

加入者線路保守システムへのオブジェクト指向データベースの適用性

加藤康之 片岡康郎

NTTフィールドシステム研究開発センタ

加入者線路網は今後益々複雑な形態へと変貌しつつあり、線路設備の実態性を容易にかつ経済的に表現可能なデータベースが望まれている。本稿では加入者線路設備の世界をオブジェクト指向概念でもって再整理し、設備データクラスの基本概念としてデータベースの実態性、データベースの現状性およびデータベースの設備表現性が重要であることを明確にした。これら実態性、現状性、設備表現性についてはそれぞれイメージオブジェクトや履歴オブジェクト、ならびに独自のビューメソッドの実装によって実現し、データの操作、アプリケーション開発において有効であることを示した。

An Object-Oriented Database Application for Subscriber Outside Plants Operating System

Yasuyuki Kato and Yasuo Kataoka

NTT Telecommunication Field Systems R&D Center
162 Tokai, Naka-gun, Ibaraki 319-11, Japan

Abstract Several features of Object-Oriented Database for subscriber outside plants operating systems are observed in this report. Three categories which indicate substantiality, condition visibility, and display ability to improve applications for the operating systems are clarified. These categories are implemented with an image object, a history object, and an individual display method, and its usefulness is confirmed in the development of the application.

1. はじめに

加入者線路設備は架空線路など目にみえる設備以外にも地下管路設備などによって地域の隅々まで布設されており、その保守領域は国土面積のそれに匹敵する。また地下設備から架空までを考慮すればまさに3次元的な広がりを持っている。従来のデータベース^[1]で表現可能な領域はコスト的にごく限られたものとなっており、線路設備の建設、保守などの実作業においては、携わる人間の経験や知識、感覚がこれを補っている。設備運用、保守の省力化を目指してこれまでいくつかのデータベースの検討がなされてきたが^{[2]~[4]}、現場において作業が必要とする設備の実態性とはかなりの隔りがあり、抜本的な省力化や効率向上につながっていないのが現状である。一方ニーズの多様化、サービスの高度化に伴い、Passive Optical Subscriber Loops^[5]に代表されるように、今後加入者線路網は益々複雑な形態へと変貌しつつあり、線路設備の実態性を容易にかつ経済的に表現可能なデータベースが望まれている。

本稿では近年盛んに検討されているオブジェクト指向データベース^{[6]~[8]}の技術を導入すべく、加入者線路設備の世界をオブジェクト指向概念でもって再整理し、その適用性について述べる。

2. 線路設備の構成

2.1 線路設備の概要

加入者線路設備とは交換局舎内の主配線架から屋外に延びている地下とう道、管路等の土木設備および通信ケーブルなど全般の総称であり、その概要は図1のようになる。

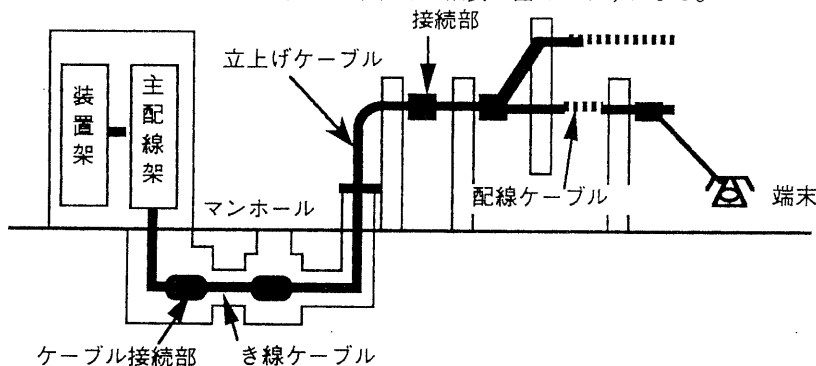


図1 加入者線路設備の構成

2.2 線路設備のレイヤ

土木設備および通信ケーブルに関わる設備および部品項目数は合わせて4千項目を優に越える。これらの設備はその機能から大別すると2つに分けられる。

- (1) 信号伝送媒体
- (2) 伝送媒体の保護設備

保護設備はケーブル等のように信号伝送媒体と切り離すことのできないものと、地下管路のように別工程で建設できる土木設備がある。従ってこれらは線路オペレーションシステムにおいても図2のように3レイヤに分類されている^[9]。各レイヤは以下のような特徴を有している。

