

メタ定義データ合成による

4M-2

ディクショナリ構築方式の提案

田中 和明*, 金居 貞三郎*, 高橋 典幸**, 秋葉 俊夫**

((株)日立製作所 *システム開発研究所 **ソフトウェア工場)

1. はじめに

大規模化・複合化した計算機システムの大量な定義データの管理負担を軽減するため、システム全体の定義データを一元管理するディクショナリへの期待が高い。

従来、DB/DCシステムの定義データ、特に、数万個に及ぶデータ項目を専門に管理するデータディクショナリが実現されてきた。[1]

その管理対象をユーザシステム全般に渡って管理していこうとする考えは、ISOにおいて標準化中のRMDM(Reference Model of Data Model)やIRDS(Information Resource Dictionary System)の規格案の中でいわれている。IRDSは定義データを管理するディクショナリに加え、ディクショナリ構造を定義したメタ定義データを管理するメタディクショナリを持ち、柔軟なディクショナリ構造をとれる仕様となっている。[2][3]

ところで、ユーザシステムは、メカ提供の種々のプログラム、例えば、OS、DB/DC、システム開発支援等から構成され、個々に、その動作方法を規定する定義データを必要とする。ユーザシステムを構築するプログラム群の多様性からディクショナリで管理する定義データの種類も変動する。これに追隨してメタ定義データをいかに提供し、収集し、ディクショナリ構造に反映させるかは、IRDS規格の範囲外である。本稿では、このインプリメンタ任せのディクショナリ構築方式を提案する。

2. データ構造

ユーザシステムの全プログラムに必要な定義データを登録するディクショナリを創成するためには、①ユーザシステムとそれを構成するプログラム、②プログラムとその実行に必要な定義データ項目、とを把握する必要がある。①②が決まれば、③定義データ項目のインスタンスである定義データを登録するディクショナリが創成できる。『多:多』であるこれらの関係を図1に示す。

本稿では、①②③の各インスタンス、および、それらのメタ定義データの提供者と管理方法を、表1のように規定する。

3. メタ定義データの提供方法

ユーザシステムのディクショナリ構造は、次の理由から予め特定できない。

- (1)ユーザシステムのプログラム構成は、ユーザによっても、更には、1ユーザ内でも本番・開発システムとで異なるのが一般的である。
- (2)プログラムのバージョンが異なると、メタ定義データの一部が異なる場合がある。
- (3)各プログラムに必要な定義データ項目に重複がある。

本稿では、各プログラムの提供媒体であるマスターテープ中に、そのプログラムの実行に必要な全定義データに関するメタ定義データを含めて提供する方法を提案する。

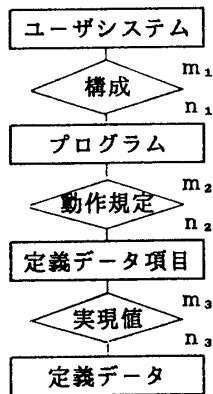


図1 データ構造

表1 管理データの提供者・管理場所

管理データ	提供者	管理場所
メタ定義データ	ディクショナリシステム, ③の一部 プログラム提供者	メタ ディクショナリ
ユーザシステム内のプログラム名	利用者	ディクショナリ
プログラム内の定義データ項目名	各プログラムの提供者	ディクショナリ
定義データ	利用者	ディクショナリ

メタ定義データの構成

```

<メタ定義データ>
 ::= { <メタ定義データ名>
      (属性)
      } ...

<メタ定義データ名>
 ::= <テーブル名>
      <カラム名>
      (世代番号)

<属性> ::= <データ型>
          (制約条件)
  
```

4. ディクシヨナリ構築方式

4.1 構築手順

- (1) 先ずメタディクシヨナリを創成する。創成のためのメタメタ定義データはディクシヨナリシステムに付随して提供されるものとする。
- (2) メタ定義データをメタディクシヨナリに登録する。
- (3) メタ定義データに基づき、ユーザシステム名とプログラム名、プログラム名とメタ定義データ名と属性を登録する基本ディクシヨナリ構造を創成する。
- (4) 利用者が購入したプログラムのマスタテプから入力した、プログラム名とメタ定義データ名とその属性データをディクシヨナリに登録する。
- (5) 利用者が入力したユーザシステム名とそれを構成するプログラム名をディクシヨナリに登録する。
- (6) ユーザシステム名を指定したディクシヨナリ構築要求に応じて、ユーザシステムを構成する全プログラムの定義データを登録するディクシヨナリを構築する。

4.2 メタ定義データ合成手順

- (1) 創成時：
 - (a) ユーザシステムを構成する全プログラムで必要とするメタ定義データを集めて、メタ定義データ名でソートする。
 - (b) メタ定義データ名が一致するものは一方を排除する。
 - (c) (テーブル名)(カラム名)が一致し、(世代番号)が異なる場合、次の基準で一方を排除する。

