

システム再構築のカギとなるシステム基盤

野村総合研究所 河合 孝

既存システムの再構築を行う場合、すべての課題を解決することが難しく、どのような方針を策定すべきか悩むことが多い。システムは業務を適切に遂行するために存在するのであり、システム側の論理のみで構築するものではない。本稿では、このような悩みを解決するシステム基盤について、保険システムを題材に考察する。

再構築をためらっている間に…

再編の進む保険業界では、競争力強化のために、IT投資を継続しながら同時にITコスト削減を進めていくことが求められている。さらに既存のシステムも維持および改善していかなければならない。

その既存システムでは、汎用大型コンピュータ、クライアントサーバーシステム、Webシステムというように、複数の方式とプラットフォームが混在している。新しいシステム基盤上に再構築したいのだが、プログラムやデータの移行量を考慮するとなかなか踏み切れない。そのうちに、新しい技術を使わなければならない場合が出てきてしまう。新しい技術を使わなければならない機能が実現できない、あるいは新しい技術を使ったほうが低コストで再構築できる場合などである。

新しい技術を使うには新しいシステム基盤を構築しなければならないのだが、すでに開発期限も決まっているので局所的な対応でしのがざるを得ない。このようにしてさまざまなシステム基盤が乱立し、複雑さは一層増して、コスト削減もなかなか進まないということになりかねない。

システム基盤として考慮すべきこと

システム再構築を決断したら、システム基盤としてどのようなことを考慮すべきであろうか。

システム基盤構築には、ハードウェアと基本ソフトウェアの組み合わせの選定、アプリケーションを開発するためのミドルウェアの開発が最低限必要とされる。しかしそれだけではシステム開発はスムーズに進まず、メンテナンスフェーズで多くの問題を引き起こす。システム基盤は、実行・開発・運用すべてのフェーズをカバーしなければならない。レガシーと呼ばれる過去のシステム基盤では、実行環境のための仕組み以外に、開発環境および運用環境の仕組みが備わり、さらにそれらを円滑に運営するための標準化や運用における人間系のルールなどが存在しているはずである。それらは、当初はなかったにせよ、長年かけて蓄積されてきたものである。

新たなシステム基盤に習熟した人が少なく、人間系のルールも不十分な状況では、要員スキルを考慮した開発方式を確立することが重要である。最近の技術はオブジェクト指向を前提とするものが多いが、オブジェクト指

向スキルを全面的に採用するのは一考の余地がある。既存システムが残存している以上、構造化手法をなくすことは困難だからである。さまざまなスキルの要員がスキルセットによらず開発できるようにするには、オブジェクト指向スキルの必要な領域は基盤でラッピングして局所化するのもひとつの方法である。

また、メンテナンス開発を考慮した仕組み作りも重要であろう。たとえば保険システムにおいては、とくに損害保険のように、アプリケーションで扱う項目数が非常に多く、修正範囲の確認に多くの手間がかかる。短期間でシステム改訂を行うためにも、項目を一元管理し、修正範囲を的確に把握するとともに、修正に必要な各種情報を自動生成できる仕組みが必要ではないだろうか。

現在の技術で実現する

現在、実用化されている技術を組み合わせると、上で述べたことをどのように具体化できるだろうか。

まずオブジェクト指向スキルの必要性を局所化するためには、EJB（サーバー用のJava部品規格）やXML（拡張可能なマークアップ言語）を利用した組み立て指向型のミドルウェアを使うことが考えられる。オブジェクト（部品）を事前に用意し、その部品の実行順序を定義することでアプリケーションを構築するのである。組み立て指向型のアプリケーション開発であれば、構造化手法に慣れた

要員でも比較的取り組みやすい。またオブジェクト指向スキルは部品作りを担当する要員に局所化できる。

メンテナンス開発を容易にするためには、項目を中心に設計情報をリポジトリ（データ項目に関するデータを格納するデータベース）で一元管理できる仕組みを構築する必要がある。このリポジトリを分析する機能を追加することで、修正時の影響範囲が的確に把握できるようになる。修正のための情報の自動生成については注意が必要である。「ソースコード」ではなく、「定義情報」を自動生成するのであれば、生成したものを修正し、コンパイルする必要はない。

技術面を支える環境整備の重要性

技術的には、中核となるシステム基盤として開発フレームワーク製品（ミドルウェアや開発ツール）を利用することが効果的である。技術の進歩が速いので、自社ですべてを開発していると、出来上がった頃にはサポートされない技術になってしまうリスクがあるからである。しかしながら注意すべきは、こういった開発フレームワーク製品は、システム基盤で求められる要素をすべて満たしているわけではないことである。ただ採用・導入するだけでは、使いこなすことは難しい。

前述した実行・開発・運用環境をカバーする仕組み、標準化、運用における人間系のルールを整備することが不可欠である。 ■