6C－5　オブジェクト分類による オブジェクト指向フレームワーク利用支援手法の提案

高木 浩則　瀬野尾 健
NTT ソフトウェア研究所

1．はじめに

オブジェクト指向フレームワーク（以降，フレームワーク）を基にしてAP（アプリケーションシステム）を開発することにより，これまでのクラスライブラリと比較して，より高いレベルでコードの再利用や設計の再利用が可能になる．このような利点がある反面，フレームワークの学習が難しく，フレームワークを十分に活用してAPを開発するためには非常に労力を要するという問題がある．

本稿では，フレームワークを利用したAP開発を円滑に進めることを目的に，（1）フレームワーク構築過程における分析・設計の視点，並びに（2）その構築過程におけるオブジェクトの出現，メソッドの変更などのオブジェクトの変化に着目したフレームワーク内部のオブジェクトの分類階層化/視点を提案する．これによって，開発者がフレームワークを基にしてAPを開発する際に，フレームワーク理解の容易化，変更箇所特定の容易化が可能になる．

2．フレームワークに基づくAP開発の問題点

フレームワークを基にAPを開発するためには，開発者はそのフレームワークの設計上の決断を理解し，どのように実現されているか把握した上でAPの機能要件に合致するようフレームワーク中のオブジェクトやメソッドを変更する必要がある．しかしフレームワークはクラス構造が複雑であり学習するのが難しく時間もかかるため[1]容易ではない．この点がフレームワークの利点が十分に活かせない，あるいは導入が進まない要因の一つになっている．

3．提案する手法

本手法では上記問題点を解決するために，フレームワークを設計する際の設計の視点，設計詳細化の過程の情報を利用し，以下の2点を提案する．

（1）フレームワークに含まれるオブジェクトの分類・整理，並びに設計上の判断との対応付け

（2）フレームワークが想定するAPの利用者や関連するシステムといった周りを取り巻く環境面とオブジェクトとの対応付け

これにより以下の2つの利点が生じる．

（1）フレームワークの理解が容易になる

フレームワークはその構築過程で順次詳細化されていくに従って新たなクラス（オブジェクト）が出現したり，メソッドが影響を受ける[2]．我々のフレームワーク構築の経験からも，最終的なフレームワークには業務を実現するために必要なオブジェクト（ビジネスレベルのオブジェクト），ネットワーク上でのデータの分散やサーバ間の通信などの物理的な制約やシステム的な要素を考慮したオブジェクト（アプリケーションレベルのオブジェクト），並びに使用するミドルウェアなど実装環境を考慮したオブジェクト（システムレベルのオブジェクト）など多様なオブジェクトが含まれる．そのためオブジェクトを分類階層化し，設計上の判断と対応付けすることによってフレームワーク自体が理解しやすくなる．またフレームワークを利用する際には，業務の理解，業務を実現するビジネスレベルオブジェクトの構造・関係の理解，並びに設計上の観点・判断の理解が必要であるが，フレームワークが想定するAPの利用者や関連するシステムといった周りの環境面から捉えた上でビジネスレベルのオブジェクトを参照した方がより一層理解しやすくなる．

（2）変更箇所の特定が容易になる

オブジェクトの分類階層化，設計上の判断との対応付け，並びに周りを取り巻く環境面との対応付けによって，フレームワークを基にしてAPを構築する際に，要件から変更箇所の特定が容易になる．

3．1　オブジェクト分類

フレームワークに含まれるオブジェクトを大きく，（a）ビジネスレベルオブジェクト，（b）アプリケーションレベルオブジェクト，（c）システムレベルオブジェクトに分類し対応付ける．この分類自体も設計の視点・判断そのものとなる（図1）．

またこのビジネスレベルオブジェクト並びにそれに含まれるメソッドを，その役割に応じて更に詳細に以下の8種類に分類し対応付ける．（a）から（e）は主に機能的な意味を持つ階層であり，（f）から（h）は主にデータ的な意味を持つ階層である．

A Method for supporting Framework-based Application Development Using Object Classification
Hironori TAKAKI, Ken SENOO
NTT Software Laboratories