(1) 媒体レイヤ

平衡対、同軸、光ファイバ、コネクタ、心線切替部品等からなり信号を伝送する機能を有する。

(2) ケーブルレイヤ

き線ケーブル、配線ケーブル、引込み線、屋内線、クロージャ等からなり伝送媒体を直接保護する機能を有する。

(3) 土木設備レイヤ

ケーブルの保護又は把持機能を有し、とう道、地下管路、電柱、つり線等からなる。

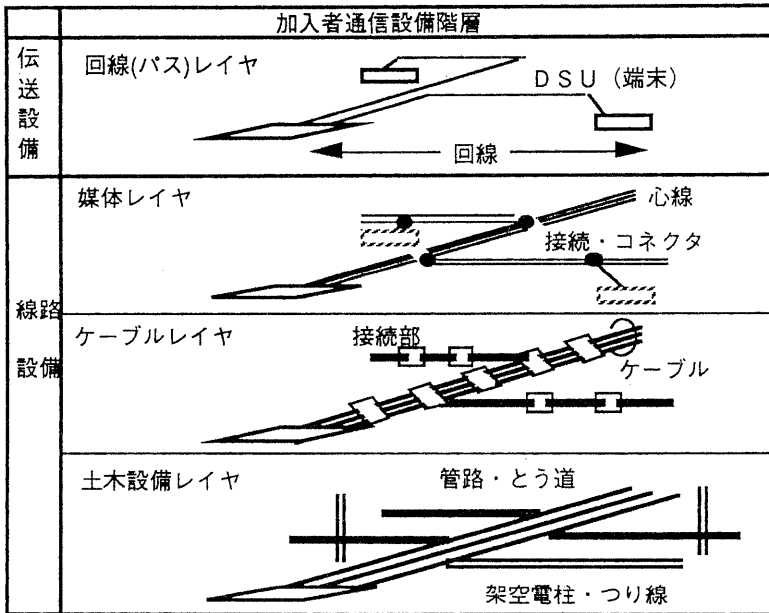


図2 加入者線路設備の階層構成

2.3 線路設備の動向

加入者線路網も光ファイバの導入によってその形態が複雑化して行く可能性がある。それは光ファイバ自体が多重化しやすい特性を有しているからであり、既に報告されているものを考慮すると^[10]、その移行時期においては図3のような複合形態が存在するものと思われる。すなわち局舎と1対1に結線する従来形態と波長多重によって伝送路の途中から1対多に分岐する形態である。従って端末の増設に対して分岐で対応する形態にデータベースが対応できるようにしておかなければならない。

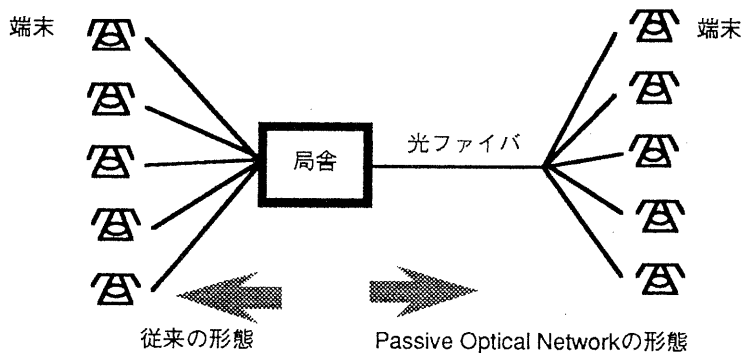


図3 メタル・光複合形加入者線路網の例

3. 線路設備データベースの基本的な条件

3.1 データの視点で捉えた線路設備業務

線路設備に関わる業務は設計（計画）、建設、保守（運用）に大別されるが、データという概念で捉えると図4に示すように、全てデータの移動と見ることもできる。すなわちデータベースは実体のないデータであり、線路設備は実体を持つデータと見るわけである。

