

オブジェクト指向手法による金融機器用ソフトウェアの開発

4D-10

— ミドルソフトウェアの構成 —

中澤 修 末竹 義郎 首藤 文人 森 健 新田 哲二

沖電気工業株式会社

1 はじめに

オープン化時代に対応したATMを構築するにあたり、我々は金融機関毎に異なるATM上で稼働するアプリケーション（例えば、支払、入金、振込等）の生産性の向上、ATM機器の制御/情報の管理を行なうミドルソフト層の再利用性/カスタマイズ性の向上等を目標とし、オブジェクト指向手法による分析/設計/開発を実施することにした。しかし、システムすべてを完全にオブジェクト指向化するには技術的にリスクが大きいため[1]、機器制御/情報管理の仕様が明確化されているミドルソフト層を対象として、オブジェクト指向化を実施した。

なお、具体的な開発プロセスの推進にあたり、汎用OSやオブジェクト指向言語を用いた場合の性能問題の調査、ATM機器制御/情報のオブジェクト指向による表現方式の確立/プロトタイプ開発および評価等のフェジビリティスタディを行なうことで、問題点を洗い出し、実現可能性を検証している。

以下、本文では、AP開発の効率化手法、オブジェクト指向による分析/設計モデル、ミドルソフト層の構成を説明し、最後に現状および今後の予定等について述べる。

2 AP開発の効率化

従来のATMソフトウェアの開発は構造化手法を用いて行なわれている。つまりAP層で利用する機能をパッケージ機能モジュールとして構築し、そのモジュールを図1のようにAP層から利用するという、機能中心の考え方を採用している。

しかし、機能中心の方式では金融機関毎に異なる

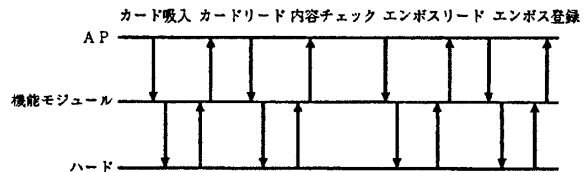


図1: 従来手法によるAP開発イメージ

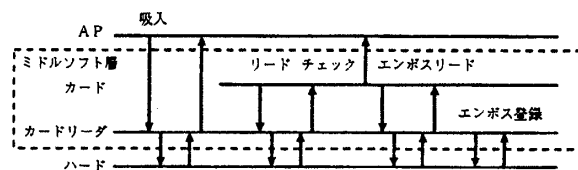


図2: オブジェクト指向手法によるAP開発イメージ

仕様の吸収の方法、AP開発の生産性等に問題がある。そこで、機能中心からデータ中心の考え方への転換を図り、ATMの媒体（カード、通帳等）を表すオブジェクトへの処理依頼によりAPが開発できる図2のような方式を採用することにした。なお、オブジェクト化に際しては以下のような方針を用いた。

- カプセル化による取引AP開発の効率化  
情報管理/機器管理等の細かな処理はオブジェクト内部にカプセル化する。
- 部品化/再利用、継承（カスタマイズ）の推進  
媒体系、機器系、通信系、画面系、顧客操作系の扱う情報の観点からオブジェクト化し、金融機関毎のカスタマイズが必要な場合は継承機能や動的リンクによる機能の置換機構等を利用する（APからは違いを意識する必要がない）。

これによりAPから発行すべき処理依頼の数を減少させると共に、APの基本構造は次の方式により開発可能となる。

- ミドルソフトオブジェクトをいつどのような順序で利用するかを制御を行なう。
- ミドルソフト層からのイベントにより実行する動作を決定する。

Object Oriented Software Development for Financial Application - Configuration of Middleware -  
Osamu Nakazawa, Yoshiro Suetake, Fumito Shuto,  
Takeshi Mori, Tetsuji Nitta

Oki Electric Industry Co., Ltd.

