

# オブジェクト指向データベースによる イントラネットデータベースシステムの構築

安村 義孝

NEC C&C 研究所

〒 216 川崎市宮前区宮崎 4-1-1

tel: 044-856-2133

e-mail: yasumura@swl.cl.nec.co.jp

オブジェクト指向データベースを利用したイントラネットシステムの構築事例におけるシステム構成法について述べる。インターネットの WWW 上のコンテンツや電子メール、ニュース記事などのリソースデータをオブジェクト指向でモデル化し、それらのデータを関連づけてオブジェクト指向データベースに格納する。格納されたデータは WWW-OODB 連携システムを用いてユーザにサービスする。WWW-OODB 連携システムでは、データベースに格納された複合オブジェクトを、その構造を意識して HTML 文書に変換するための方式を提供する。このようなアプローチを採ることにより、より柔軟な機能を持つイントラネットデータベースシステムの構築が容易になる。

## An Implementation of Intranet Database System using Object-Oriented Databases

Yoshitaka Yasumura

C&C Research Laboratories, NEC Corporation

4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216

tel: +81-44-856-2133

e-mail: yasumura@swl.cl.nec.co.jp

In this paper, a system construction method for constructing an intranet system example using object-oriented databases is described. Resources (WWW contents on the internet, electronic mails, and news items) are modeled by the object-oriented model. These data are connected each other, and stored in object-oriented databases. Stored data in the databases can be served with users using a WWW-OODB integration system. In the WWW-OODB integration system, the converting method that converts composite objects into HTML text based on its structure is provided. As a result of these approaches, it is easy to construct intranet database systems that have flexible functions.

## 1 はじめに

WWW (World Wide Web) や電子メールなどのインターネット技術を活用して社内のイントラネットシステムの構築が盛んに行われている。インターネット技術は近年盛んに研究開発が進められている分野であり、技術動向も目まぐるしく変化している。そのため、Java[1] や ActiveX[2] など利用可能な要素技術が次第に揃い、イントラネットシステムの基盤技術としてはかなり成熟しつつある。イントラネットシステムでは、特許情報や報告書などの社内文書を HTML (HyperText Markup Language) 文書化して汎用の WWW ブラウザで閲覧可能にしたり、電子メールにエージェント機能を付与して業務ワークフローに応用したりして、積極的に工夫がなされている。既に一部のシステムは製品化されているものがある。これらのシステムは扱うリソースが多様になるため、データベース管理システムの支援が必要になってくる。

イントラネットシステムでデータベース管理システムを利用する場合、一般的には WWW-DB 連携システムが利用されることが多い [5]。通常、WWW-DB 連携システムは関係データベース管理システムをバックエンドに持ち、ユーザから指定された条件に基づいてデータベース内の検索処理を行い、その検索結果として得られたあるテーブルのタブルの属性データを、何らかの方法で変換して HTML 文書に埋め込む必要がある [4]。各属性データ毎にアプリケーション開発者が変換規則を決めていく形式になっている。したがって、データベース内のデータ構造とは全く独立にそれらの変換規則を決めなければならない。そこで、オブジェクト指向データベースと WWW 技術を融合させる研究も行われている [?]。

本稿では、イントラネットシステムで必要とされるリソースをオブジェクト指向によりモデル化し、それらのデータをオブジェクト指向データベース管理システム PERCIO[8] が管理するデータベースに格納して、ユーザにサービスするシステムの構築方法について述べる。オブジェクト指向データベースの複合オブジェクトによりリソースを表現し、それらに関連を付与してデータベースに格納する。データベース内の複合オブジェクトを自由に組み替えることで、リソースの加工・修正を行う。これらのリソースをユーザに提供するために、WWW-OODB 連携システムを利用する。この WWW-OODB 連携システムでは複合オブジェクトをそのデータ構造を意識して変換規則を与えることができる仕組みを持つ。

## 2 イントラネットデータベースシステム

イントラネットデータベースシステムとはインターネット上の電子メールやニュース、WWW コンテンツなどの様々なリソースをデータベースに格納して一元管理し、それらのリソースデータを活用して部門内の情報システムを構築したシステムと考える (図 1)。このデータベースが存在しているのがサーバのマシン、

