SOAを適用したレガシーシステムの再構造化

硬直化した基幹業務システムの保守生産性の低下が大きな問題となっている企業は多い。その一因には、長年の機能改善の積み重ねによりシステム機能の構造が複雑化、密結合化していることがあげられる。本稿では、レガシー化した基幹業務システムの機能を再構造化するためにSOA（サービス指向アーキテクチャー）の設計原則を適用するアプローチを紹介する。

レガシーシステムの柔軟性向上への期待

近年の著しいビジネス環境の変化は、多くの企業に、変化へ迅速かつ柔軟に対応する能力の重要性を再認識させた。そして、環境の変化にただ追従するのではなく、自らを変革し、個別化・複雑化する顧客の要求に応え続けることが、企業にとって重要な差別化要因であることは論を待たない。このため、企業のITには、以前にも増してビジネス要件の変化に柔軟かつ迅速、低コストで対応可能な構造となっていることが重要となっている。

しかし、企業の基幹業務を支えるシステムは、長い時間を経て複雑化、ブラックボックス化し、それを放置してきた結果、機能仕様変更の影響範囲の見極めにすら多大な調査工数が必要となる例も少なくない。このような状況をシステムのライフサイクルの終わりととらえ、ただ再構築していたのでは、いつかまた同様の結果を招く。ITの柔軟性を継続的に確保するためには、あるべきシステム機能を構造化した上で、再構築やパッケージ適用などのソリューションを立案するべきである。

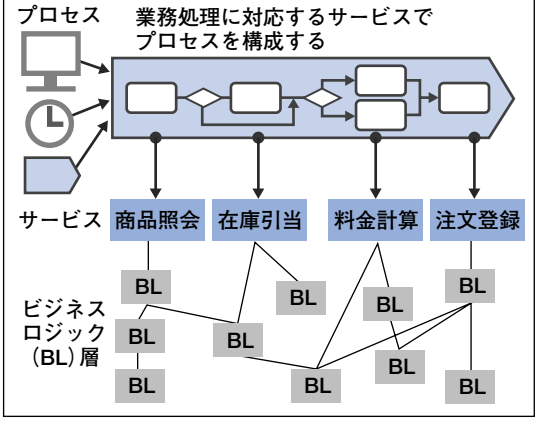
システム機能の構造化にはさまざまな手法がある。そのなかで、“システム機能のサービス化”をとり入れたSOAと呼ばれるアプローチを適用した事例が、筆者が参画した案件のなかにある。

SOAは、その出現の経緯からさまざまな説明、解釈がなされるが、本稿では、「システム機能を“サービス”としてソフトウェア部品化しておき、これを必要に応じて組み合わせ、再利用することで、必要なシステム機能を柔軟に構成するシステム設計の考え方」を指すこととする（図1参照）。

レガシーシステムの更改に特有の課題

システム機能の構造化はビジネスプロセスモデリングが出発点となる。ビジネスプロセスモデリングとは、あるべきビジネスプロセ

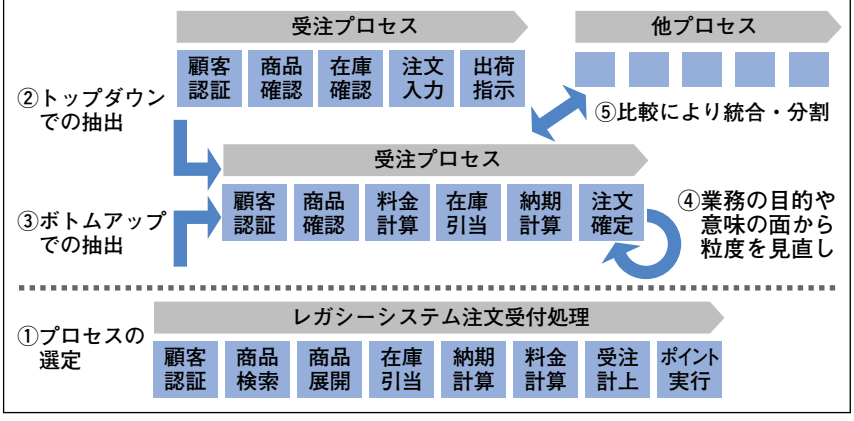
図1　サービスで構成されたビジネスロジック



野村総合研究所
システムコンサルティング事業本部
システムデザインコンサルティング部
上級システムコンサルタント
中山愛啓（なかやまよしひろ）
専門はシステム化構想・計画、要件定義

スに従ってトップダウンで業務イベントを整理し、業務要件や業務処理手順を明らかにしてゆく作業である。抽出した業務処理手順はサービスの候補となり、ビジネス視点での分割・統合が繰り返し検討され、サービスの粒度が決定される。

図2　トップダウンとボトムアップを組み合わせたアプローチによるサービス抽出



しかしながら、レガシーシステムの更改においては、システムによる自動化で埋没した既存の業務要件や業務処理手順を、どのようにして漏れなく抽出するかということがつねに問題となる。その原因は、長期間にわたって保守が繰り返されてきた結果、業務要件と機能を結びつけて理解できる人材が業務部門にもIT部門にもおらず、本来ならビジネス要件から業務要件、業務処理手順へとつながるはずの業務知識が失われてしまっていることにある。

したがって、システムから機能を抽出して、業務処理手順から業務要件へと遡及してサービス化するボトムアップのアプローチが必要となる。ただし、ボトムアップでの抽出にも注意すべき点がある。レガシーシステムの機能は、たとえ背景にあるビジネス要件の変化が小さい場合でも、適切な設計でない、場合によっては機能要件上の不具合を内包している可能性もあることである。

このように、レガシーシステムの機能の構造化においては、トップダウン、ボトムアップのいずれのアプローチにもそれぞれ限界がある。そのため両者を組み合わせたアプローチが必要なのである。

