

# 保険業界における SOA (Service Oriented Architectures) の活用 ～introduction of SOA into the insurance industry～

IT研究会 第3グループ

## <担当委員>

西田 宏一 (日本興亜損害)

## <メンバー>

吉田 亮 (アメリカファミリー生命)

難波 主年 (三井生命)

山根 敬士 (ニッセイ情報)

山本 亙 (朝日生命)

阿部 愛博 (ニッセイ情報)

中村 圭介 (プルデンシャル生命)

旭 賞 (アリコジャパン)

関根 一樹 (太陽生命)

縦木 一登 (AIU保険)

## <目次>

はじめに

第I章. 保険業界を取り巻く環境とシステムが抱える問題

第II章. SOAとは

第III章. SOA導入手法

第IV章. 保険会社におけるSOA導入案

第V章. SOA導入後のシステムについての検証

第VI章. SOA導入への提言

おわりに

## はじめに

90年代後半から始まった規制緩和により、保険業界を取り巻く環境は大きく変化し、各保険会社は厳しい企業間競争にさらされている。この競争を勝ち抜くために各社は「顧客満足度の向上」、「スピード、利便性の向上」のための、「組織と業務の最適化」、「独自性を意識したビジネスモデルの構築」を意識した経営が求められている。一方で保険会社のシステムはメインフレームを中心としているケースが多く、追加要件を継ぎ足して構築してきた経緯もあって、近年ではシステムの複雑化が大きな問題として顕在化している。経営変革を実現する上で必要不可欠なはずであるITが、複雑化によってむしろ経営変革の妨げとなってしまっている例は少なくない。

この解決方法の1つとして、SOA (Service Oriented Architectures) が注目されている。SOAとはシステムをサービスの集まりとして構築する手法である。様々な導入事例が報告されているが、現状では、保険業界におけるSOAの導入事例は未だ殆ど無い。

本論文は保険業界には浸透していない「SOA」について研究を行い、如何にして導入していくべきかを纏めたものである。まず保険業界を取り巻く環境とIT部門との関係の把握を第I章で行い、第II章／第III章で耳慣れない「SOA」の概念や導入手法についての解説を行う。続いて、導入手順について、IT研究会第3グループで「一番現実的」と思われたアプローチ方法を用いた具体的解説と、効果の検証を第IV章／第V章で行い、最後の第VI章では一年間の研究の成果として、SOA導入に際しての提言を行う。

最終章までお付き合いいただければ、「SOA」とは何なのかを理解頂けるとともに、保険業界においても十二分に有用なアーキテクチャーであることを実感して頂けるはずである。

## 第 I 章 保険業界を取り巻く環境とシステムが抱える問題

近年、保険業界を取り巻く環境は大きく変化している。90年代後半から始まった規制緩和を筆頭に細かく挙げると枚挙にいとまが無い。代表的なものとしては以下の通りとなる。

- ・自由化／規制緩和によって、他業種からの参入による競争の激化
- ・出生率の低下、平均寿命の伸長による、少子高齢化社会の到来
- ・インターネットの普及による顧客販売チャネルの多様化
- ・コンプライアンスの重要性
- ・ライフスタイルの多様化

このような環境の変化によって、保険会社は厳しい企業間競争にさらされており、競争を勝ち抜くために変革を迫られている。顧客データの集約／分析や個人にカスタマイズされた商品の提供など顧客ニーズへの柔軟な対応、また新商品の市場投入までの期間短縮、及びチャネルの多様化によるスピード／利便性の向上を行うことによって、顧客満足度を向上しなければならない。一方で、企業が顧客からの強い信頼を獲得するために、コンプライアンス／内部統制／危機管理体制の整備といった統治体制の強化による「ブランドイメージの向上」に努めなければならない。こうした変革を達成するために保険会社は、「組織と業務の最適化」「独自性を意識したビジネスモデルの構築」という2つのテーマの解決を迫られている。

このような変革を迫られる中で保険会社のシステムは、柔軟性に富み、高品質なものを短期間で、且つ低コストで提供することが求められる。これらはどの保険会社においても、IT部門に対する共通のリクエストであると考えられる。この状況を鑑み、我々は保険会社40社のIT部門の方にご協力を頂き、IT部門を取り巻く状況の把握をしようと試みた。

以下の図I-1は「顕在化しているシステムの問題点」というテーマに対する回答結果である。この結果を見ると、「システムの複雑化」、「開発期間が短い」、「システムの老朽化」、「ユーザー要件の複雑化」といった共通の問題を抱えているのが一目で分る。