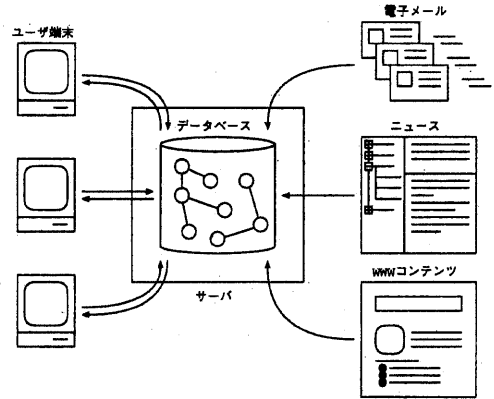


図 1: イントラネットデータベースシステム

ユーザが実際にネットワークを介してそのデータベースにアクセスするのがクライアントのマシンである。サーバ側にはデータベース管理システムが存在しており、データベース内のリソースデータを HTTP (HyperText Transfer Protocol) サーバを経由して要求を出したクライアント側の WWW ブラウザに HTML 文書として送信する。

各種のリソースはそれぞれに規格化されているプロトコルを通して取得することが可能であり、それぞれに規格化されている文書 (コンテンツ) フォーマットを持っている。これらの異なるリソースを一元管理するためには、異なる種類のリソースの統一されたモデル化が必要となる。さらに、各リソース間の関連付けや、複数のリソースの部品を組み合わせるなどの工夫も要求される。

データベースに格納されているリソースデータをユーザに見える形に提供するために、テキスト形式に変換しなければならない。そのためには、HTTP サーバが持つ CGI (Common Gateway Interface) の機能を利用することが多い。ユーザから与えられた情報に基づいて、あらかじめ CGI スクリプトに埋め込まれているデータベースアクセスのためのコードを実行し、データベース管理システムから返却されたデータを HTML 文書形式に変換してクライアントに返答するという利用形態になる。しかし、CGI スクリプトを記述することは容易ではない。まして、複雑に関連付けられたリソースデータを変換するためのコーディングは困難な作業になる。

ここでは、イントラネットデータベースシステムの例題としてプロジェクト管理システムを考える (図 2)。ソフトウェア開発時のプロジェクト管理を想定し、管理者と設計者、プログラマが様々な情報を共有している。設計者が記述した仕様書に基づいてプログラマが開発を行い、その進捗状況を報告する。検査時にはバグ報告も随時行われることになる。管理者は設計者とプログラマの進捗管理を行い、工数の見積もりやスケジュールの変更などを行う。また、他社状況や社外の

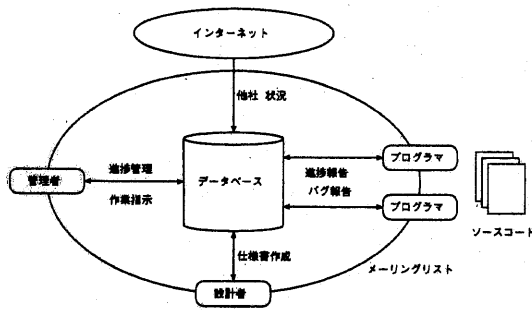


図 2: 例題: プロジェクト管理システム

状況も WWW などを利用して把握する必要がある。これらの情報をすべてデータベースに格納し、各人が任意の時間に参照したり、更新したりすることができる仕組みが必要となる。

### 3 オブジェクト指向によるモデル化

#### 3.1 リソースモデル

前節で述べたように、様々なリソースをオブジェクト指向データベースに格納し、それらのリソースデータを統合的に管理するためにはモデル化が重要になる。しかし、オブジェクト指向に限るものではないが、情報のモデル化は一意に決まるものではないため、システム自体にある程度の汎用性を持っていなければならない。

本稿で扱うプロジェクト管理システムで必要とされるリソースとその属性としては以下のものが考えられる。

##### 1. 電子メール

- フォルダ
- 送信者、受信者、送信日時、受信日時
- タイトル
- テキスト本文
- 添付ファイル

##### 2. ニュース

- ニュースグループ
- 投稿者、投稿日時
- テキスト本文
- 添付ファイル

##### 3. WWW コンテンツ

- ディレクトリ (URL)
- HTML 文書
- 画像

- その他 (アプレットなど)

