



コラム:

企業におけるパターン指向ソフトウェア開発の実践

[小特集] ソフトウェアパターン

—時を超えるソフトウェアの道—

位野木万里 東芝ソリューション(株)

【ソフトウェア資産としてのパターン】

ソフトウェア開発において、品質の保証された既存の成果物を再利用し製品開発を進めることは、生産性の向上、品質の安定、リードタイムの短縮に有効である¹⁾。再利用の対象となる成果物としては、設計ドキュメントやコンポーネントが一般的であるが、筆者らはパターンを再利用の対象に含めている。

【誰がパターンを利用するのか?】

製品ソフトウェア開発プロジェクトには、管理面を担当するプロジェクトマネージャ、製品の設計方針の定義を担当するアーキテクト、分析/設計/開発/テストを担当するエンジニアなどの技術者がかわる。技術者にはベテランと若手が混在し、大規模でグローバルな開発では、さまざまなカルチャーをバックグラウンドに持った技術者がメンバにアサインされることになる。

ベテラン技術者がソフトウェア開発プロジェクトで手本を示し、若手やオフショア開発メンバに開発の仕方を伝達し、ソフトウェア開発を遂行することが暗黙的に行われてきた。ベテラン技術者の持つ開発のノウハウをパターンとして形式知化すると、これまで暗黙的に行われてきたノウハウの伝達行為を「パターンを利用する」という明確な作業に置き換えることができる。このように、開発プロジェクトの若手やオフショアメンバがパターンの利用者となる。

【誰がパターンを定義するのか?】

開発プロジェクトにおいて得られたノウハウをパターンとして蓄積するには、ノウハウをパターンの形式で記述し、パターンの有効性を確認し、使い方や適用範囲の調整を行うことなどが必要になる。特定の開発プロジェクトに従事する技術者は、プロジェクトの遂行を優先するため、ノウハウのパターン

化という作業にリソースを費やすことは困難である。ノウハウのパターン化は、特定の製品開発プロジェクトに依存しない、横断的な役割の組織が担当することが現実的である。

筆者はプロジェクト横断の組織に所属しており、パターンを含む再利用のための各種成果物の開発と普及展開に携わっている²⁾。横断的組織がパターン化を担うことで、開発者にしか分からない暗黙知をどこまで形式知化できるかという課題はあるものの、成果物の重複開発防止や均質化が期待できる。

【パターンはどのように定義するのか?】

筆者らはノウハウを可視化する方法として2つの方式をとっている。1つめは、複数の成功プロジェクトの成果物を分析し、良い設計アーキテクチャや設計方法についての特性を洗い出し、それらを一般化してパターンとして記述する、ボトムアップ方式による方法である。この方式では、実際の成功事例からノウハウを抽出するので、実践的な内容にできる利点がある。しかし、既存の成果物の分析が必要のため、ノウハウの抽出や形式知化に時間とコストがかかる。

もう1つの方法は、既存のパターンや開発に関する最新の技術情報から、開発のあるべき姿を想定しパターンとして可視化する、トップダウン方式による方法である。この方式は「ノウハウの抽出」というよりは「ノウハウの創造」に近い。本方式では既存のやり方に限定されず、新しい開発方法や考え方の発見につながる利点がある。しかし、できあがったパターンの有効性を示すため、実プロジェクトで適用検証が必要になる。

