

ネットワークシステムに於ける データベース構築の一考察

広野 真吾 島田 正徳 谷田 敏嗣

富士通ネットワークエンジニアリング株式会社

(口頭発表のみ)

ネットワークシステムに於けるデータベース構築の一考察

広野 真吾 島田 正徳

富士通ネットワークエンジニアリング株式会社

最近のネットワークシステムでは、端末数やセンサー点数やメディアの種類あるいはサービス地域などのあらゆる面に於いて大規模なものになってきている。例えば、ビル管理システムでは管理点数が10,000点にも及ぶ規模のものが多く、またマルチメディアネットワークシステムでは音声、データ、画像など多くの情報メディアを取り扱うことと、サービスエリアが広範囲にわたることとで大規模で複雑なシステムになるものが多い。

これらのシステム設計は、膨大なデータ作成及び管理の積み重ねであり、人手による作業では限界にきており、データベースシステムを利用した設計管理ツールが必要となっている。

本稿では、複合システムの構成機器の特質を生かしたデータベースの構築手法や内部構造などの例を紹介するとともに、データベース管理システムへの要求課題について述べる。

Construction and use of the Network design database system

Shingo Hirono Masanori Shimada

Fujitsu Network Engineering Limited

Recently, the scale of network system becomes larger as for a number of terminals and sensors, kind of media and service area.

For example, building-automation system has to manage more than 10,000 parts, and multi-media network system is getting more complex and larger to construct, because it handles multi-media (sound, data, graphic) networks and vast service area. As it is getting harder to construct the network and change its structure, we need a design management tool using database.

In this paper, we introduce the way of network construction, inside structure of database system, and point out the attentions when they use it.

1. まえがき

近年、ネットワークシステムを取り巻く環境は、多様化、高度化するニーズの高まりや標準化・国際化の進展などにより複雑で大規模化する傾向にある。

ネットワークシステムの種類は、ビル管理システムのような狭域でありながら接続数が膨大なものから企業内ネットワークのように国際通信かつマルチメディア通信を実施しているものまで多種多様なシステムが存在する。

これによりネットワーク設計作業に大きな負荷が掛かるようになった。

ネットワークシステムの設計は、一般的に物理的な設備の設計や論理的な通信路の設計などがあり、設計するデータ量は膨大でありデータ管理作業も多く、人手では限界にきている。

そこで、ネットワーク設計者（ネットワークエンジニア）に必要な設計支援ツール、データ管理ツールを検討し、開発を行った。

ネットワーク設計に必要なデータ群の分析と、試行開発したツールで利用したデータベース管理システムについて述べる。

また、今後の開発にあたり考慮すべき点についてまとめたものである。

2. ネットワーク設計支援システムにおけるデータベース構築について

ネットワーク設計支援システムで管理しなければならない情報とは、大きく分け以下のように分類される。

(1) ネットワークシステムを構成する機器の情報

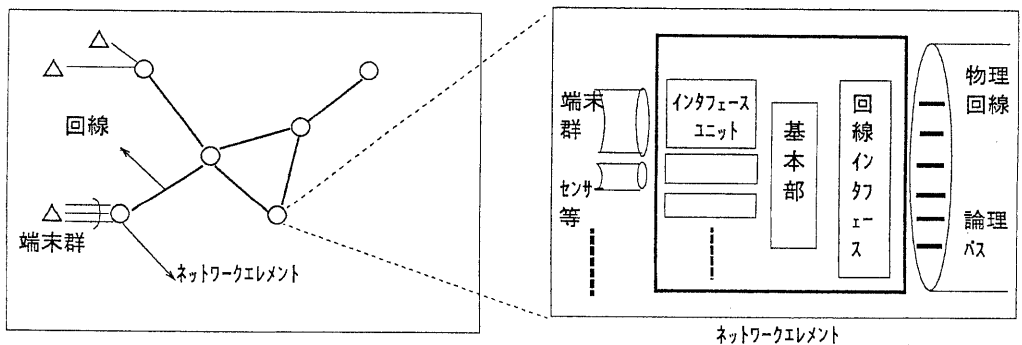
ネットワークシステムを構成する機器は1つ1つの内部情報を管理する必要がある。

各装置を分析すると、ユニット化を採用していることが多いことがわかる。

(2) 回線との接続情報

ネットワークシステムは、装置間を回線で接続されており、物理的な接続と別に論理的なパスで結ばれている。

これらは、ビル管理システムのようなセンサーベースを中心にしたネットワークシステムや企業内ネットワークのような広域なネットワークでも同じようにとらえることができる。



3. 試行開発に利用したデータベース管理システム

システムを構築するにあたり、膨大なデータ量、利用部門の多さ、運用管理システムの柔軟性を考慮に入れ、運用機器として富士通の大型計算機Mシリーズを採用した。

Mシリーズで利用できる代表的なデータベース管理システムには、ネットワーク構造型のデータベースシステム"AIM/DB"と関係モデルに基づく関係データベースシステム"AIM/RDB"の2つがある。

