ビジネス系システム開発におけるオブジェクト指向開発技法（２）
４Ｕ－４　－３階層設計におけるＤＢ処理パターンの利用 －

萬木優子, 原田道明, 鈴木由美子, 北畠重信, 上原憲二
三菱電機（株）情報技術総合研究所

1. はじめに

ビジネス系システムにおけるソフトウェアの再利用促進の一環として，我々は環境から独立した「業務処理」の部品化とそれらの部品を利用した実装自動化を目指し，３階層アーキテクチャによるオブジェクト指向設計について検討を行なっている[1].

ビジネス系システムの大部分はDB操作を伴う処理で構成されるため，これらの処理から業務に関する処理を除いたDB処理本体は適用範囲が大きいと考え，今回２つの実システムを例題にDB処理に関する設計パターンの抽出を試みた.

2. ３階層アーキテクチャにおけるDB処理

オンラインDB処理中心のビジネス系システムでは，通常，処理の前後には情報はDB中に存在するため，基本となる処理単位は，「DB処理」と「業務処理」の組み合わせで表現される．実際，ある会計システムでは，オンライン処理画面の約８～９割が登録，更新，削除，表示，一覧検索等の基本的なDB処理に分類され，この基本処理に種々の業務ロジックがぶらさがる形になっている.

業務処理は業務毎に固有な処理であり，複数のエンティティオブジェクトが様々な形態の関係を持ちながら業務を実行するものである．しかし，サービスオブジェクトから見てエンティティオブジェクトのインタフェースが規約化されていれば，DB処理に関する３階層全体の構造は業務から独立して均一に設計することができるはずである.

例として，図１の３階層設計された読込処理を考える．この処理では，ユーザインタフェースからの指示をビジネスサービス経由でデータサービスに伝え，データサービスがDBからデータを取得してエンティティオブジェクトを作成する．そのエンティティオブジェクトはビジネスサービス層に返され，そこで業務ロジックによる事前処理を加えられたあとユーザインタフェース層に返されて画面に表示される.

この読込処理においてエンティティオブジェクトの持つ作成や事前処理等のインタフェースに規約があれば，全体の構造を変えることなくエンティティオブジェクトを同じインタフェースを持つ別のエンティティオブジェクトに入れ替えるだけで異なる業務の読込処理を実現できる.

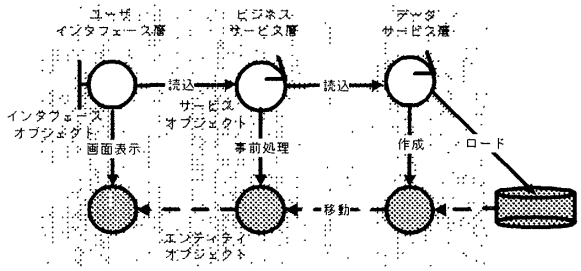

図1：読込処理

3. DB処理パターン

3.1パターン化の目的

DB処理自体は単純であり，どのようにも実装できるが，業務処理の組み込みを前提に各階層への具体的処理配分を明確にしたガイドラインを示すことは，設計作業の効率化，設計結果の均一化に役立つ.

3.2 DB処理の分類

DB処理のパターン化にあたり，DB処理を，次の観点で分類した.

- DB処理の種類
- 画面上での指定の有無

この分類に従ってパターン化を行なったところ全部で12の基本パターンが定義できた．１つのパターンは，オブジェクト間の関係やメソッドの引数にいくらかのバリエーションを持つ．またこれらの基本パターンの組み合わせたもので使用頻度の高いものを応用パターンとして定義した.