- ①カラム長だけが異なる場合、長い方に合せる。
 - ②制約条件だけが異なる場合、制約の緩い方に合せる。
 - ③①②以外ならば、最新の世代番号の方を選ぶ。
- (d) CREATE TABLE により、定義データ管理用ディクシヨナリを創成する。

(2) 拡張時：

- (a) ユーザシステムに追加された新プログラムで必要とするメタ定義データに対して、(1)(a)(b)(c)を実施。
- (b) 創成時に作成した(1)(c)で得られたカラム群と比較し、(1)(b)(c)の基準で重複部分を排除する。
- (c) 既存ディクシヨナリ構造に新テ-ブル、新カラムを追加するだけならば、CREATE TABLE、ADD COLUMN により、ディクシヨナリ構造を拡張する。
構成変更が必要ならば、定義データを一旦退避し、ディクシヨナリを再創成後、定義データをロードする。

5. おわりに

本稿では、メタ定義データ合成によるディクシヨナリ構築方式を提案した。本方式によれば、ユーザシステム構成に合せたディクシヨナリ構造を実現できる。

6. 参考文献

- [1] 和歌山、他：XDM(4)-データディクシヨナリ/ディレクトリ-, 情報処理学会第32回全国大会, 1986
- [2] 穂鷹良介：データ管理参照モデル, 情報処理(1988.3)
- [3] 溝口徹夫：情報資源辞書システム(IRDS), 情報処理(1988.3)

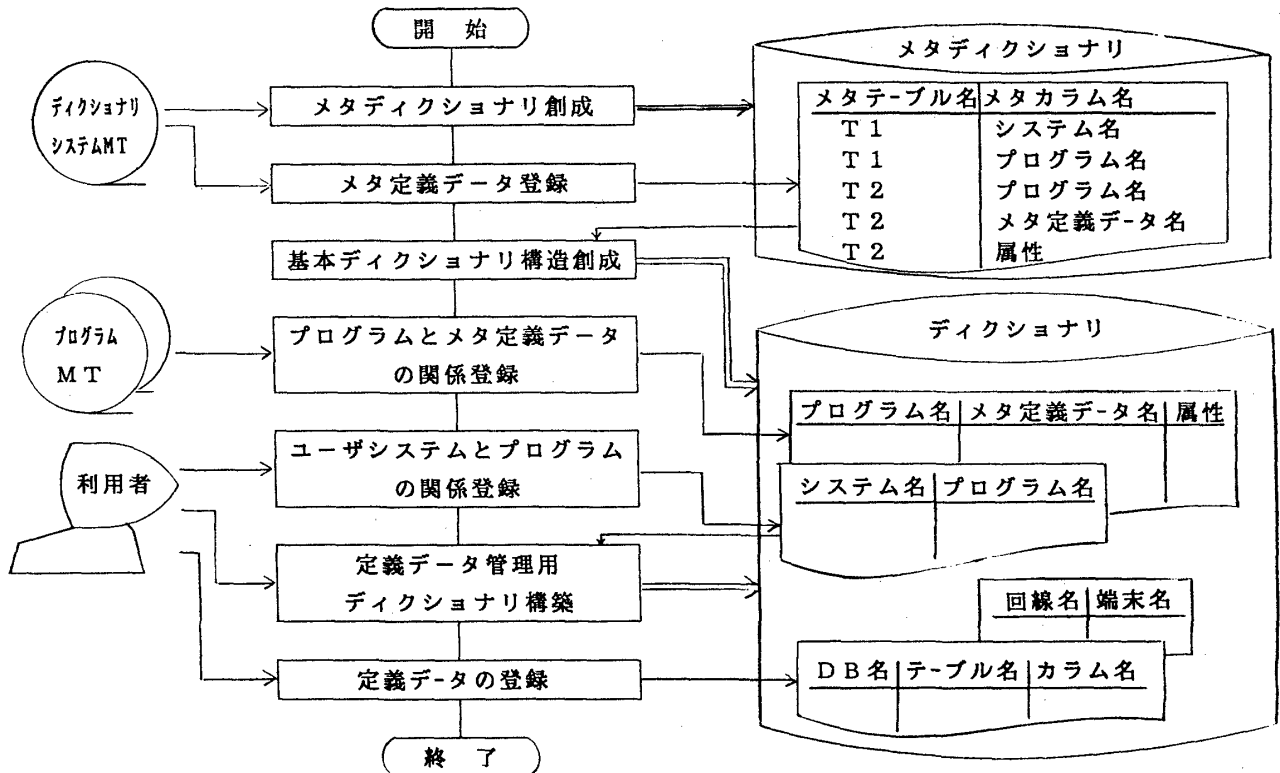


図2 ディクシヨナリ構築方式