3.1 AIM/DBを利用した背景

先に述べたように、殆どのシステムに用いる装置の構成はユニット化方式を採用しており、データベースに格納する情報も構造型の記述が適当であると判断し、ネットワーク構造型を採用することとした。(装置は階層的に表現できる。)

AIMのデータベース管理は、次のような特長をもっている。

- (1)データベースを、DD/D(データ辞書/ディレクトリ)に登録された論理構造、格納構造および物理構造の3つの構造で統一的に管理する。これにより、システムの柔軟性を持つことができる。
- (2)データベースの領域を管理する。
- (3)応用プログラムのデータベースアクセスを統一的に受け付け処理する。

*1 構造型データベース管理システムは、データの実体をレコードの概念でとらえたものであり、そのデータ間の関連を木構造、ネットワーク構造の方式で表現している。

関連付けの方法により、階層型木構造で表現するもの(IBMのIMSなど)とネットワーク構造で表現するCODASYL型と呼ばれるものがある。AIM/DBはネットワーク構造型である。

3.2 データベース構築

センサーの種類に対応すべきインタフェースユニットの種類や、各種メディアに対応すべきユニットなどのように、各種データには様々な要素がある。データベース構築の際にどの要素をどこに格納するかを分析する必要がある。

AIMデータベースでは、これらの要素を分類し、次に示す4種類のとらえ方をすることによって、データベースの管理と実現性とを検証しようとした。

- ①論理構造：データの論理的な関係からとらえる。
- ②格納構造：データの論理的な配置からとらえる。
- ③物理構造：データの物理的な配置からとらえる。
- ④仮想論理構造：プログラムからみたデータの論理的な構造のとらえ方を定義する。

データベース管理システムは、この構造間を対応づけることによって応用プログラムからのデータベースへのアクセスを受付、必要なデータベースレコードへのアクセスを行う。

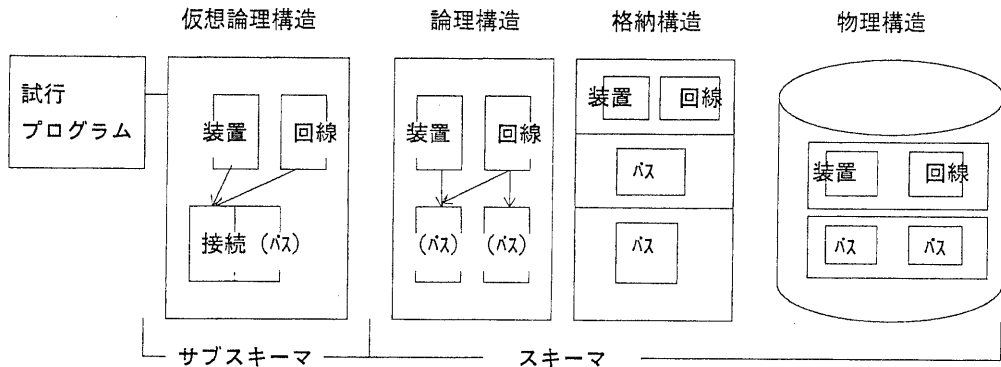


図1 構造間の関連

(1)論理構造

ADL (AIM Description Language) 中の SCHEMA コマンドによって定義する。

- ①レコード：COBOLのレコードにあたり，1つ以上のデータ項目から構成する。
- ②レコードタイプ：同一レコード属性をもつレコードの集まりを識別する単位。
- ③階層関係（セットタイプ）：2つのレコードタイプ間の関連をつけるため階層関係を定義する。

(2)格納構造

論理構造と物理構造の中間に位置するものである。

- ①ページ：格納構造の最小単位（入出力動作の最小単位）
- ②論理ページ：1ないし16個までのページで構成される。あるページにレコードを追加するときに、格納スペースがない場合は、そのページの属する論理ページの他のページに格納する。
- ③レンジ：レコードタイプを記憶媒体に配置するにあたって、アクセス効率，格納効率，リム管理を行う手段としてレンジを使用する。

(3)物理構造

格納構造で定義したデータベースを実際の記憶媒体に割り当てる方法を定義する。

1つのデータベース（1つのスキーマに対応）は、複数のデータセットから構成する。物理構造の設定によって、領域拡張の柔軟性，格納効率の向上，アクセス時間の短縮を図る。

(4)仮想論理構造

論理構造の一部から構成され，応用プログラムが取り扱う範囲の仮想データベースである。

1つの論理構造から，応用プログラムの用途に応じて複数個定義できる。1つの仮想論理構造は複数個の論理構造にまたがって定義でき，この機能によって応用プログラムは，広い範囲のデータベースにまたがって処理することが可能である。

