

SOA—サービス指向アーキテクチャー

企業合併や事業部門の再編、統合が相次ぎ、ビジネスプロセスを迅速に変更、拡張したいというニーズが高まっている。こうしたビジネス環境の変化に対し、柔軟かつ迅速に対応可能なITシステムを実現できるものとしてSOA (Service Oriented Architecture) が注目を集めている。本稿では、SOAの概念と効果的導入のためのポイントについて考察する。

高まるシステムの柔軟性への要請

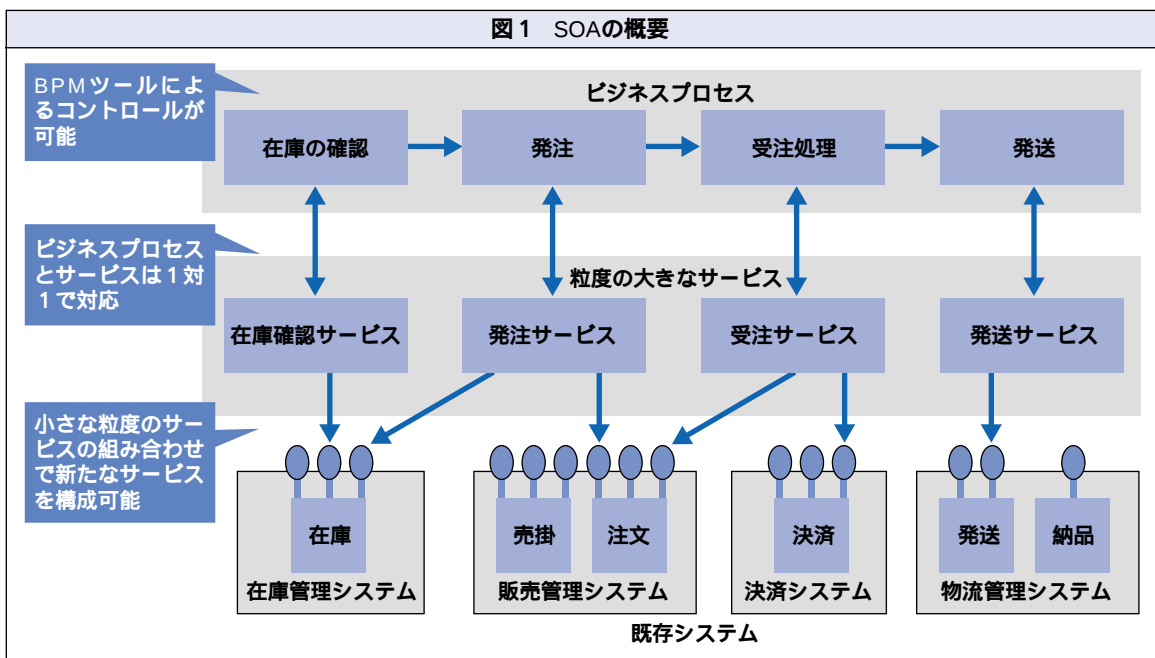
近年、ビジネス環境の変化が著しい。企業合併や事業部門の再編、統合などが相次ぎ、システムの適用範囲が拡大、縮小されると同時に、ビジネスプロセスも見直されるケースが多くなっている。また、こうしたドラスティックな変化を伴わないケースでも、事業の合理化、効率化の追求により、ビジネスプロセス自体が変化、拡張されるケースも少なくない。こうした時代にあって、システムが硬

直的なもので変化に対応できないものであれば、システムが足枷となって、事業そのものの競争力が失われることになってしまう。ひいては、たちまち企業そのものの存亡に関わることとなる。

このため、今日のITシステムは、「変化に柔軟かつ迅速に対応できる」ことが求められ、かつ、それにかかるコストも、それまで使ってきたアプリケーションの再利用などにより軽減されることが望ましい。

システムアーキテクチャーが、大型汎用コ

図1 SOAの概要





コンピュータに代表されるモノリシックな（巨大な一枚岩的な）構造から、クライアントサーバー型、さらには、Web 3 階層型のアーキテクチャーへと主流が移ってきていることなど、一連のIT革新