（a）ビジネスプロセスに着目し，複数の人が行う処理をひとまとめにしたワークフローを意味する階層
（b）システムを利用する人の役割に着目し，その人が行う処理をまとめたクラスやメソッドを意味する階層
（c）一人の人が行う一つの処理に着目し，その処理の中で行う一連の作業の流れをまとめたクラスやメソッドを意味する階層
（d）処理の中に含まれるアトミックなトランザクションに着目し，そのトランザクションの中のステップをまとめたクラスやメソッドを意味する階層
（e）データに対する演算である計算ロジックを意味する階層
（f）基本的なデータ並びにデータにアクセスするためのアクセスメソッドをまとめた基本クラスを意味する階層
（g）基本クラス間の関係として意味を持つ関係クラスを意味する階層
（h）基本クラスや関係クラス並びに自身のクラスからも導出される派生クラスを意味する階層

またこれらの分類に加えて，フレームワークの構築過程おける設計上の判断と，それによって出現したオブジェクトや変更を受けたメソッドなとを関係付けておく（図1）．設計上の判断には，フレームワークに柔軟性･拡張性を持たせるためにとった判断なども含む．

3．2　オブジェクトと環境面との対応付け

上記のオブジェクトの分類とは別に，フレームワークが想定するAPの利用者や関連するシステムといった周りを取り巻く環境面からフレームワークを捉えるために，以下の3つの視点と対応付ける（図1）．

（a）Organization：企業間の関係，企業の組織階層や人の役割などの機能組織構造
（b）Process：ビジネス・プロセスやタスクを考慮した垂直方向の分類

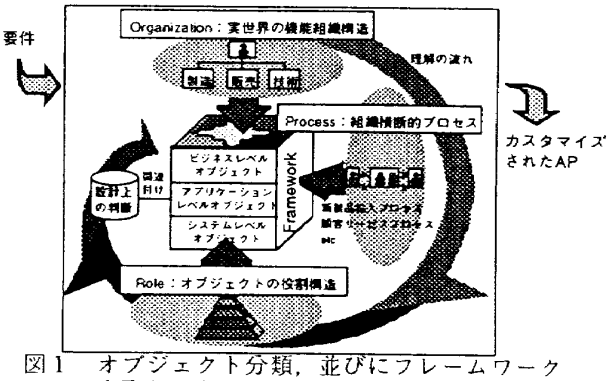

図1　オブジェクト分類，並びにフレームワークを取り囲む3つの視点

（c）Role:オブジェクトの役割を考慮した水平方向の分類

視点（a）はシステムを利用する人，所属する組織間の関係や企業間の関係などの実世界の役割構造を意味する．視点（a）によってフレームワークと実世界の役割構造とのマッピングを取ったモデルを考慮する．視点（b）はユーザの側面から見た，システムの外面を表現しているのに対して，視点（c）は開発者の側面から見た，システムの内側を表現している．ここで，視点（c）は上記3．1のオブジェクトの分類を意味する．

これら3つの視点とフレームワークとの対応をとる理由を以下に示す．

本来，開発されるシステムは企業の組織階層や実世界の人の役割関係と無縁ではなく，視点（a）によって実世界の機能組織構造とビジネスルールやデータの観点とを関連付けることによって，フレームワークが理解しやすくなる．ユーザの観点からフレームワークを捉えた視点（b）は，視点（a）同様，ユーザも理解可能であり，マシンやネットワーク，プログラミング言語などの観点からではなく，ユーザと開発者が同じ観点からフレームワークを捉えることができる．そのため両者の間のコミュニケーションギャップを埋めるという効果が期待できる．視点（c）でオブジェクトの関係を階層的に記述することによって，開発者がフレームワークの内部を，役割，イベント，ビジネスルール，サブジェクトなどの概念で捉えることができ，理解しやすくなると同時に変更箇所の特定が容易になる．

4．おわりに

本稿では，フレームワークを利用したAP開発を円滑に進めることを目的に，（1）フレームワーク構築過程における分析･設計の視点，並びに（2）その構築過程におけるオブジェクトの出現，メソッドの変更などのオブジェクトの変化に着目したフレームワーク内部のオブジェクトの分類階層化/視点を提案した．これにより，フレームワークが想定するAPの利用者や関連するシステムといった，ビジネスレベルのオブジェクトより更に上位の概念から理解し，設計・実装へとつなげていくことができる．実際の詳細な方法論等は今後明確にしていく必要がある．

参考文献

[1] Fayad, M. E. and Schmidt, D. C., "Object-Oriented Application Frameworks", Communication of the ACM, pp.32 - 38, Vol.40, No.10, Oct., 1997.
[2] 北山，"ビジネスオブジェクト開発における設計と実装"，情報処理学会OO'97シンポジウム，pp.108 - 111. 朝倉書房．