18-8, Chuou-1, Warabi-shi, Saitama 335, Japan

### 3 オブジェクト指向分析/設計モデル

具体的な分析/設計プロセスは

- ATMソフトウェア仕様の分析/設計  
OMT手法[2]をベースに機能仕様書からオブジェクト設計を実施する。
- 現行ATM仕様の再整理  
ソースプログラム等から機能(関数)を抽出する。

というトップダウン的な作業とボトムアップ的な作業の協調作業により行なうものである。図3が開発プロセスのモデルである。

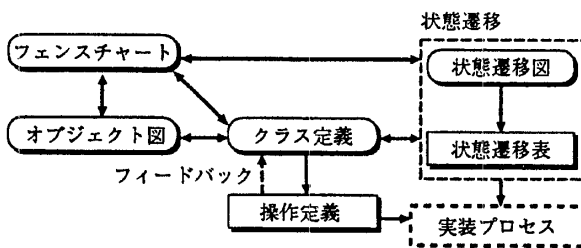


図3: オブジェクト指向分析/設計のモデル

- オブジェクト図: 各オブジェクト間の関係(操作依頼の相互関係/データ交換等)を整理する。
- 状態遷移図: オブジェクトが処理する主要なイベント(操作依頼等)を基に、オブジェクトの振舞いを整理する。
- 状態遷移表: 状態遷移図を表形式に整理する。ただし、例外処理等を含めすべてのイベントに対するオブジェクトの振舞いを整理する。
- フェンスチャート(タイミングチャート): 主要な処理フローにおける各オブジェクトの役割とオブジェクト間のインタフェースを整理する。
- クラス定義: クラスとして保持管理する属性、他オブジェクトとの関連、処理依頼として受け付ける操作(および内部処理操作)を定義する。
- 操作定義: 既存ATM仕様を基に処理内容と操作関数の入出力パラメータを定義する。

### 4 ミドルソフト層の構成

ミドルソフトは図4に示すようなフレームワークから構成される。API関数層はAP開発者が利用するいわゆるAPI関数群であり、ミドルソフト内

の各オブジェクトへの処理依頼から成る。情報オブジェクト層は

- ATMで扱う各種情報の保持/管理
- 情報に関する操作(チェック/編集/収集等)
- IOオブジェクトへの操作依頼

の各処理を行なうオブジェクト群から成り、更にIOオブジェクト層は

- 各IO機器(ドライバ)への動作依頼
- 機器からのアテンション/応答の受信

の各処理を行なうオブジェクト群から構成される。

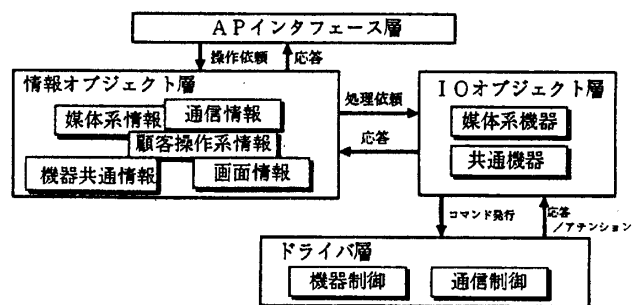


図4: ミドルソフト層の構成

### 5 おわりに

現在、ミドルソフト層としてのシステムテストが完了し、アプリケーション層を含む総合テストを実施中である。94年7月のフィービリティスタディ開始から約2年、C++言語による開発でクラス数33個、約200KLOCである。

オブジェクト指向化の実施による実質的な評価は今後の実際の運用/保守作業等を通して行なうことになるが、オブジェクトという単位で仕様が明確に整理できたこと、バグ発生件数を従来技法と比べ抑えることができたこと等から判断すると、着実に効果が現れるものと考えられる。

今後は、ミドルソフトの拡張性/ポータビリティ性の向上等を目指し、ブラッシュアップを進めると共に、オープンネットワーク化への対応等の検討を行なう予定である。

### 参考文献

- [1] 本位田 他: オブジェクト指向分析・設計, 共立出版, 1995
- [2] J.Rumbaugh: Object-Oriented Modeling and Design, Prentice Hall, 1992