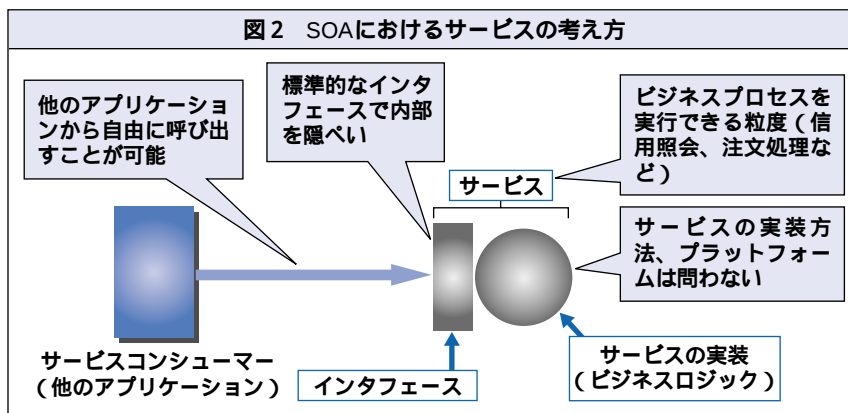
はこうした時代の要請に対するものととらえることができる。そんななか、現在高い注目を集めているのがSOAである。

粒度の粗いコンポーネント化という発想

SOAは、直訳すればサービス指向型アーキテクチャーだが、このままでは何のことかわからない。その具体的な内容は、アプリケーションの機能を「サービス」としてコンポーネント化（部品化）しておき、このサービスを必要に応じて組み合わせることによって、新たなアプリケーションを構成しようとするシステム構築の考え方を表している（図1参照）。

ここで言うサービスとは、標準的なインタフェースを備え、独立して稼動するソフトウェア部品で、外部から自由に呼び出すことができるものを指す。これらの条件を満たしていれば、サービスの実装方法やプラットフォームの種類は問わない（図2参照）。

実のところ、ソフトウェア部品をコンポーネント化して再利用するという考え方自体は



決して新しいものではない。むしろ、システム構築のコストカットという視点から、一般的に取り組まれてきたものである。

しかし、同じ「ソフトウェア部品のコンポーネント化」と言っても、SOAは、従来の考え方と決定的に異なるところがある。それは、再利用するソフトウェア部品の粒度の大きさにある。すなわち、SOAで言うところの「サービス」は、“在庫検索サービス”“発注サービス”“決済サービス”といった、現実のビジネスプロセスを構成できる業務処理をコンポーネント化の単位とするのが一般的であり、従来のコンポーネント（部品）の概念よりも大きい。

SOAの場合、コンポーネントをこうした比較的大きな単位でとらえるため、ビジネスの変化にともなってプロセスの変更や拡張を行う必要があるとき、それまで使っていたサービスを新しいサービスに置き換えるだけで済み、システム全体を一から作り直す必要がない。すなわち、変化に対して迅速かつ柔軟

に対応することが可能となるのである。

SOAの基盤となるESBとBPM

こうしたSOAの設計原則にのっとって考えれば、サービス（アプリケーション）の開発は、あくまでコンポーネントに過ぎないと言えることができる。コンポーネント化を進めることにより、アプリケーションの再利用性を高め、部分的にシステムを変更することができるというメリットは当然あるが、それだけでは、従来のコンポーネント化とさほど変わるものではない。

重要なのは、開発されたサービスを「組み合わせ」、新たな業務アプリケーションを構築することである。すなわち、各サービスを必要に応じて連携させ、ビジネスプロセスに従ってコントロールする仕組みこそがSOAの有用性を高める大きなポイントとなる。これを実現するミドルウェアとして昨今、注目を集めているのが、ESB（エンタープライズサービスバス）とBPM（ビジネスプロセス管理）である。

ESBは、SOAで設計されたサービスどうしの連携を仲介する伝送路のようなものであり、SOA環境を実現するための基盤となるものである。また、BPMは、連携させたサービスをあらかじめ定義した実際の業務プロセスに従って呼び出し、その実行を自動化するものである。具体的には、GUI（グラフィカルユーザーインタフェース）ツールを用い

て、複数のアプリケーションにまたがるプロセスをフローチャートを作成するようにグラフィカルに定義し、さらに定義したプロセスが適切に実行されているかの管理機能も備えているのが一般的である。