〔設計〕ある機能を有する設備を想定して実体のない設備データを作成することを意味する。

〔建設〕作成された設計データから設備の実体を作成することを意味する。

〔保守〕設計データと設備の実体とを比較しデータの差異が常に最小になるように設備を保持すること意味する。

線路設備に関わる業務をあえてこのようにとらえる理由は、次節で述べるデータベースの条件をより明確にするためである。

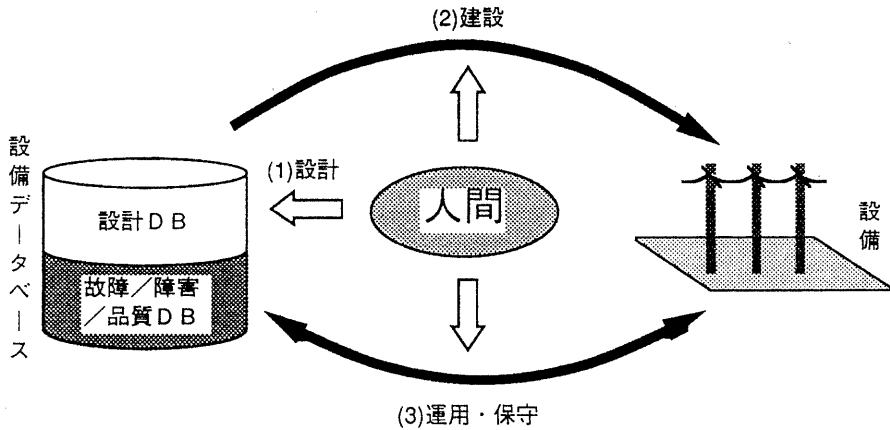


図4 データの概念で捉えた線路設備業務

3.2 データベースの実態性

線路保守者は、ほぼ一定の期間毎に巡回する形をとっているため、図4の(1)～(3)の業務を何度となく経験するのが一般的である。従って、関わる人間にとっては図の(1)～(3)のデータが常に同じように見えていることが重要である。つまりデータの振舞いやその存在が空間的つながりを持った実際の設備と同様に感じられるということである。ここではこれをデータベースの「実態性」と呼ぶ。

3.3 データベースの現状性

建設された設備は自然環境の中でわずかずつ変化するが、ある一定以上劣化するまで改修は行わないのが普通である。従って劣化予測や更改計画を効率よく行うためには現状の設備の品質や故障/障害の状態を当初の設計データの中に記録して行かなければならない。いかにデータベースが設備の現状および履歴を忠実に記録しているかの尺度を「現状性」と呼ぶ。

3.4 データベースの設備表現性

図4の(1)～(3)の業務の中には多くのアプリケーションが存在するため、業務毎のアプリケーションによらずデータベースの実態性を維持するためにはデータベース本来の機能によって設備が表現されなければ難しいと思われる。このようなビュー機能をデータベースが有することを「設備表現性」と呼ぶ。

4. データベースの基本設計

4.1 クラス設計のコンセプト

3章までの議論をふまえ線路設備データクラスの設計コンセプトを以下に示す。データクラス全体の設計においては、膨大な設備データを効率良く生成することを前提に継承性に配慮する。一方クラスデータベースの実態性についてはスーパークラス（線路設備クラス）で確保し、現状性、設備表現性などについてはサブクラスで実現することとする。まず2章で示した設備の階層化に基づいて基本データクラスを次のように定義する（図5）。

