1963年	COBOL
IMS	Information Management System	IBM, North American Rockwell	1969年9月	COBOL PL/I 360 アセンブラ
SC-1	System Control - 1	Western Electric, Auerbach	1970年7月	なんでも可?

2. データベース操作システムとデータ独立 (序論参照)

データベース操作システムとは、さまざまな利用者がそれぞれの目的に応じて使用するデータを有機的に統合し、その統合されたデータ(これをデータベース(data base)と呼ぶ)の操作に必要な機能を備えたソフトウェアとハードウェアの組合せである。このようなデータベース操作システムでは、データベース中の同一のデータを異なる利用者が使用することが起こるので、データをできるだけプログラムから独立させておくことが望ましい。このことは一般にデータ独立(data independence)という概念で呼ばれている。

データ独立の概念を導入すると、データベースの利用者を二つのグループに分けることができる。その1つはデータ管理者(data administrator)と呼ばれる利用者であり、これはデータベース全体の構成と保守に全責任をもつものである。もう1つはその他の利用者と呼ばれ、データ管理者によって管理されたデータを利用するものである。後者はさらに応用プログラマー(application programmer)、プログラムしない利用者(non-programming user)、パラメータ利用者(parametric user)に細分される。

3. データ構造と記憶構造 (第2, 3, 6, 8, 9章参照)

一般にデータは、ある適用実体(application entity, たとえば従業員, 組織など)に関して発生する。適用実体のもつ内部的構造ならびに相互関係に即したデータ間の論理的関係をデータ構造(data structure)という。

適用実体を構成する最小単位のデータ要素を項目(item), 項目をいくつか集めたものを集団(group)という。集団はさらに別の集団や項目と共にまとめてもよく、こうしてできる集まりも集団という。適用実体によっては、それを構成する集団の間に支配・従属の関係をもたせるのが必要なことがあり、このような関係を集団関係(group relation)という。以上のようにして構築したデータ要素の集まりが適用実体を表わすとき、この集まりをエントリ(entry)と呼ぶ。そしてエントリの集まりをファイル(file), ファイルの集まりをデータベースという。

なお、一つのファイルを構成する二種類以上のエントリの間とか、データベースを構成する二種類以上のファイルの間に支配・従属の関係を定義してもよく、このような関係も集団関係と呼ぶ。

項目、集団、集団関係、エントリ、ファイル、データベースをそれぞれ**構造の型 (structure type)**と呼ぶ。

構造の型をその属性(データの値の範囲、表現形式など)と共に表現したものを**スキーマ (schema)**という。また、スキーマに対応する実際のデータをそのスキーマの**具体値 (instance)**という。

データベース操作システムには、スキーマを具体的に記述する手段が必要になる。調査されたシステムのうち DBTG の提案では、この目的のために**データ記述言語 (Data Description Language, 略して DDL)**を定めている(その他のシステムでも同じ目的のための命令を持っているので、以下の説明ではデータ記述言語という用語で統一しよう)。利用者(通常、データ管理者)はデータ記述言語によってスキーマを記述する。これをシステムが処理すると、システム内にデータ構造が定義される(この作業を**データ定義 (data definition)**という)。

一方、スキーマの具体値(つまり実際のデータ)は、用いる記憶装置にふさわしい方法で格納される必要がある。格納される実際のデータのありさまは、データ構造に対して**記憶構造 (storage structure)**という。データ構造と記憶構造の対応づけを**写像 (mapping)**または**結合 (binding)**という。写像の時点はデータ定義時、プログラムコンパイル時、プログラムロード時、ファイルを開く時、値の呼び出し時というようにシステムによっていろいろある。また写像の方法は、用いる記憶装置の物理的機構に依存するので大部分はシステムが定めるが、利用者がこれがある程度指定できるシステムもある。写像の情報は**登録簿 (directory)**の形でシステム内に保持され、データベースに対する問合せや更新の要求(いずれも後述)があったときにそれを参照する、という方式のシステムが多い。

4. 利用者の機能 (第4, 5, 7章参照)

データベース操作システムには、利用者がデータベース中にデータを投入し、管理し、利用するための諸機能が設けられる。データベースに新たなファイルを作成するには、データの投入に先立ってデータ定義を行わなければならない。また、既存のファイルにデータを追加、変更、削除することをファイルの**更新 (update)**と呼ぶ。ファイル更新では、既存のデータ構造を改訂することはできない。データ構造の改訂は**データ再定義 (data redefinition)**と呼び、データ定義に準ずる扱いを要する。なお、新しいファイルを作成することをファイルの**生成 (creation)**と呼ぶが、このファイルにデータを投入するには、一般に空なファイルに対して実際のデータによる更新を行なう方法がとられる。以上の機能は主としてデータ管理者用の機能である。

一方、データ管理者によって生成、更新されているデータベースから、その他の利用者がデータを抽出、分類、処理加工して報告書を作成することを一括して**問合せ (interrogation)**と呼ぶ。問合せの方法は用いるシステムによって変化に富むが、大別すると (i)システムに組み込まれた手順にパラメータを付与する方式と、(ii)COBOLまたは類似の手続き言語を用いたコーディングによる方式に分かれる。一般に前者の方式を**独立言語方式 (self-contained system)**、後者の方式を**親言語方式 (host language system)**という。利用者にとっては独立言語方式はCOBOLなどを学ぶ必要がないから簡便という点で長所があるが、問合せ機能の種類が限定されるという短所がある。実際のシステムでは、独立言語方式に手続き言語的な要素を加える(またはその逆)というように両方式を組み合わせるものが多いようである。

第2次報告書は以下のメンバーによって現在翻訳中である。西村怨彦、植村俊亮(電総研)、島田勝八郎(日科技研)、木南公統司(開発計算センタ)、中川友康、鈴木道夫、高橋誠、田中正明(電力中研)

本 PDF ファイルは 1965 年発行の「第 6 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、一定期間の権利者搜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>