#### 4. 仕様書

- 仕様書番号
- テキスト本文
- 図面

これらのリソースをモデル化した一例を図3に示す。図中で、電子メールフォルダ階層とニュースグループ階層、HTMLディレクトリ階層の階層化されたノードから各リソースデータに参照を持ち、リソースデータ同士にも関連付けが与与できるようなモデルになっている。

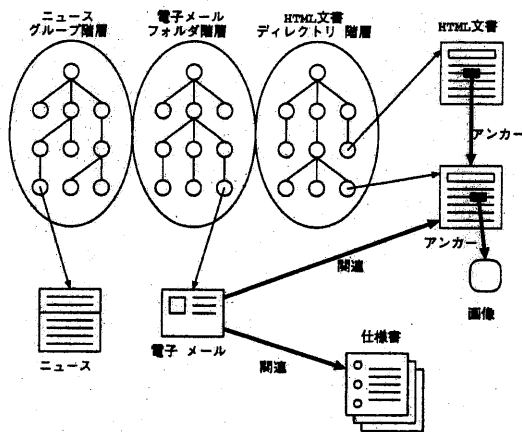


図 3: リソースモデル

#### 3.2 リソース間の関連

リソース間には任意の関連付けができるようになる必要がある。例えば、ある電子メールがある WWW コンテンツに関するものであれば、電子メールのリソースから WWW コンテンツのリソースへの関連が付けられる。また、WWW コンテンツなどは頻繁に内容が更新されるので、それらのバージョン管理が必要になってくるが、これもリソース間の関連で解決することが可能である。本モデル化では、リソース間の関連を関連リンクというオブジェクトで表現し、そのオブジェクトに関連を辿るための属性や操作を定義する。

ここで、ノード A とノード B が関連リンクにより関連付けられ、ノード B は複数のバージョンを持つ場合を考える (図4)。関連リンクが持つカレントバージョンがノード B1 であるが、ノード A が要求するバージョンがノード B2 となると、次のような手順でリソース間の関連を遷移する。

1. ノード A から関連リンクに辿り、特定のメソッドを起動する。

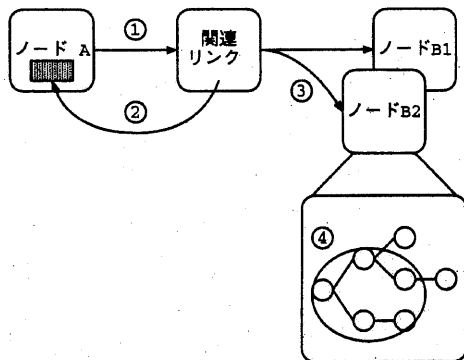


図 4: リソース間の遷移

2. 起動されたメソッドからノード A の情報を取得する。
3. 取得した情報に基づき、ノード B への参照をノード B1 からノード B2 に張り替える。
4. ノード B2 へ遷移し、必要なデータを取得する。

したがって、ユーザ側にリソースデータを提供する際には、以上のような操作を行えるような仕組みが必要となる。

## 4 WWW-OODB 連携システム

### 4.1 システム構成

WWW-DB 連携システムは通常 CGI やサーバ API などの HTTP サーバで提供されるゲートウェイインタフェースを利用して実装されている。これらのシステムは HTTP サーバとデータベース管理システムの間にあるゲートウェイシステムである。我々が開発している WWW-OODB 連携システム [10] もこれらのシステムの 1 つであり、システム構成は図 5 に示す通りである。

本システムは 3 つのコンポーネントから構成される。

- ゲートウェイプログラム

HTTP サーバから CGI 経由で起動されて呼び出されるプロセスであり、データベースサーバとの間のやり取りを行う。それ以外の特別な機能はない。HTTP サーバとデータベースサーバがサーバ API で接続されている場合は、このコンポーネントは不要である。

- データベースサーバ

WWW-OODB 連携システムのメインサーバであり、CGI またはサーバ API を通して HTTP サーバと連携して動作する。内部ではデータベースアクセスのセッション管理部や、取得したデータを利用して HTML 文書を組み立てる HTML 文書生成部などを持ち、クライアントから要求された処