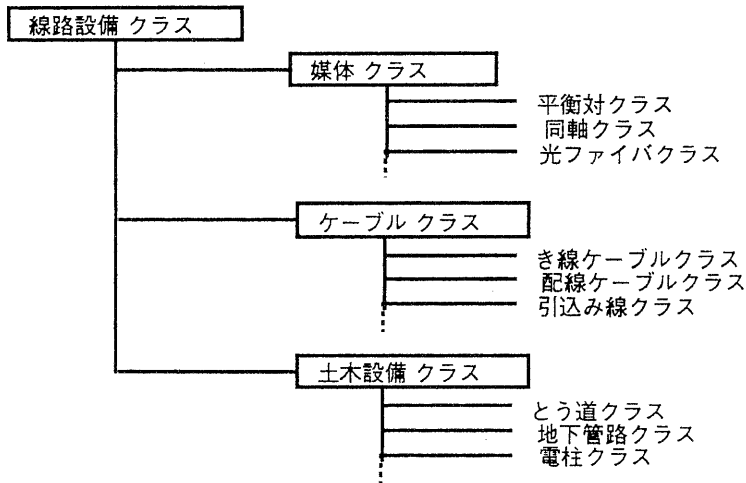


図5 線路設備のデータクラス

4.2 線路設備クラス

4.2.1 基本形

線路設備に共通の特徴は、全てのオブジェクトがなんらかの形でつながったノード形になっていることである。従って図6に示すようなノードの上位を示すIDと下位のノードリストを示すリスト形IDを持つパスノード形とする。これによって接続された管路やケーブル等の形態をそのままオブジェクトとして持つことが可能となり、データの実態性を確保することができる。

4.2.2 基本メソッド

線路設備クラスには図6に示すように4つの基本メソッドグループを付加している。

- (1) サブノード操作メソッド
サブノードの追加、削除、IDの獲得のためのメソッド
- (2) パス操作メソッド
新たなノードの追加、削除、IDの獲得のためのメソッド
- (3) インスタンス操作メソッド
・インスタンス変数のセットおよび獲得のためのメソッド
・付加的な属性を決めるオブジェクトの生成や変更メソッド等
- (4) オブジェクトアーカイブメソッド
オブジェクトの永続性を実現するためのメソッド

```

@interface SenroSetsubi:Object
{
    id      superNode;      // 上位ノード
    id      subNodeList;    // 下位ノードリスト
    id      attributeClassList; // 付加的な属性を示すIDリスト
    id      iconImage;      // ノードのアイコン表示用イメージID
    char    *nodeName;     // ノードの名前
    BOOL    leaf;          // サブノードの有無
}
// サブノード操作メソッド
-addSubNode:(char *)aNodeName;
. . .
//パス操作メソッド
-addNode:(char *)pathName;
. . .
//インスタンス操作メソッド
-setSuperNode:anObject;
-addAttributeClass:newAttributeClass
. . .
//オブジェクトアーカイブメソッド
-read:(NXTypedStream *)typedStream;
-write:(NXTypedStream *)typedStream;

```

図 6 線路設備クラスの定義例 (objective-Cを使用)

4.2.3 オブジェクトの永続性

オブジェクトをストリームに格納したり、アプリケーション間通信データとしてオブジェクトデータを送受するために用意したメソッドであり、図7に示すような「read」、「write」の一对のメソッドからなっている。双方ともストリームに対して型付でインスタンス変数の書込み、読込みを行っている。全てのクラスにこのメソッドを用意することによって、ノードの最上位のオブジェクトに「write」メッセージを送れば再帰的にストリームに書き込むことができる。

```

-write:(NXTypedStream *)typedStream;
{
    [super write:typedStream];           //スーパークラスの保存
    NXWriteObjectReference(typedStream,superNode); //上位ノードの保存
    NXWriteObject(typedStream,subNodeList); //下位ノードリストの保存
    . . . . .                          //必要なインスタンスの保存
    return self;
}

-read:(NXTypedStream *)typedStream;
{
    [super read:typedStream];           //スーパークラスの読込
    superNode =NXReadObject(typedStream); //上位ノードの読込
    subNodeList =NXReadObject(typedStream); //下位ノードリストの読込
    . . . . .                          //必要なインスタンスの読込
    return self;
}

```

図 7 オブジェクトアーカイブメソッドの例 (objective-Cを使用)

4.3 線路設備サブクラス

図5の媒体クラス以下を線路設備サブクラスとする。線路設備サブクラスでは設備の属性をより明確に表すためのインスタンスを記述しているが、さらにデータの現状性や設備表現性を確保するためのインスタンスリストやビューメソッドを付加している。ケーブルクラスを例にしたものを図8に示す。オブジェクト自身がインスタンスの表示やそのエディット機能を標準装備するために、viewWindowおよびviewInterfaceという2つのインスタンスIDとopenView:senderというメソッドを付加している。それぞれこれらはビュースクリーンをオープンするためのウィンドウIDとメソッドが呼ばれた時にロードすべきインターフェイスファイルのIDを示している。ビューのオープンメソッドは次のようになっており、

```
- openView:sender
{
    viewInterface=[NXApp loadNibSection:"CableView.nib" owner:self];
    . . .
    [[viewWindow display] makeKeyAndOrderFront:self]; //ウィンドウの表示
    return self;
}
```