ちなみに、BPM自体は必ずしもSOAの環境においてのみ適用可能な技術ではない。これまで、アプリケーション統合のミドルウェアとして利用されてきたハブ&スポーク型のEAI（企業アプリケーション統合）技術と組み合わせることももちろん可能である。しかし、以下に述べる特徴ゆえ、SOAとの親和性が非常に高い。

その特徴とは、業務プロセスをルールという形で定義し、ルールエンジンで実行するといったように、アプリケーションの外側に置く点である。

これは従来、IT部門でしかコントロールできなかったビジネスルールを、ビジネスユーザー側でも変更できるようになったことを意味し、SOAの考え方にマッチする。先に述べたように、SOAはビジネスプロセス単位のサービスを実装するものであり、BPMを用いることでSOAを効果的かつ効率的に実現できるということになる。

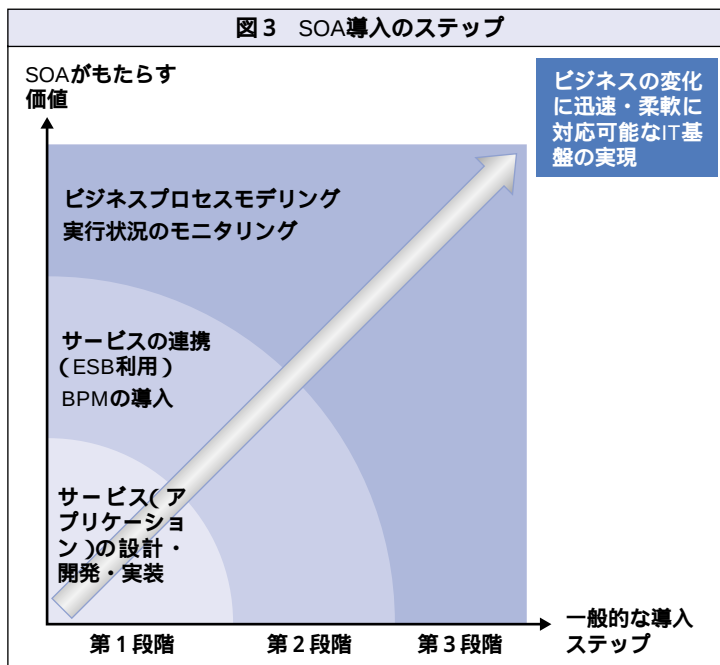
ビジネス視点から考えることの重要性

以上が、SOAの概略であるが、これを効果的に導入するためのポイントをあげておくことにする。

実際のところ、一般的にみられるSOAの導入プロセスには、いくつかの段階がある（図3参照）。サービス（アプリケーション）をコンポーネントとして設計、開発し、実装を行うことを第1段階とすると、ESB、BPMによって各サービスを連携させるのが第2段階である。現在、最も多くみられるのは、これら第1段階に加え、第2段階の導入を検討するというものである。

しかし、SOAがもたらす価値をさらに高めるには、さらに高次の第3段階が存在する。それは、SOAを単なるシステム上のツールとして導入するのではなく、ビジネスプロセスのモデリングを行い、システムの実行状況のモニタリングからPDCA（Plan—Do—Check—Action）サイクルを改善に役立てていくことである。

このモデリング作業では、ビジネスプロセスにおける業務イベントを抽出し、内容を明確にすると同時に、具体的な業務の処理手順や各プロセスの所要コスト、所要時間などについて、あるべき姿を描く。このため、モデリング作業はIT部門ではなく、実際の業務に精通しているビジネス部門が行うのが望ましい。モデリングを行った後に、前述したBPMツールなどを用いて実際の現場へ適用しBPMを実行していくことになるが、実行



状況を継続的にモニタリングし、そのパフォーマンスを分析していくことになる。これにより、ボトルネックを発見し、改善策を検討することができるというわけである。

当たり前のことであるが、ITシステムは事業を効率的に運営していくためのツールに過ぎない。したがって、最初にITシステムありきで考えるのではなく、ビジネス視点から考えることが重要である。システム構築の原点に立ち返り、上流から体系化していくことが必要である。こうしたことを考えれば、いきなりIT部門によるアプリケーション開発から始めるのではなく、「何のためのITか」をよく考え、「ビジネス視点発」を前提とし、ビジネスプロセスのモデリングからSOA導入を開始することが望ましいと言えよう。■