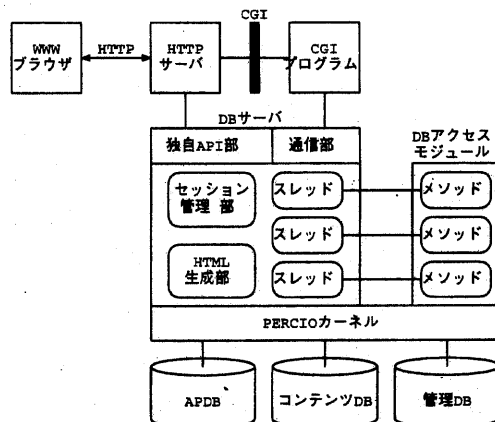


図 5: WWW-OODB 連携システム

理を実行するためのスレッドがここに生成される。実際のデータベースへのアクセスはデータベースアクセスモジュール内のメソッドが行うため、各スレッドはそれらのメソッドと動的にリンクされて処理が進められる。

- データベースアクセスモジュール

実際のデータベースへのアクセスを行うコンポーネントであり、ActiveX コントローラ群からなる。データベースサーバが ActiveX コンテナとなり、動的に ActiveX コントローラであるメソッドを呼び出して、データベースアクセス処理が行われる。したがって、ActiveX 技術がサポートしている様々な方法 (プログラミング言語) で、データベースアクセスモジュール内のメソッドが記述可能である。

本システムでアクセスされるデータベースにはアプリケーションデータベースとコンテンツデータベース、管理データベースの 3 種類がある。アプリケーションデータベースは各アプリケーションに依存したデータを格納するデータベースであり、プロジェクト管理システムでは統合されたリソースデータが格納される。コンテンツデータベースはアプリケーションデータベース内のデータをクライアント側の WWW ブラウザに表示するための HTML テンプレートのデータが格納されている。また、管理データベースはユーザ情報やセッション情報などのデータベースサーバの管理用データが格納されている。

### 4.2 データベースアクセス処理

WWW アプリケーションはデータベースアクセスモジュールのメソッドを通してデータベースにアクセスする。ユーザが URL のパラメータとしてメソッドを指定する。このパラメータは HTTP サーバに受けとられ、そのままデータベースサーバに渡される。データベースサーバは要求されたデータベースアクセスを処

理するスレッドを生成する。それらのメソッドを含んだ共有ライブラリが動的にデータベースサーバと ActiveX 技術によりリンクし、メソッドが呼び出される。データベースアクセス処理が終了した後も、HTML 文書生成時に追加データのアクセス要求が出されることもある。

データベースアクセスモジュールのメソッドは PER-CIO がサポートしている問合せ言語やプログラミングで記述可能である。データベースサーバとデータベースアクセスモジュールの間のインタフェースは ActiveX 技術による API である。したがって、データベースアクセスモジュールのメソッドが ActiveX コントロールで、データベースサーバが ActiveX コンテナという位置付けになる。メソッド名とメソッドインタフェースは参照テーブルで管理され、その対応関係はセッション終了時まで維持される。

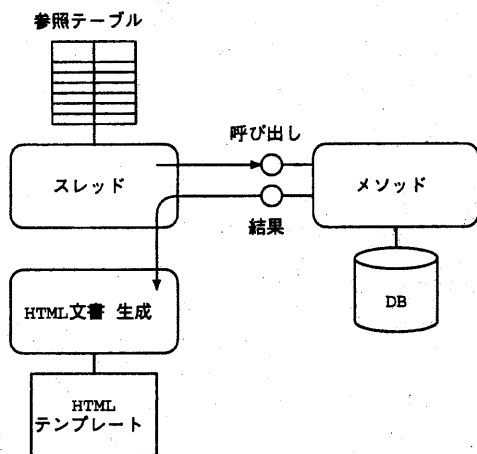


図 6: データベースアクセス処理

データベースアクセスの結果はコレクションまたはオブジェクトへの参照として利用される。HTML テンプレートでは、それらのオブジェクト参照は結果のデータを埋め込む場所に記述される。HTML テンプレートで利用されるオブジェクト変数名とそれに対応するオブジェクト変数値は、データベースアクセスモジュールのメソッドに記述されている。この対応関係も参照テーブルによって管理されて、HTML 文書生成時に任意に参照される。このとき、オブジェクトの実体はデータベース管理システムが管理するオブジェクトキャッシュに存在するが、これはデータベースアクセスのメソッドが実行した時のものをそのまま残しておく必要がある。