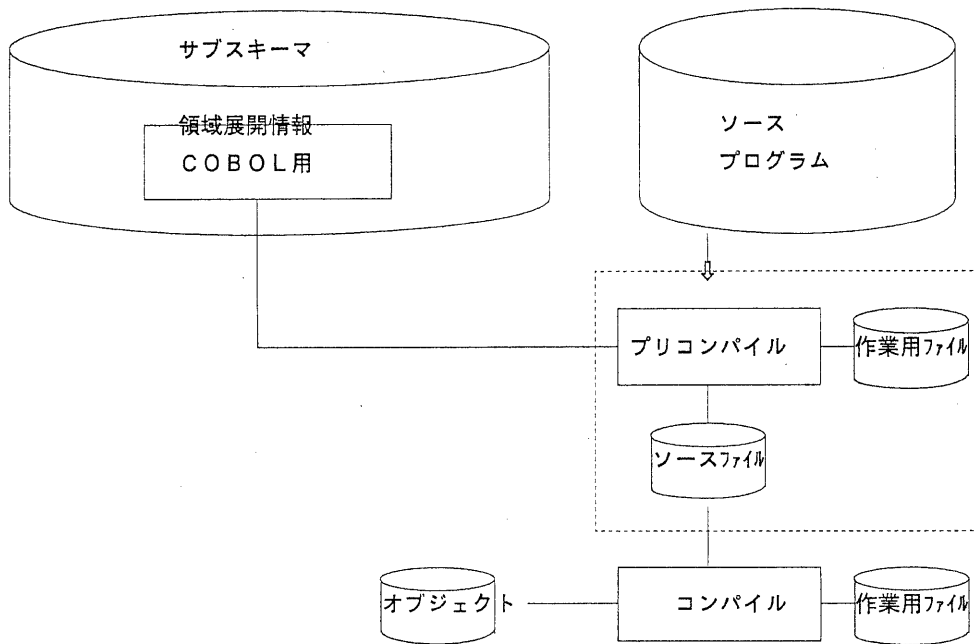
(5) 応用プログラムの動作環境定義

応用プログラムが、AIMシステム内でどのような条件下で動作するかを指定するため、応用プログラムの属性や動作環境の定義を行う。この定義体をプログラム環境定義体 (PED: Program Environment Description) と呼び、PED コマンドによって DD/D に登録する。その定義体は PED 名で識別され、AIM は PED を参照することによって応用プログラムの動作環境を設定し、その実行を制御する。PED で定義する属性の中で主な項目は、

- ① アクセスするデータベースの種スキーマグループ名
- ② 各種バッファの大きさ (ロギング、ページ、インデックスなど)
- ③ 処理する仮想データベースのサブスキーマ名

などがある。

応用プログラムを動作させるときには、そのジョブを起動するジョブ制御文に PED の名前を指定する。PED 用とデータベース用の DD 文で指定する。



4. 本システムの検証から得た課題

データベースシステムの良否はデータ内容の完成度で決まり、データ・モデリングの成否にかかっているといっても過言ではない。

データ・モデリングの目的は、できるだけ安定性の高いデータベースを生成することや、種々のユーザ要件に対応できるようにすることである。

データベースの内容は将来にわたり変化していく。その場合でも既存アプリケーション・プログラムをできるだけ修正しなくてもよいようにすることが、データベース設計の目的である。

今回のデータベースの検証に当たり、仮想モデルにしたセンサー集約装置や回線集約装置などの機器機能および外部ネットワークの機能などは目ざましい進歩がある。

例えば、センサーの新種類が相次いで開発されたり、端末装置のISDN対応のようにインタフェースが多様化している。

データベース格納項目の変化でプログラムに影響がでないようにすべきであるが、現実的にはプログラムに依存する部分が多く、将来を見越したデータベースの構築には配慮すべき点が多々ある。

機能追加が大きい時は、既存システムの大幅な改造を避けるため、別のファイルやデータベースを作成することも机上で検討したが、この場合はデータベースの柔軟性の低下をもたらし、また保守費用もかさむことが予想される。

5. 開発する上での配慮

データ・モデリングの結果を検討することが必要であるが、特に将来性の考慮が重要である。初期設計時にはよい論理設計であっても、将来の要求を予見し対応しておかなければデータベースの設計は失敗である。

このため運用者、利用者、データベース開発者の間で、将来性を含めたデータ・モデリングについて充分検討しておく必要がある。

6. まとめ

今回の検証に採用したデータベースシステムは、第1世代データベースシステムだが、第2世代(RDBM)、第3世代(OODS)でもデータベースを採用する際の本質的な問題は変わらないと考えられる。

環境の変化に伴うソフトウェアの要求定義は非常に難しく、あいまいなものである。

初期設計時に完璧なものはいない。

このような本質的な問題にソフトウェアをいかに対応させるか今後検討していきたいと考えている。

参考文献

- [1] James Martin : データベース構築の理論と手法, 日経BP社
- [2] 富士通, FACOM AIM 解説書
- [3] 今井, 原田, 平木 : データベース構築の理論と実際, コロナ社