【どのようなパターンを定義したのか?】

現在、筆者らが可視化しているパターンは以下の2種類である。

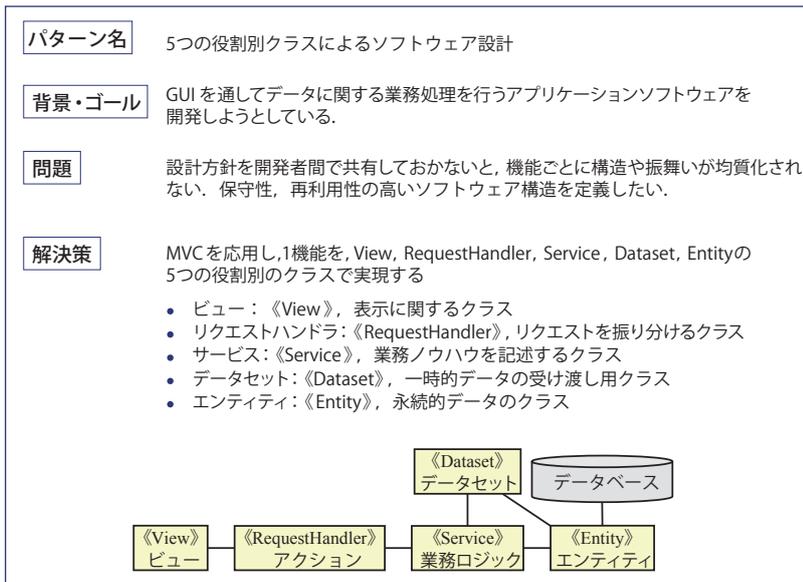


図-1 パターンの例

• 製品のアーキテクチャに関するパターン

これは、製品ソフトウェア・システムのサブシステム構成や、利用するオペレーティングシステムやミドルウェアのレイヤ構造などの基本的な設計方針に関するパターンである。

• 製品化までの分析や設計方法に関するパターン

これは、業種・業務分野の特性を抽出するための分析方法、要求変更に対応するための仕組みを実現するための設計方法など、分析や設計のノウハウに関するパターンである。

紙面の都合もあるため、本稿では、後者のパターンを1つ紹介する。図-1に、「5つの役割別クラスによるソフトウェア設計」パターンを示す。これは、ソフトウェアを、Model（データ）、View（画面や帳票）、Controller（制御）の3つの種類の要素で設計するというMVCパターンの応用形である。本パターンでは、ソフトウェアの構成要素をView, RequestHandler, Service, Dataset, Entityの5つの要素で定義するというノウハウを示している。本パターンはトップダウン方式により、横断的組織において考案し定義したものである。本パターンの目的は、画面を通じたデータの業務処理を行うための製品ソフトウェア全体を、統一化したソフトウェア構造とすることである。本パターンはさまざまな業種・業務分野に適用可能な汎用的な内容である。

本パターンを設計で適用することにより、ソフト

ウェアの構成要素が、MVCパターンの適用結果よりも細分化して定義されるので、個々の要素の役割が特定され、修正や変更の影響を限定しやすくなる。また本パターンを組織全体で適用すると、さまざまな業種・業務分野のソフトウェアが均質な構造で定義するため、開発・保守の方法が統一化でき、開発者のスキルに依存することなく、品質の安定化、保守性の向上に寄与する。

[パターンによる知識継承の有効性]

開発組織にとってソフトウェアの作成方法や考え方は、詳細に手順化し標準として共有することが重要である。しかし、組織の標準には明示的に含まれていない、モノ作りのコツや工夫は、暗黙知として技術者個人に埋もれがちである。プロジェクトに横断した組織がそのようなコツや工夫点をパターンとして形式知化することが、ベテラン技術者の持つ暗黙知の知識継承に有効であると考えられる。

ところで、パターンを利用した開発は技術者の思考停止を誘発するのでは？という疑問をお持ちの方もいらっしゃるであろう。開発対象のソフトウェアを分析し、適切なパターンを選択し、組み合わせ活用することは、高度な思考に集中することにつながる。価値あるノウハウがパターンとして蓄積できていれば、それらの再利用の体験は価値のある経験となり、技術者の思考力向上に有効である。

参考文献

- 1) 位野木万里：プロダクトライン開発への移行技術—既存シリーズ製品の再構築とコア資産管理, 情報処理, Vol.50, No.4, pp.280-288 (Apr. 2009).
- 2) 位野木万里, 深澤良彰：プロダクトラインの開発と進化：パターン指向コアアセット改善手法, 情報処理学会論文誌, Vol.48, No.8, pp.2508-2521 (Aug. 2007).

(2011年5月31日受付)

位野木万里 (正会員) inoki.mari@toshiba-sol.co.jp

1991年早稲田大学大学院修士課程修了。同年(株)東芝入社。2008年早稲田大学大学院博士課程修了。博士(工学)。専門はソフトウェア工学。現在、東芝ソリューション(株)IT技術研究所に所属。本会誌編集委員。