オブジェクト変数に束縛された変数はデータベースアクセスのためのメソッドの ActiveX コントローラのプロパティである。これらの値はオブジェクト参照から返却される変数や関数などに割り当てられる。さらに、HTML テンプレートに HTML テキストを埋め込むことを可能にするためのインタフェースもライブラ

リとして提供することができる。

### 4.3 複合オブジェクトの変換

HTML テンプレートの仕様は、HTTP サーバである IIS (Internet Information Server)[3] に搭載されている IDC (Internet Database Connectivity) で規定されている HTML エクステンションファイルに基づいて、キーワードや制御構文の拡張を行っている。HTML 文書生成はこれらのキーワードや制御構文により出力形式が決まる。さらに、HTML テンプレートの仕様はオブジェクト参照やコレクションをサポートするために拡張されている。

各オブジェクトアクセスは HTML テンプレートの拡張タグで指定される。HTML 文書が生成される時は、これらの拡張タグがデータベース中のデータ値に変換されて埋め込まれる。このデータ値が埋め込まれる場所は `<$begindetail$>` と `<$enddetail$>` で囲まれているなければならない。オブジェクト変数と制御構文は HTML テンプレートのこの部分に記述される。オブジェクト変数はオブジェクトまたはコレクションへの参照なので、文字列や数値を持つメンバ変数はこのオブジェクト変数から取得することが期待される。そこで、ドット表現を導入する。オブジェクト変数に束縛されているオブジェクトのメンバ変数にアクセスする場合は、次のように HTML テンプレート内に記述する。

```
<$ (オブジェクト変数名).(メンバ変数名) $>
```

このメンバ変数の値は文字列に変換されてから出力される HTML 文書に埋め込まれる。

また、コレクションに束縛されたオブジェクト変数の場合は次の制御構文を利用する。

```
<$ foreach (変数名) in (コレクション名) $>
<$ do [ (条件) ] $>
    (HTML テキスト)...
<$ done $>
```

上記の構文において、(オブジェクト変数名) は任意の名前、(コレクション名) はコレクションに束縛されたオブジェクト変数名、(条件) は“(value1) (operator) (value2)”または“until (number)”の形式である。指定されたコレクションの各要素は 1 つずつ宣言された変数に割り当てられ、`<$ do $>` と `<$ done $>` で囲まれた場所で他の変数と同様に扱うことができるようになる。さらに、条件が指定された時には、コレクション内の要素を処理するかを区別することが可能である。

この様子を図 7 に示す。この図にあるように `foreach` 文はネストすることができ、深さ優先でそれらの複合オブジェクトへのアクセスに適用される。

## 5 考察

本稿で述べたイントラネットデータベースシステムでは、次のような特徴がある。

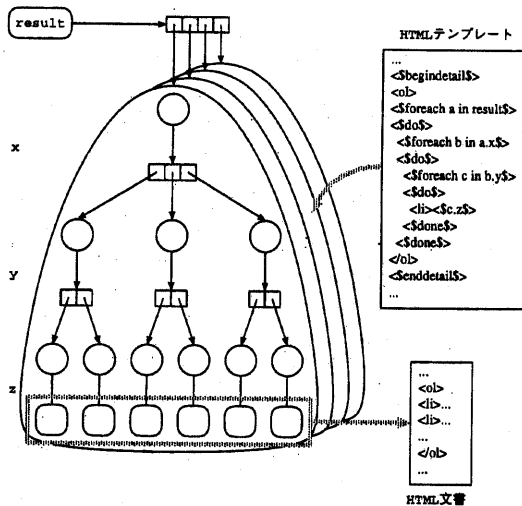


図 7: 複合オブジェクトの変換

- 異なる種類のリソースをオブジェクト指向モデルでモデル化し、オブジェクト指向データベースに格納しているため、リソースの統合的な管理が可能になる。また、リソース間の関連付けによりバージョン管理などが容易に実現できるようになる。
- オブジェクト指向データベースの複合オブジェクトの機能により、リソースの内部データにまで構造化を行うことができ、内部データの部品化によるリソースの加工・修正が容易になる。
- オブジェクト指向データベースに格納されたデータを WWW ブラウザに表示するために、WWW-OODB 連携システムにより HTML 文書にデータを埋め込む必要があるが、複合オブジェクトの構造をそのまま活かして展開することが可能である。