インターフェイスファイル (CableView.nib) を動的にロードすることにより、データベース中にビューウィンドウの実体が入るのを避けている。

```
@interface Cable:SenroSetsubi
{
    char    *cableName;    // ケーブルの名前
    char    *date;        // 製造年月日
    char    *maker;       // 製造会社名
    int     length;       // ケーブル長
    . . . . .
    id     viewWindow     // ビュースクリーンのID
    id     viewInterface  // ビュースクリーンのユーザー
                        // インターフェイスID
}

// インスタンス設定およびクエリメソッド
- setCableName:(char *)aCableName;
- (char *)cableName;
. . .

// 履歴・現状態設定およびクエリメソッド
- addDiagResults:(void *)results;
- (void *)results;
. . .

// ビューメソッド
- openView:sender;

// オブジェクトアーカイブメソッド
- read:(NXTypedStream *)typedStream;
- write:(NXTypedStream *)typedStream;
```

図8 ケーブルクラスの定義例 (objective-Cを使用)

5. システムの構成および評価

5.1 システム概要

現在開発中のオブジェクト指向データベースを用いた線路設備オペレーションシステムの概要を図9に示す。システム全体の記述はobjective-Cを用いている。アプリケーションからデータベースにアクセスするためには、アプリケーションにデータベースアクセスモジュール (DBmodule) をインクルードしておき、DBmoduleに対して書き込みたいオブジェクトのIDまたは読み込みたいオブジェクト名を渡す。一方データベースの基本機能である一貫性制御や障害復旧等は評価用ということで簡略化している。

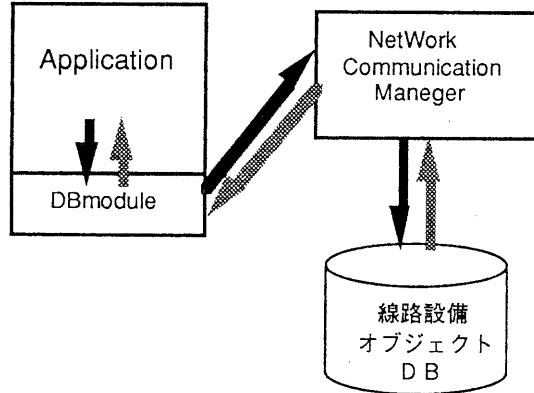


図9 システム概要

5.2 評価

データベースのパフォーマンスを定量的に評価するところまでは至っていないため、ここでは主としてアプリケーションの開発効率や3章で議論したデータベースの実態性等の観点から評価を進める。DBmoduleからのデータベースへのアクセスはネットワークコミュニケーションマネージャを通じて行う。データベースのオープン、クローズ、ネットワークへのアクセス手順などはDBmoduleが全て受け持ちこれらを隠ぺいしている。またオブジェクトは4.2.3節で述べた方法によりオブジェクトイメージをそのままストリームに変換する操作もDBmoduleが行うため、アプリケーションはデータベース上のオブジェクトとアプリケーション固有のオブジェクトとの違いを意識する必要はない。従ってアプリケーションからのデータベースへのアクセスは、プログラミング上もまた実行上もきわめて効率的であることが確認された。

一方データベースの実態性については、オブジェクト間の構成そのものを実際の線路設備構成と同じように構築できるため、心線の接続替え等においては現場の作業形態と同様の操作でデータベースを変更できる。さらにオブジェクト自身が設備のイメージを持っているため図10に示すようにブラウザ上に設備のつながり関係と同時にイメージを表示することができる。

