

3 J-1 オフィス業務構築環境
- オブジェクト指向フォーム管理方式 -

山崎 圭子* 若林 宏**

日本電気(株)

*C&C システム研究所, **C&C システム技術本部

1 はじめに

オフィス業務を支援するシステムとして、マルチメディアデータを扱い、ユーザインタフェースを重視したシステムが望まれている。オフィス業務は業務の変化に応じて柔軟に対応できる必要があり、単一または極少数の部門内で閉じたシステムのため、システム化が遅れていた。しかし、近年の高性能端末・WSの普及とダウンサイジングの機運に伴い、オフィス業務を支援するシステムを導入し、他社との競争力に差をつけようとする方向にある。

このようなシステムの開発方法としては、高性能な端末やWS上でのGUIアプリケーション開発ツールとDBサーバとの連携による方法があるが、真の開発環境が提供されているわけではない。

一方で、従来のデータベース(DB)で扱いにくかったマルチメディアデータをオブジェクト指向データベース(OODB)で容易に扱えるようになってきた。

このような状況から、OODBを用いてオフィス業務を構築する環境を検討・試作した。本稿では、試作したオブジェクト指向フォーム管理方式について述べる。

2 システムモデル

オフィス業務の中でも、共用データを対話的に入力、参照、更新することを行う業務を対象とし、対話のためにGUIを扱うことを前提とした。業務システム自体の構築は以下の3つの処理に分離される。

- ユーザインタフェース(UI)処理: 対話画面の出力、入力制御、入力を契機に業務処理起動
- DBアクセス処理: DBからの検索、DBへの登録・更新・削除
- 業務ロジック処理: 業務本来のロジック

従来のシステム構築では、上記処理を一体化して業務分析し構築してきた。本システムでは、ひとつの業務を構築するにあたり、上記3つの処理を完全に独立させ、かつ各処理の関係としてデータ項目の関係を明確にした。すなわち、変更の少ないDBと、変更の多いUIをきり分け、UIの自由な設計と共有データのDB設計を独立させること、DBと画面間の関係を明確にし、業務本来のロジックを単純化させることを狙った。

さらに、多様な端末・WSで対応できるだけでなく、同じ人間でも多様な見方を実現するため、表示すべきデータと、表示方法を分離し、同一のデータに対して複数の表示方法をもてるようにした。

本システムモデルを図1に示す。

3 オブジェクト指向フォーム管理方式

UIとして、フォームメタファを採用した。フォームの利点は、事務処理業務における帳票・伝票と同じであり、

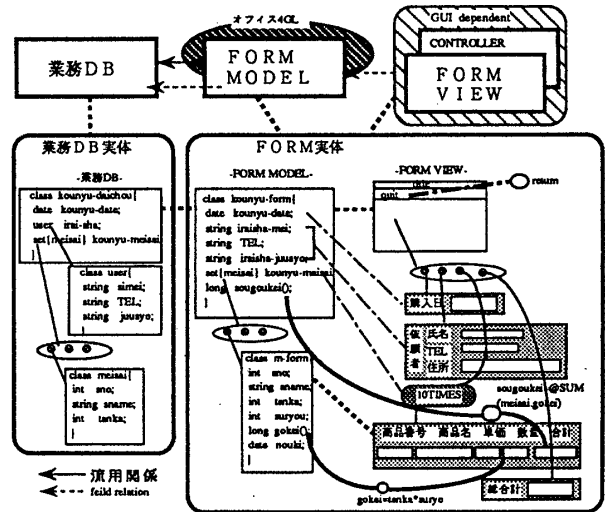


図1. システムモデル

利用者にとっては親しみやすく、オフィス業務を構築するにあたり、業務分析がやりやすい。

フォームは、表示すべきデータ構造と、その表示方法だけでなく、入力などのイベントに対応した手続き群を含む。フォームの中への手続きのカプセル化は、[1]でも行われている。ここでは、さらにフォームと業務DBの項目関係も含ませ、手続き中での転記を動的に生成することを可能にした。

データ構造はインスタンス変数、表示方法および、業務DBとの項目関係はフォームオブジェクトのクラス変数として定義し、手続きはフォームのインスタンスメソッドとして定義する。

フォームはオブジェクトであり、そのデータ構造と手続きについては、継承関係をもつことができる。フォームは業務DBの一種のビューであるとも考えられるので、フォームからの継承だけでなく、業務DBのデータ構造より継承させることも可能とする。

表示方法自体は表示部品クラスのインスタンスとして定義する。表示部品クラスはデータ構造をどう表示するかの手続きをもつ。

フォームオブジェクトの構造を図2に示す。

フォームの表示方法は複数形態もつことができる。単純なものは、あるデータ項目をTEXT WIDGETに対応させて表示する。複雑なものでは、あるデータ項目に入力されるべきデータが限定される場合など、その限定されるデータをLIST WIDGETで表示し、カーソルの位置でどのデータが入力されているかを示すことを可能にする。(図3)

4 オフィス業務記述

オフィス業務を記述するための必要な手続きを以下の4つに分類した。

- フォーム制御: 画面遷移を実現

Office Application Development Environment
- Object Oriented Form Management -
Keiko YAMAZAKI*, Hiroshi WAKABAYASHI**
*C&C Systems Research Laboratories,
**C&C Systems Engineering Division,
NEC Corporation