以上の特徴により、従来の関係データベースを利用したシステムよりも、機能が豊富で開発しやすいシステムになると考えられる。

また、WWW-OODB 連携システムとしては従来からある製品も存在するが、オブジェクト指向データベースの機能をフルに活用していないものがほとんどである。例えば、O<sub>2</sub>Web[7] はオブジェクト指向データベース管理システム O<sub>2</sub> と WWW を連携させるシステムであり、データベースクラスが HTML 文書を生成するためのテンプレートとなる特徴を持つ。つまり、クラス定義が WWW ページのテンプレート定義となり、そのクラスのインスタンスが WWW ページに対応することになる。インスタンス間の参照関係が WWW ページ間のリンク関係となり、その WWW ページにリンクが OQL 文として埋め込まれる。しかし、変換規則を変えるためには、その変換規則を O<sub>2</sub>C と呼ばれるプログラミング言語で記述しなければならない。

一方、ObjectForms[6] はオブジェクト指向データベース管理システム ObjectStore と WWW を連携させるシステムであり、そのシステム構成は本稿で述べた WWW-OODB 連携システムと同様に、データベースにアクセスするためのユーザ定義関数を作成することができる。ただし、HTML テンプレート内で利用可能なデータはそれらの関数の返却値のみであり、任意の複合オブジェクトへのアクセスを HTML テンプレート内で表現することができない。

## 6 おわりに

本稿では、イントラネットデータベースシステムの構築方法としてオブジェクト指向データベースを利用したアプローチについて述べた。オブジェクト指向データベースにより様々な種類のリソースを統合的に管理でき、それらの関連付けや部品化されたりソースデータの加工・修正が容易に行える。また、WWW-OODB 連携システムによりそれらのデータを WWW ブラウザに表示可能な HTML 文書の形式に変換するが、複合オブジェクトの構造を活かした変換方法を提供することで、開発作業が容易に行える。

今後の課題としては、Java や ActiveX とも連携させて CORBA のような分散オブジェクト環境でオブジェクト指向データベースを活用したイントラネットデータベースシステムの構築方法も考えていく予定である。

## 参考文献

- [1] Arnold, K. and Gosling, J., *The Java Programming Language*, Addison Wesley, 1996.
- [2] Chappell, D., *Understanding ActiveX and OLE*, Microsoft Press, 1996.
- [3] Microsoft Corp., *Internet Information Server*, <http://www.microsoft.com/iis/>.
- [4] Nguyen, T. and Srinivasan, V., "Accessing Relational Databases from the World Wide Web," *Proceedings of 1996 ACM SIGMOD*, pp. 529-540, 1996.
- [5] 日経データプロ, WWW-データベース連携システム構築法, 日経 BP, 1996.
- [6] Object Design Inc., *ObjectForms*, <http://www.odi.com/>.
- [7] O<sub>2</sub> Technologies, *O<sub>2</sub> Web*, <http://www.o2tech.fr/>.
- [8] 鶴岡邦敏, 木村裕, 波内みさ, 安村義孝, "オブジェクト指向データベース管理システム PERCIO の開発と今後の課題," 電子情報通信学会論文誌, Vol. J79-D-I, No. 10, pp. 587-596, 1996.
- [9] Yang, J.J. and Kaiser, G., "An Architecture for Integrating OODBs with WWW," *Fifth International World Wide Web Conference*, 1996, [http://www5conf.inria.fr/fich\\_html/papers/P31/Overview.html](http://www5conf.inria.fr/fich_html/papers/P31/Overview.html).
- [10] Yasumura, Y. "WWW Application Development Framework using Composite Objects," *Sixth International World Wide Web Conference*, 1997, <http://poster.www6conf.org/poster/719/poster719.html>.