

COBOLのデータベース機能

3D-6

鈴木 英輔 床分 真一 横山 一郎 (日立製作所 ソフトウェア工場)  
 田中 和明 (日立製作所 システム開発研究所)

1. はじめに

データベースシステムの大規模化、複雑化に伴い、データベースを利用するユーザプログラムも拡大、複雑になってきており、より整った開発体制が望まれている。その中でも、COBOL言語に対する要望は高い。しかしながら、そのCOBOLのデータベース機能は、従来から次のような問題点があった。

(1) 標準的なデータベース仕様として、CODASYL COBOLの仕様が存在しているが、これ自身規格化されていないため、現実には、処理系により様々な仕様があり、統一がとれていない。

(2) プリプロセッサなどでのサポートが多いため、デバッグツールなどの開発支援ツールの使用が困難である。また、コンパイル中に十分なエラーチェックができず、実行時でしか発見できないエラーが多くある。

これらの問題点を解決するため、日立 VOS3 COBOLでは、CODASYLとANS (案) のデータベース機能を参考にし、COBOLデータベース機能をサポートした。

本報では、COBOLデータベース機能の標準仕様について説明しながら、日立 VOS3 COBOLのデータベース機能の特徴について述べる。

2. データベース機能の標準化の動向

CODASYL COBOLのデータベース機能は、データベースの定義と操作の分離、スキーマやサブスキーマの概念の導入などに特長がある。そこでは、「スキーマのためのデータ記述言語」はCOBOLと独立の言語として扱うこととし、「COBOLサブスキーマのためのデータ記述言語」と「COBOLデータ記述言語」は、1976年版のCODASYL COBOLから言語仕様として含まれるようになった。

ANSでは、CODASYLの仕様を基にスキーマ用データ記述言語、サブスキーマ用データ記述言語及びデータ操作言語 (DML: Data Manipulation Language) を、NDL (Network Description Language)

としてまとめ、NDLがISOのDraft proposalとなり、1986年に発表された。

これらの結果を基に、ANS COBOLのデータベース機能案が1985年にまとめられた。この特徴は、次の2つである。

(1) ANS NDLとの親和性を前提とし、NDLの機能との対応に基づいて、仕様を規定している。

(2) サブスキーマ節は、実行単位中で1つしか記述できない。

図1の左側にデータベース仕様の標準仕様の関係を示す。

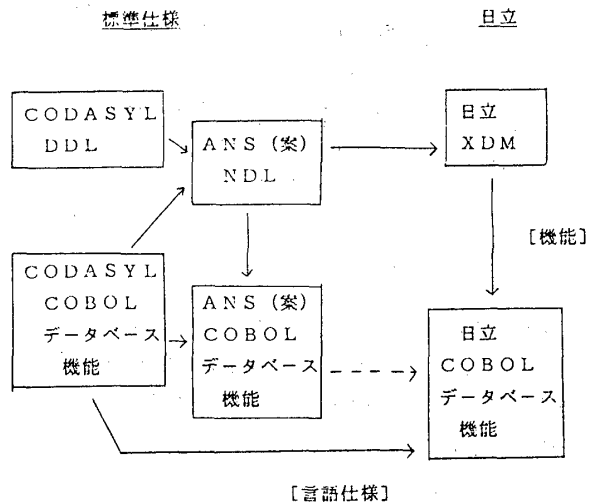


図1. データベース機能の標準仕様と日立COBOLの関係

3. 日立COBOLデータベース機能の特徴

日立では、DBMSとして、ANS NDLを基に統合型XDMを開発した。このXDMに対する、COBOLのデータベース機能をサポートした。

日立COBOLのデータベース機能の特徴を次に示す。

The Data Base Facility of COBOL

Eisuke SUSUKI, Shinichi TOKOWAKE, Ichirou YOKOYAMA, Kazuaki TANAKA  
 HITACHI, Ltd.