ビジネスプロセスモデリングの手法

ここでは、筆者が参画した案件に基づいて、以下の5つのステップをもつアプローチを紹介する（図2参照）。

①ビジネスプロセスの選定

まず、対象のビジネスプロセスを選定する。ビジネスプロセスの単位は、単一または複数の業務イベントにより開始され、連続的に実施される業務処理をグルーピングしたものである。この考え方でビジネスプロセスを洗い出し、そのなかで優先順位をつける。優先すべきプロセスは、たとえば注文から出荷指示に至るプロセスのように、顧客やサプライヤ

ーと直結するビジネスプロセスである。なぜなら、それは最も柔軟性が求められる部分だからである。

②プロセスの業務処理への分割

次に、選定したプロセスを業務処理に分割する。分割においては、業務処理の“対象”と“動作”（たとえば“在庫”と“引当”、“請求書”と“発行”）に着目する。留意すべきは、最初から細かい業務ケースをすべて明らかにすることにとらわれないようにすることである。

③ビジネスプロセスの抽象化・階層化

②のステップでは、先に述べたようにシステムに埋没した業務処理は粗いレベルでしか抽出されない可能性がある。そのため、レガシーシステムに埋没したビジネスプロセスの抽象化・階層化をボトムアップで行う。まず、設計ドキュメントなどから対象プロセスに該当する機能群を絞り込み、機能間の呼出関係を可視化し、その機能名称やコメントなどからロジックを抽象化する。この作業はシステムの規模によっては大きな労力を要する。実施にあたっては、リバースエンジニアリング（ソースコードから設計情報を解読する手法）ツールの活用などによる効率化策が不可欠である。

④業務処理の見直し・体系化

①〜③までの作業を終えると業務処理が多数抽出されるが、この段階では類似や重複があり、粒度の幅も大きい。そこでトップダウン、ボトムアップの両方のアプローチにより抽出された業務処理を見直し、体系化していく。具体的には、業務処理の“対象”を軸に、ビジネスプロセスで取り扱われる“アイテム”（注文、在庫など）やそこに登場する“ステークホルダー”（顧客、サプライヤーなど）に応じて分類する。分類別に、トップダウンで抽出された業務処理とボトムアップで抽出された業務処理を比較すると、粒度が異なるばかりでなく相互にかなりの不足が生じている場合がある。そのため業務とITのギャップが明らかになるのである。

このアプローチで重要なのは、いきなりギャップを埋めていくのではなく、ビジネス視点で、なぜこれらの業務処理が必要かという理由を出発点に業務処理を再構成する点である。これら一つひとつの業務処理がサービスの候補となる。ここまでは、SOAを適用するかどうかにかかわらず、一般的に行われるべきことである。ITの柔軟性を確保するためのサービス化の工夫は次の段階にある。

⑤サービスの統合・分割

第五のステップでは、サービスの候補を、共通部分と各プロセスに特有の部分に分け、共通部分を核としたサービスに個別部分を内包させるか切り出すかをきめ細かく整理していく。一般に、サービスの粒度は比較的大きいほうが望ましいと言われる。それは、サービスの利用者からみて、複雑な業務処理を組み合わせてビジネスプロセスを構成する負担

を軽減することで再利用性を高める効果があるからである。たとえば、新しいメディアを活用した販売プロセスの企画者が、顧客の与信状態や支払方法によって異なるチェック業務の中身を必ずしも知っている必要はないというわけである。

ただし、業務の目的や意味が同じでも、サービスの分割を検討すべき場合もある。たとえば、1つのサービスの要件決定を複数の業務部門が行う場合である。一度は共通化されたシステム機能も、異なる部門からの変更要求に対応するうちに、機能の複製により類似機能が増加し、保守性が低下することは多い。

機能の共通化による将来の保守コストの低減は不可欠であるが、柔軟性確保のために、過去の変更要求、サービス改修の意思決定者、サービスを利用する業務部門の違いを踏まえて、共通化されたサービスの分割の可能性を検討することは有効である。

このように、適用対象とするプロセスを拡大することで、業務視点でのサービスはより洗練されたものになる。

疎結合化と実装の複雑さはトレードオフ

実際にサービスを設計・実装するには、ソフトウェア設計の側面からの留意点がある。

ソフトウェア設計において機能の疎結合化を進めることは、柔軟性確保のために重要だと古くから言われてきた。しかし、サービスが疎結合であることを突き詰めると、個々のサービスが担う機能がシンプルになる一方で、業務処理よりも小さいサービスとなってしまう。それは、1つのビジネスプロセスを実現するのに組み合わせるべきサービスの増加をもたらし、サービスを連携させてトランザクション処理を実現する仕組みへの要求も高度になる。場合によっては、複雑なトランザクションの補償処理を作り込まなくてはならない。このような懸念は、ミドルウェアの成熟によって解決されることになろうが、現時点では配慮が必要である。

サービス単位のビジネスプロセス設計

サービスを疎結合として設計する目的は、再利用性・保守性を向上させるためであり、いたずらに疎結合であることを追求するのは本末転倒である。現実的には、サービスがサブシステムをまたがる場合には分割を原則とし、ソフトウェア機能面での疎結合化は、サービスの構成要素であるビジネスロジック層で確保することが望ましいと考える。

レガシーシステムの更改において、SOAの実装技術を適用するかどうかはともかく、業務視点のサービスという単位を組み合わせてビジネスプロセスを実現する設計手法は有効である。業務部門が、業務の背景や意味を熟知しているとは限らない。だからこそ、IT部門が対等に参画して、業務の意味からシステム機能を構成し、ビジネスプロセスを実現するという考え方が重要性を増している。 ■