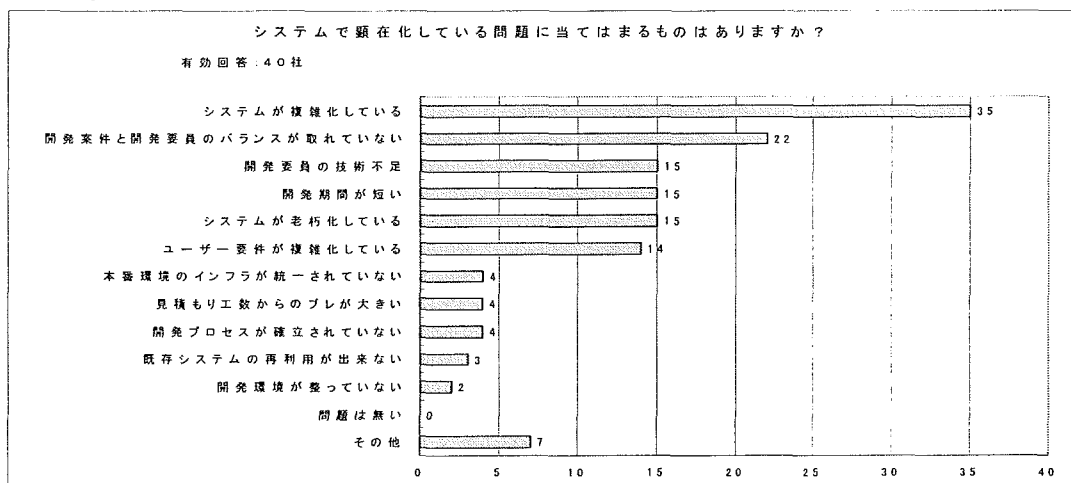


図 I-1

では、なぜシステムはこのような問題を抱えてしまったのか。その要因の一つは、「保険会社特有のシステムの変遷」が挙げられる。また、保険商品特有の特徴として、以下2点が挙げられる。

- ・保険商品は数十年にわたる長期契約が多く、契約の続く限り該当システムの保守／運用を続けなければいけない。
- ・保険商品の管理には、複雑な計算を必要とするため、計算能力が高くかつ安定した仕組みが必要とされている。

このような特性から、一度作ってしまったシステムをスクラップ&ビルドすることは難しく、既存システムに機能を追加し続けることで、ユーザー要件に対処してきた。

近年では、柔軟なインターフェースを実装するために、オープン系の技術を取り込み、メインフレームと両立させることも重要視されている。また昨今ではオープン系の技術が向上しているので、メインフレームに替わり業務を担うケースも増えてきている。現状のシステム構成と今後の方向性についてのアンケート結果が図I-2である。現状はメインフレームが主体になっているが、今後の方向性はメインフレームとオープン系の両立を想定している会社が8割を占めていることがわかった。

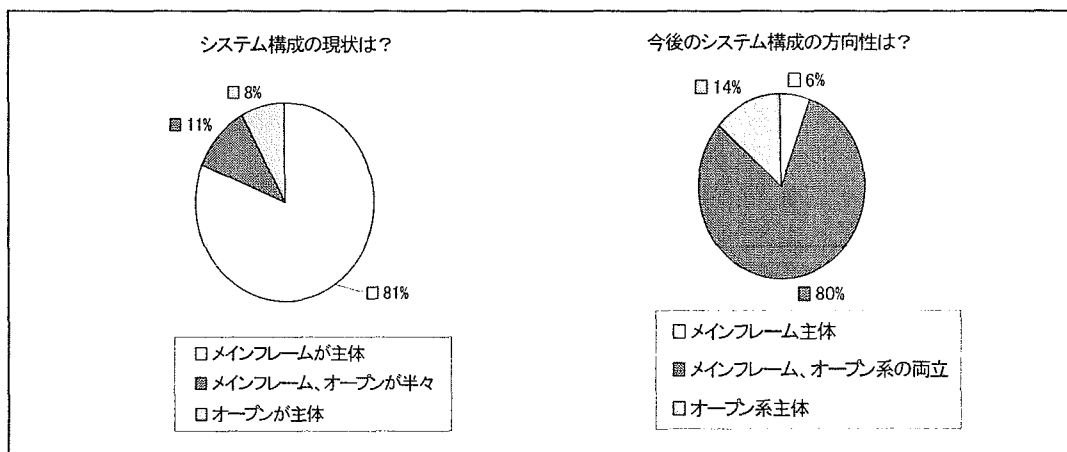


図 I-2

前述の保険会社のシステムが抱えるさまざまな問題に加え、GUIでメインフレームに勝るオープン系の技術革新や、インターネットの普及に伴う、webサービスの展開が「システムの複雑化や老朽化」に拍車をかける結果を招いている。システムは複数のインフラ基盤や、異なる言語のアプリケーションを抱えざるを得なくなり、これらを行き来する機能をユーザーに提供しなければならないからである。