Object Oriented Software Development Method for Business Systems(2) － An Approach Using Database Access Patterns －
Yuko Yurugi, Michiaki Harada, Yumiko Suzuki, Shigenobu Kitabatake, Kenji Uehara
Mitsubishi Electoric Corpolation

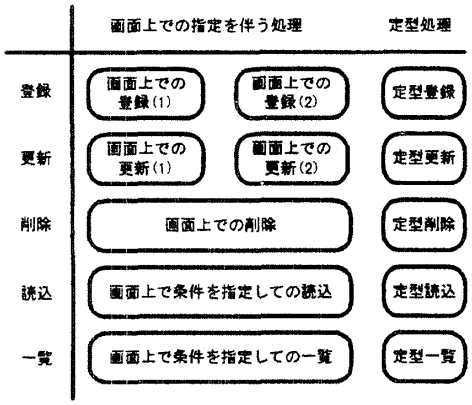

図2:12個の基本パターン

3.3 パターンの表現

DB処理パターンを表現する手段としては，いくつかの図法を試してみて，処理のシナリオ及びオブジェクト間の呼び出し関係を記述できるオブジェクト交信図を主な表現方法として選択した．図3は，ある登録パターンを示すオブジェクト交信図である．

図3にはエンティティオブジェクトが記述されているが，これはサービスオブジェクトからエンティティオブジェクトへのアクセスの仕方を示すためのものであって，エンティティオブジェクト同士の関係についてはごく単純な関係を想定している．DB処理パターンは，オブジェクト全体の構造，CRUDに関する依存関係等がわかっていることを前提にしているので，それらを考慮してパターンを選択し，エンティティオブジェクトのメソッド定義やデータサービス同士の呼び出し関係を定義する．

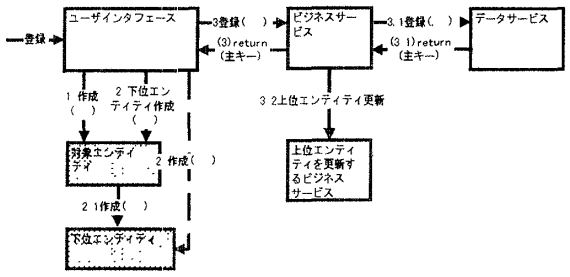

図3: 登録パターンのオブジェクト交信図

3.4 DB処理パターンを利用した設計

DB処理パターンは，設計の作業において３階層に分散したオブジェクトへの具体的な処理配分を規約化するものである．DB処理パターンを利用する場合，あらかじめデータ構造や業務ロジック等からオブジェクトを大雑把に定義しておいて，そこにパターンを適用する．

図4は実際にDB処理パターンを利用して設計を行なった例である．この例では２つのパターンを適用している．

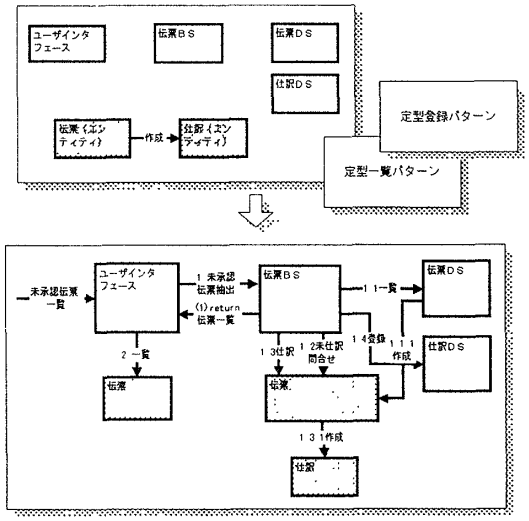

図4:DB処理パターンを利用した設計の例

3.5 DB処理パターンの利用による効果

DB処理パターンは，デザインパターン[2] のように汎用的に利用するものではなく，３階層という一つの環境を前提にDB処理に関する部分を機械的に設計することを目標としている．DB処理パターンを利用することにより期待できる効果として次のようなものが考えられる．

- 設計作業の効率化
- 過去の成功した設計を再利用することによる設計ミスの防止
- 標準パターンの利用による仕様書類の簡略化

4. おわりに

ビジネス系システムにおけるDB処理について３階層設計のためのパターン化を試みた．DB処理パターンを利用することにより，設計作業の効率化，設計ミスの防止等の効果を期待している．

今後の課題として，DB処理パターンの実適用評価，実装自動化への検討を行っていく予定である．

参考文献

[1] 鈴木由美子,萬木優子,原田道明,徳本修一他，ビジネス系システム開発におけるオブジェクト指向開発技法 (1)～(4)，本大会予稿集，1997

[2] E.Gamma，オブジェクト指向における再利用のためのデザインパターン，ソフトバンク(株), 1995