

業務パッケージ開発におけるオブジェクト指向の適用<sup>†</sup>

4 C - 8

豊嶋 淳史 吉田 尚志 木下 誠一

NTTデータ通信(株)<sup>‡</sup>

1. はじめに

業務パッケージの適用は、システム開発の生産性および再利用性を向上する目的で行われる。多くのパッケージはシステム開発における変動要因を外部から操作する手段を提供することにより、パッケージ自身の変更を行わずに要件に対応できるような仕組みを提供している。しかしながら、ユーザ要件によってはパッケージ自体のカスタマイズが避けられない場合もあり、これにかかるコストは最悪の場合、パッケージを利用しないで開発する場合よりも大きなものになってしまう。

そこで、この問題を解決するためにオブジェクト指向技術を適用し、パッケージの構造を柔軟にすることで、変更が避けられない場合のコストを最小限にとどめられるようにしようと試みた。

本稿では、流通小売業をターゲットとするパッケージ開発にオブジェクト指向技術を適用した事例を元に、柔軟なパッケージの構造に関して考察する。

2. システム概要

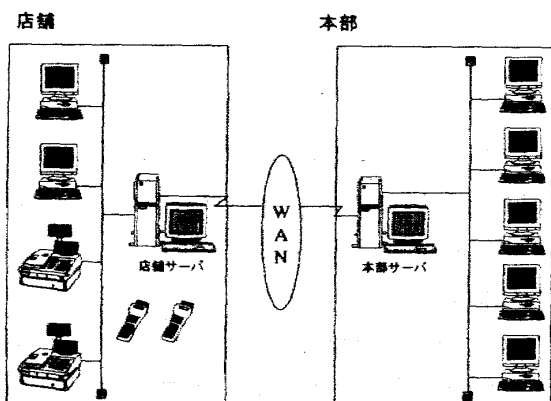


図 1 システム構成図

開発している流通小売業向けパッケージは、図 1 に示すような 1 本部-複数店舗構成の C/S 型システムを想定している。

本部では、バイヤーが各商品毎に販売計画を策定し、各店舗の特徴に見合った形で展開を指示する。各店舗は基本的に本部の指示に従い、商品の陳列・販売を行う。小売業の大きな流れとしては、発注-仕入-販売となるが、小売業であっても業種間の差異は大きく、また、同一業種内であっても会社毎の経営方針等による差異は大きく、これらの変動要因を予めすべて想定することは困難であるため、出来る限り構造を柔軟にすることが重要である。

3. 適用範囲

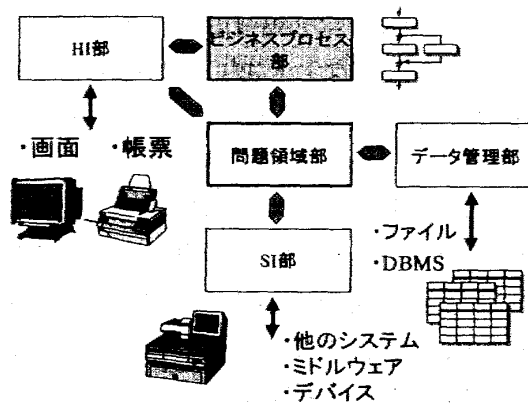


図 2 マルチコンポーネント概念図

オブジェクト指向技術を適用するに当たり、システム全体の変更に対する強度を上げるために、図 2 に示すような Coad の階層的なコンポーネントの概念[1]を拡張し適用した。一般的な階層コンポーネントに対して追加される“ビジネスプロセス部”は、通常はすべて問題領域部として考慮される部分を、業務の流れに代表される、個々のユーザによ

<sup>†</sup> Introduction of Object Oriented Technology in Business Package Software Development  
<sup>‡</sup> Atsufumi TESHIMA, Hisashi YOSHIDA, Seiichi KINOSHITA  
 NTT DATA Corporation, Software Engineering Center

て要件の異なることが予想される部分と、問題領域として扱うデータとして一般化され、ユーザによる変更があまりない部分とに切り離すことにより、コンポーネント間の影響を低減するためのものである。また、ビジネスプロセスはユーザにもわかりやすいため、この概念を切り出すことは非常に有効であると考えた。

一方、多様なユーザ要件を予めすべて把握することは不可能であり、想定されていない条件を満たせるようにするためには、問題領域部のクラス構成を柔軟に保つことは不可欠である。これを行うために、クラスの構造的および、業務知識から得られる柔軟性の両面に対して対策を打たなければならない。これに対してデザイン・パターン[2](以下、D.P.と略す)等を活用することにより、この柔軟性を上記両面に対して保証できるような方針を採った。

また、問題領域部として切り出されたクラス群の多くはシステム上の要件から、RDB上に格納しなければならない。クラスとテーブル間のマッピングに関する指針は様々な文献([3]他)として紹介されているが、決定的な手法はないのが現状であると考えられる。設計の方針としては、ユーザ要件としてユーザインタフェース部の応答時間をはじめとする処理速度の問題があり、クラス間の通信だけではすべての要求を満たせないことが判明した。この対策としてDBの機能を利用した一括処理方式を併用することにした。この場合、クラスからテーブルを設計したのでは一括処理要求を満たす場合の性能を保証できなくなるため、最終的には一括処理における性能を考慮し、DBの設計を問題領域部の設計と並行して行うこととした。問題領域部と物理的なDB構造との乖離はデータ管理部として、D.P.の適用等により柔軟性を保証するという方針を採った。

#### 4. 評価

##### 4.1 ビジネスプロセス部

パッケージ化対象業務の問題から、業務の流れをおさえなければならないような個所はほとんどなかったため、予め機構的に準備したビジネスプロセス部

自体を有効に活用するには至らなかった。

##### 4.2 問題領域部の構造

DBへのアクセスを高速化するために、インスタンスのブラウジング機構を実装した。これにより、インスタンス木からの検索効率を挙げる様にした。また、クラス構造の柔軟性は業務モデルに対するD.P.の適用で高くなったが、業務の面からの柔軟性は実適用前のため評価できなかった。

##### 4.3 データ管理部の構造

D.P.を適用することにより、問題領域部の階層構造をデータベース上ではフラットにしたことで、仕様変更された場合にテーブル追加が発生しても、既存の構造への影響がないようにした。

また、OIDの解釈をテーブルのキーとマッピングすることにより、既存のテーブルからクラスを設計した場合においてもテーブルのスキーマへの影響を押しえることができるようになった。

#### 5. 考察

本パッケージの開発を通じて、オブジェクト指向技術の適用によりパッケージの構造を柔軟にできることを確認した。しかしながら、開発後の実ユーザ適用はまだであり実証できたわけではない。今後は実適用した際に業務上の柔軟性をどのようにフィードバックし、本パッケージを進化させていくかが課題である。

#### 参考文献

- [1] Coad, P., North, D. and Mayfield, M.: Object Models - Strategies, Patterns, & Applications - 2<sup>nd</sup> Edition, Prentice-Hall (1997)
- [2] Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: Design Patterns: Elements of reusable Object-Oriented Software, Addison-Wesley (1995)(本位田真一, 本田和樹監訳: デザインパターン, ソフトバンク(1995))
- [3] Antenangeli, A.: Patterns for Mapping OO Applications to Relational Databases, C++ Report, September, 1997, pp.15-27