当然ながら、こういったシステムの保守は莫大な費用と期間を要するようになり、ユーザー部門が求める「素早いシステム化」、「きめ細やかなサービスの実現」に至っていない。結果として保険会社のシステムは、「組織と業務の最適化」、「独自性を意識したビジネスモデルの構築」といった保険会社の重要テーマの妨げとなってしまっているのである。

このような難題を解決し、保険会社の重要テーマを実現するためのシステムを構築するのに理想的な考え方がSOAである。

現在IT業界ではSOAが非常に注目されているが、保険業界ではあまり浸透していない。SOAがどれほど浸透しているものなのか、保険会社にアンケートをとった結果が図I-3である。全体の60%以上が、「理解していない」と回答している。また、SOAの取り組み状況についてアンケートをとった結果が図I-4である。全社をあげて取り組んでいる企業はなく、大多数の会社は何もしていないという結果であった。

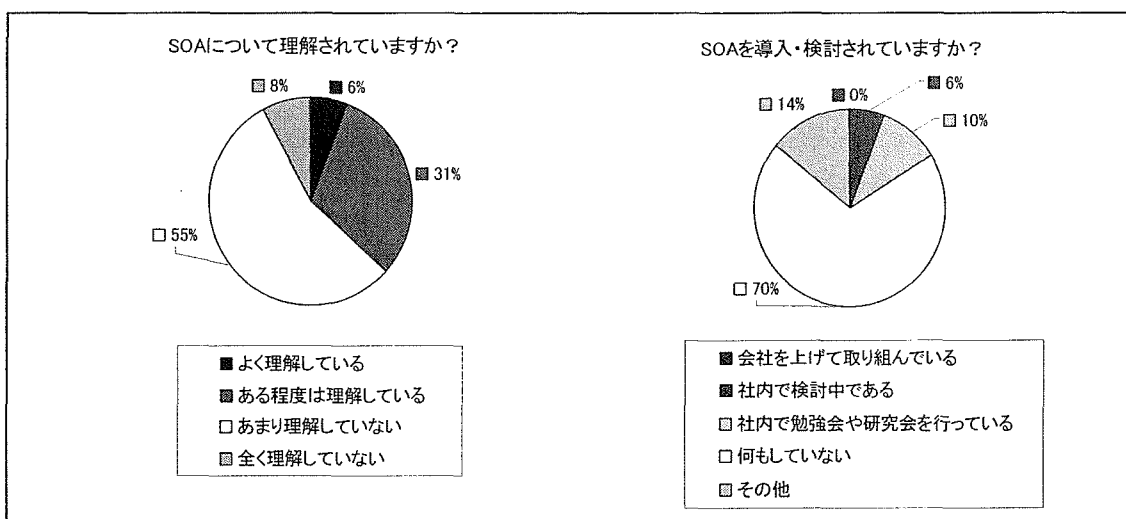


図 I - 3

図 I - 4

IT業界では注目されているが、保険業界ではそれほど浸透していない。このSOAとは、いったいどういったものなのか、どのように導入すればよいか。第II章以降では、それらのSOAについての疑問を解決し、保険業界にとって「有益なアーキテクチャー」であることを実証していく。

## 第Ⅱ章 SOA (Service Oriented Architectures)とは

前章で説明した通り、現在保険会社のシステムは複雑化や老朽化といったさまざまな問題を抱えている。このようなシステムの問題点を解決するための、理想のシステムを構築する考え方としてSOAが注目されている。第Ⅱ章ではこのSOAの定義、基本概念、メリットについて説明する。

### 1. SOAの定義

SOAを最初に定義したのは米国の調査会社ガートナーが1996年に作成したレポート「Service-Oriented Architectures, Part 1&2」であると言われている。以下はガートナー社によるSOAの定義である。

#### 【ガートナー社定義】

SOAとは、アプリケーションのビジネスロジックがモジュール（サービス）で構成されているアプリケーション・トポロジであり、完全な自立性、目的、プログラム型アクセス・インターフェースを持っている。サービスは「ブラック・ボックス」として動いており、サービスの内部設計はリクエストの特性や目的とは無関係である