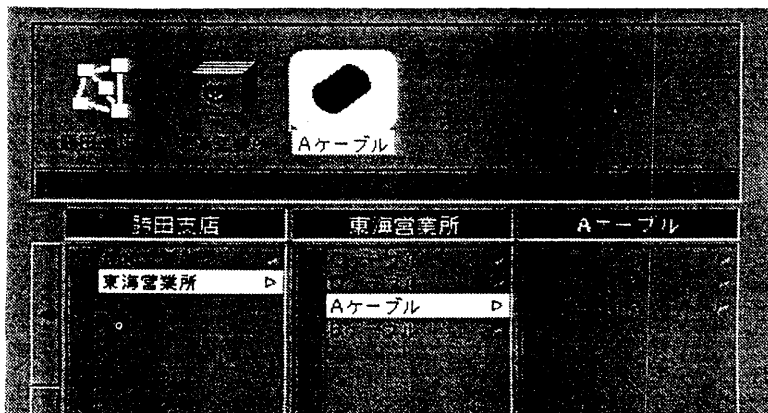


図10 設備オブジェクトのブラウザ表示の例

データベースの設備表現性については、各設備クラスが自身のビューを持つようにしたため、ブラウザ上で設備名をマウスでクリックするだけでそのクラスに適したビューを開くことが可能になっている。なお線路設備全体を表すブラウザについては、線路設備クラスに含めることも可能であるが、線路設備クラスをシンプルにする意味でここでは含めていない。アプリケーションではブラウザオブジェクトに次のように、

```
[browser cableObject] ;
```

設備オブジェクトを送る一行を付加するだけで表示が可能になっている。

データベースの現状性については、各サブクラスに任意フォーマット ((void *)results) のポインタを用意し、サブクラスの履歴を記録できるようにした。なお、任意フォーマットの妥当性については使用実績をみて判断する予定である。

6. おわりに

加入者線路設備の世界をオブジェクト指向概念でもって再整理し、設備データクラスの基本概念としてデータベースの実態性、データベースの現状性およびデータベースの設備表現性が重要であることを明確にした。これら実体性、現状性、設備表現性についてはそれぞれイメージオブジェクトや履歴オブジェクト、ならびに独自のビューメソッドの実装によって実現し、データの操作、アプリケーション開発において有効であることを示した。

本格的なオブジェクト指向形線路設備データベースを構築するためには、さらに下記に示すいくつかの課題を解決していかなければならない。

- ・現状性確保における表現形式の整理
- ・数千に上る設備クラス作成の自動化
- ・現状のSQLを主体としたアプリケーションへの対応
- ・一貫性制御などデータベースの機本機能の完備

[謝辞]

アプリケーションでの有効性評価に関して有意義なご意見、ご協力を頂いたNTTフィールドシステム研究開発センタの満永主幹研究員ならびに阿部主任研究員に感謝します。

[参考文献]

- [1] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, Vol.13, p.377, 1970.
- [2] C. M. Franklin and J. F. Vogler, "Data Base System", Bell Syst. Tech. J., Vol.61, p.1131, 1982.
- [3] 山本良一、小山田弥平、及川寿雄、前川英二、"加入者光線路網の構成法"、研究実用化報告、Vol.34, p.1099, 1985.
- [4] 伊藤弘吉、茨木茂、中増憲治、"電話局業務の総合的な支援を目指して"、施設、Vol.37, p.58, 1985.
- [5] J. R. Stern, J. W. Ballance, D. W. Faulkner, S. Hornung, D. B. Payne, and K. Oakley, "Passive optical local networks for telephony applications and beyond", Electron. Lett., Vol.23, p.1255, 1987.
- [6] G. Copeland and D. Maier, "Making Smalltalk a Database System", Proceedings of the ACM SIGMOD International Conference on Management of Data, p.316, 1984.
- [7] M. Stonebraker, B. Rubenstein, and A. Guttman, "Application of Abstract Data Types and Abstract Indices to CAD Data Bases", Proceedings of Engineering Design Applications, Database Week San Jose -May 23-26, p.107, 1983.
- [8] 増永良文、"次世代データベースシステムとしてのオブジェクトデータベースシステム"、情報処理、Vol.32, p.490, 1991.
- [9] 満永豊、古賀広昭、"加入者線路保守OpSの特徴とAI技術の適用"、信学技報 IN91-5, 1991.
- [10] Y. M. Lin and D. R. Spears, "Passive Optical Subscriber Loops with Multiaccess", J. Lightwave Technol. Vol.7, p.1769, 1989.