(1) サブスキーマの概念を導入し、CODASYLのサブスキーマ仕様を採用した。

(2) 日立COBOLのデータベース機能は、図1に示すように、その機能を日立DBMSのXDMのDMLを基に作成し、COBOL言語としての特徴をCODASYL COBOLをベースに、ANS COBOL データベース機能案を参考にしながら決定した。図2に、日立COBOLのデータベース機能の使用例を示す。

```

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. HITACHI.
000300 ENVIRONMENT DIVISION.
000400 DATA DIVISION.
000500 SUBSCHEMA SECTION.
000600 DB DBNAME01
000700 WITHIN SCHEMA-NAME
000800 STATUS ステータス.
801 DB 000100 01 レコード.
802 DB 000200 02 データ PIC X(5).
000900 WORKING-STORAGE SECTION.
001000 77 ステータス PIC X(5).
001100 PROCEDURE DIVISION.
001200 HITACHI-01.
001300 FETCH UPDATE FIRST レコード
001400 WHERE データ = 'HITAC'.
001500 IF ステータス = '00000' THEN
001600 MOVE ' ヒタチ' TO データ
001700 MODIFY レコード.
001800
001900 STOP RUN.
    
```

(プログラムの説明：サブスキーマ節でサブスキーマ記述項を書くとディクショナリから801と802を展開する。手続き部では、FETCH文でレコードを検索、入力し、ステータスの結果で、データを変更する。次に、そのレコードを更新する。)

図2. データベース機能の使用例

(3) COBOLデータベース機能をコンパイラで、サポートした。この利点として、他のCOBOLの機能と同様に、COBOLの開発支援ツールが使用可能となった。また、コンパイル時にエラーチェックが十分に行えるようになり、実行時でしかわからなかったエラーをコンパイル時に発見できるようになった。

(4) 図3に示すように、データベースの各種情報がおさめられているディクショナリをコンパイル時に、参照、更新することで、サブスキーマ、スキーマに対するユーザプログラムの参照情報が管理できるようになった。

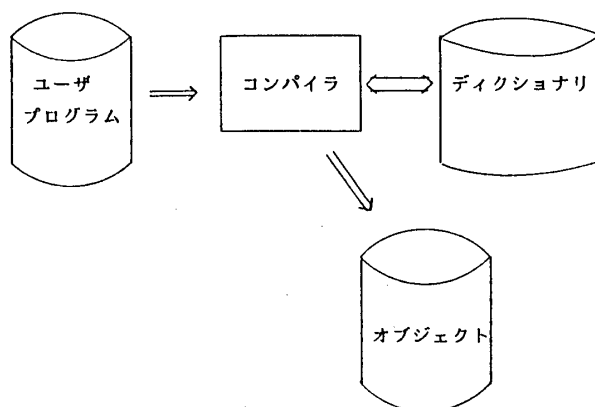


図3. ユーザプログラムのコンパイル方式

4. おわりに

今回日立 VOS3 COBOLは、CODASYL COBOLをベースにANS COBOLデータベース機能案を参考にし、データベース機能の言語仕様をサポートした。

最後に、DBシステムが、情報システムの中心となっている今日、COBOLのデータベース機能の早急な標準化を望む。

5. 参考文献

- (1) CODASYL COBOL Committee Journal of Development 1984, the Secretariat of the Canadian Government EDP Standards Committee (1984).
- (2) CODASYL Data Description Language Committee Journal of Development 1981 同上 (1981)
- (3) Draft Proposed American National Standard Database Language NDL August 1984
- (4) Network Database Language Document ISO/TC97/SC21/WG3.0 N144 1986
- (5) COBOL Information Bulletin Number 22 Preliminary Draft Proposed American National Standard COBOL Network Database Interface, American National Standards Committee (1985)
- (6) 斎藤真一他 データマネジメントシステムXDM (1) -総論- 情報処理学会第32回全国大会論文集 pp 839 840 (1986)
- (7) 横山一郎他 XDM (2) 同上 pp 841 842
- (8) 今城哲二他 COBOLデバッガでのDB/DC単体テスト機能 情報処理学会第33回全国大会発表論文集3D-7