このようにガートナー社の定義は抽象的であったがために「標準的な定義」とはならず、現在、各ベンダー会社はガートナー社の定義を元に、独自の解釈を行っている。以下はそれらの一部の例である。

#### 【A社定義】

コンピュータの独立した機能を「サービス」という単位で実装し、それらを組み合わせてシステムを作り上げる考え方。SOAとは考え方であり、テクノロジーや製品ではない。

#### 【B社定義】

効率的な再利用を目的に、自立した機能で構成される「サービス」と呼ばれるソフトウェアコンポーネントを構築し、「サービス」を組み合わせる活用する仕組みを用いて高い生産性を実現するシステムアーキテクチャーである。

#### 【C社定義】

疎結合システムであり、ビジネスとITのギャップを埋めるもの。つまりビジネス要件を「サービス」として表し、その「サービス」をコンポーネントやオブジェクトで構築していくもの。

以上のようにSOAの定義は各ベンダーによって異なる。A社のSOA定義は、概念的な説明である。一方B社の定義では、SOAの目的は再利用性にあり、その結果として高い生産性を得られると言及している。また、C社の定義では、サービ

スはビジネスとITの関係の緩衝材である、と述べておりITとビジネスの関係性について言及している。このように、各社ともガートナー社の定義をベースにそれぞれの解釈を交えて具体化する努力をしているが、SOAの最も重要な要素であるサービスの定義については考え方に違いがあり、標準的な定義は未だ存在していない状態である。

これがSOAの浸透を阻害する一因であると考え、IT研究会第3グループでは、ガートナー社と各社の定義をもとに独自の解釈を行い、SOAの「標準的な定義」を検討した。

#### 【IT研究会第3グループ定義】

ビジネスプロセスをある一定の粒度に分割／定義し、それらのサービスの組み合わせによって、業務の実現を図る考え方。

以上が第3グループのSOAの定義である。SOAの最も重要な要素であるサービスは「一連のビジネスプロセスの1ステップを再利用可能な単位で分解したもの」とする。例えば、商品受注業務を発注や在庫確認などの視点で再利用可能な単位に分解したものである。保険業界の新契約業務で考えると、告知査定、申込書不備チェック、加入限度額チェックなどがサービスに当たる。

業務視点で定義されたそれらのサービスの一つ一つは多様なプラットフォームに存在する複数のシステムコンポーネントやデータの組み合わせで実現される。つまり、業務とシステムの間にはサービスという層が入り、その繋がりを疎とすることができる。

このように、サービスを再利用可能な粒度に分割／定義しておくことにより、ビジネスプロセスの変更（組織変更や会社合併等）が行われた場合でも、個々のサービスの組み替えによって対応することができる柔軟性の高いシステムを構築することが可能となる。

## 2. SOAの基本概念

### (1) SOAの構成要素

SOAの構成要素は、以下の四つである。

#### a. サービス

第II章 2-(3)にて詳細を説明する。

#### b. ESB (Enterprise Service Bus)

サービス間の通信についてフォーマット変換、ルーティングなどを行う交通整理役である。このESBによりサービスの追加や変更にも柔軟に対応することが可能となる。システムで実装するには、サーバー製品の導入、各インフラにクライアント製品の導入等々の方法を取る。

#### c. インターフェース

サービスにアクセスするための必要な情報を定義してあり、概念的なサービスを形作る重要な要素である。具体的にはWSDL (Web Service Description Language) というインターフェース記述言語にて定義される。こ

これはサービスを形成するコンポーネントの内容とその呼び出し方、物理的な存在場所が記述されてあるサービスの定義情報である。

d. BPELエンジン

BPEL (Business Process Execution Language)とはサービス統合の記述言語である。BPELエンジンとはどのサービスをどのような順番で呼び出し、実行すればよいかを定義している。このBPELエンジンにより、複数の異なる環境が混在している複雑なビジネスプロセスを、柔軟な方法で統合できるようになる。システムで実装するには、サーバー製品の導入、各インフラにクライアント製品の導入等々の方法を取る。