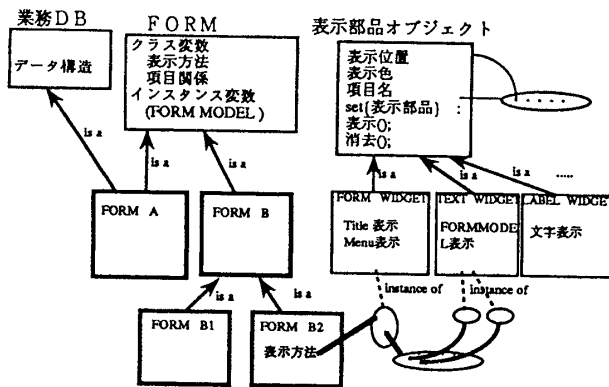


図2. フォームオブジェクトの構造

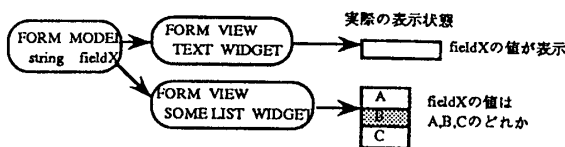


図3. 表示の一方法

- DB アクセス:DB の検索・更新・削除
- 転記:DB とフォーム間の項目転記
- 業務ロジック記述: 制御文、多種関数など

転記処理については、DB とフォームの項目関係が決定されていれば、自動的に実行できると考え、項目同士の関係情報をフォームのクラス変数としてもたせた。この項目関係情報は、継承関係のあるフォーム間で関係を継承できるようにした。(図4)

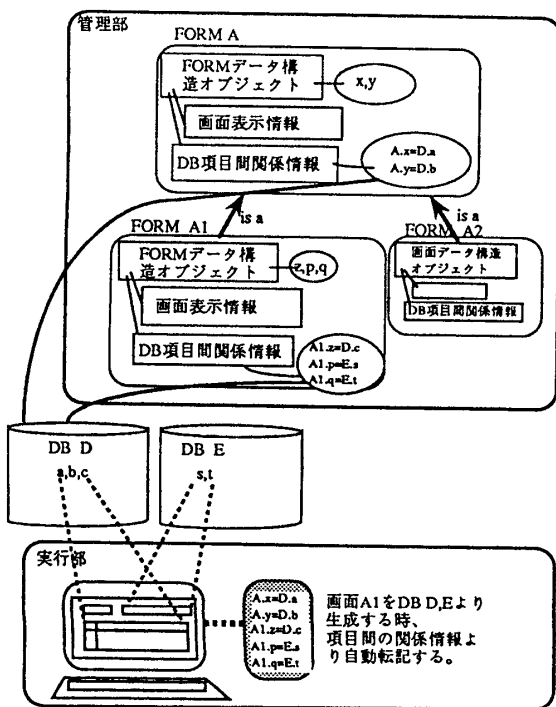


図4. 項目関係情報による自動転記

5 実現方法

フォームの管理に OODB(Odin[2]) を用いた。本システムは OODB/4GL としての位置づけにもなる。業務 DB もフォームも同じ OODB で管理されると、DB にフォームとして必要な情報を付加すればよく、システムがさらに単純なモデルで実現できる。

FORM VIEW としては UNIX の MOTIF 対応を実現したが、FORM VIEW の部分を変更すれば他の GUI にも対応できる。

本システムモデルを UI 表示側と DB サーバ側の 2 者間でクライアント・サーバ型に分離するにあたり、以下の 3 つの方法が考えられる。

- 業務 DB アクセスを分離:
UI+ 業務ロジック +DB アクセス —DB サーバ
- FORM MODEL を分離:
UI+ 業務ロジック —業務ロジック +DB アクセス
- FORM VIEW を分離:
UI サーバ—UI クライアント + 業務ロジック +DB アクセス

本システムの実現にあたっては、UNIX 上にて X ウィンドウを用い、FORM VIEW を分離する方式を採用した。今後は、高速なネットワークにつながれていない環境でも最適な分散を実現できる FORM MODEL を分離する方式を検討する予定である。

業務 DB としては OODB を利用しているため、イメージなどのマルチメディア情報を容易に扱うことが可能になっている。そのため、従来では難しかったマルチメディアを用いた業務構築をより簡単に構築することができる。

6 まとめ

現在開発したプロトタイプシステムは、開発環境の基盤となるフォーム管理部分のみであり、特に設計環境はまだない。従って、開発工数がどの程度削減できるかという観点で効果を測定することはできない。部品化の観点と、処理記述の独立性について述べる。

フォームをオブジェクト指向で実現することにより、異なるフォームの定義を既存のフォームの差分で記述することができる。この方法については、他のシステム [1] でも検証済みであり、効果が期待される。本方式では、フォームの表示部品をオブジェクトの単位で管理しておくことにより、フィールドの部品化を促進する。また、業務ロジックの大部分を占めている転記処理については、DB とフォーム間での項目関係情報を別情報として切り出すことにより、業務ロジック本体の修正部分を極小化することができる。

本システムではフォームを対話処理業務中の一時的なインスタンスとして生成しているが、フォームオブジェクトを OODB で管理しているため、永続的なオブジェクトとしてトランザクション間を流通するものとしても利用が可能である。これにより、業務の分散協調システムを可能にする。

今後は、設計環境を試作し、開発・保守工数の削減効果を測定するとともに、分散協調型のシステムへ拡張していきたいと考えている。

謝辞

当研究に関して、貴重な御助言をいただいた当社 C&C システム研究所 鶴岡邦敏課長に感謝いたします。

参考文献

[1] Hammainen, H., Eloranta, E., and Alasuvanto, J. *Distributed Form Management*, ACM Trans. Inf. Syst. Vol.8, No.1, pp.50-76, 1990.

[2] 木村, 鶴岡 「オブジェクト指向データベース操作言語 Odin/C++ について」, 電子情報通信学会データ工学研究会, DE90-26, pp.17-24, 1990.