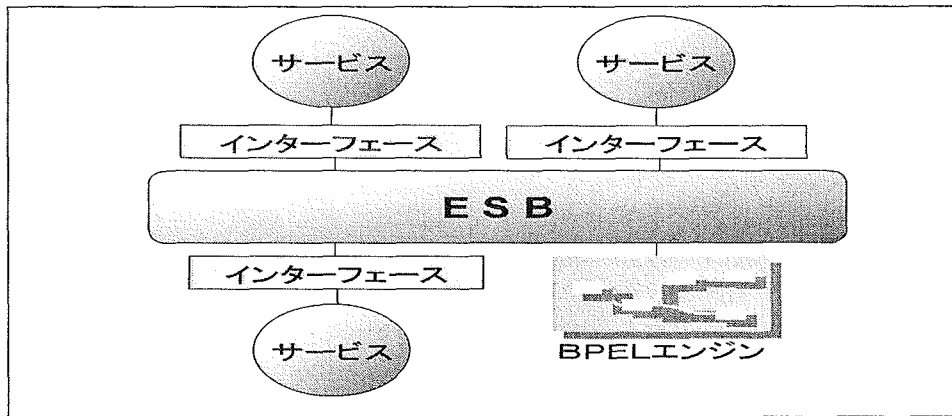


図 II-1 SOA構成要素概念図

(2) SOAのインフラ概念

SOAをインフラの視点から俯瞰すると下図のようになる。個々のサービスに関するアクセスはESBを介して行われる為、メインフレームとサーバーなど、インフラの違いによる通信方式の違いは全てESBで吸収され、様々な環境上に構築されたシステムの連携を可能にする。

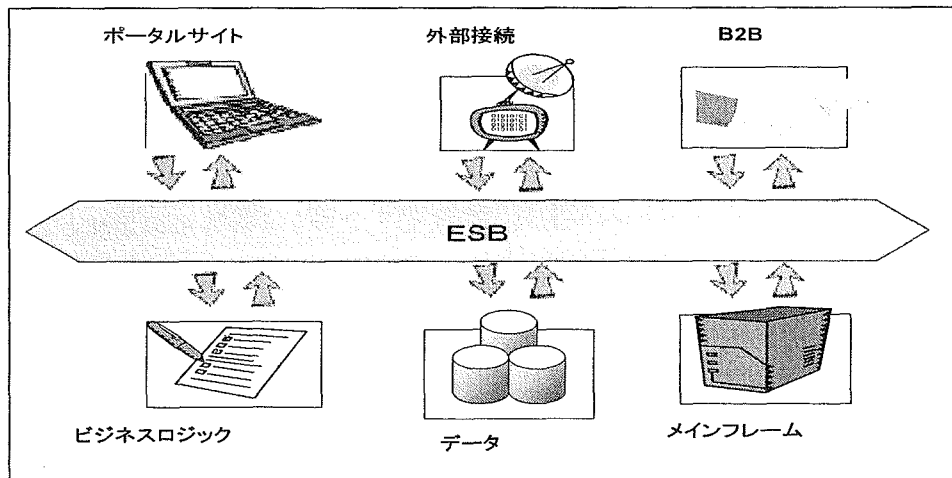


図 II-2 SOAインフラ概念図



### (3) SOAにおけるサービス

SOAにおけるサービスとは、ビジネスの単位やプロセスを見据えながら、ある程度意味のある大きさにシステムコンポーネントをまとめたものである。

サービスは単体で実際の業務の一部や特定のビジネスプロセスを実現することが想定される。WSDL (インターフェース記述言語) によって定義されたロジックによってサービスへ処理を依頼すると、サービスは内部で処理を完結させ、結果を返す。内部ではいくつかの処理がコンポーネントで分割されている。ここで、重要なことは、サービスは粒度の大きなコンポーネントではなく、ビジネス要件の一部がサービスであり、そのサービスをシステムコンポーネントで構築していくことである。つまり、SOAにおけるサービスとは以下の特徴を持つ。

- ・ビジネスプロセスの一ステップ
- ・開発言語、インフラ基盤に依存しない
- ・サービス単体で自立している
- ・外部から呼び出し可能なインターフェースを持つ

サービスの実装は java のクラスであっても COBOL であっても構わない。サービスの設計を進める上で重要なことは、自立性を確保し、かつ明確に定義された外部から呼び出し可能なインターフェースを持つことである。ここでいう自立性とは疎結合を維持するために必要な要素であり、外部サービスを介した個別コンポーネントの呼び出しや、外部サービスの管理するデータベースへの直接的なアクセスを排除することで創出することができる。

サービスは、物理的には複数のシステムコンポーネントやデータの組み合わせで実現される。以下はサービスとシステムのマッピングの例である。

サービス①：一つのシステム全てがサービスとなった例

サービス②：複数システムで構成されている一機能がサービスとなった例

サービス③：一システムの一部分がサービスとなった例

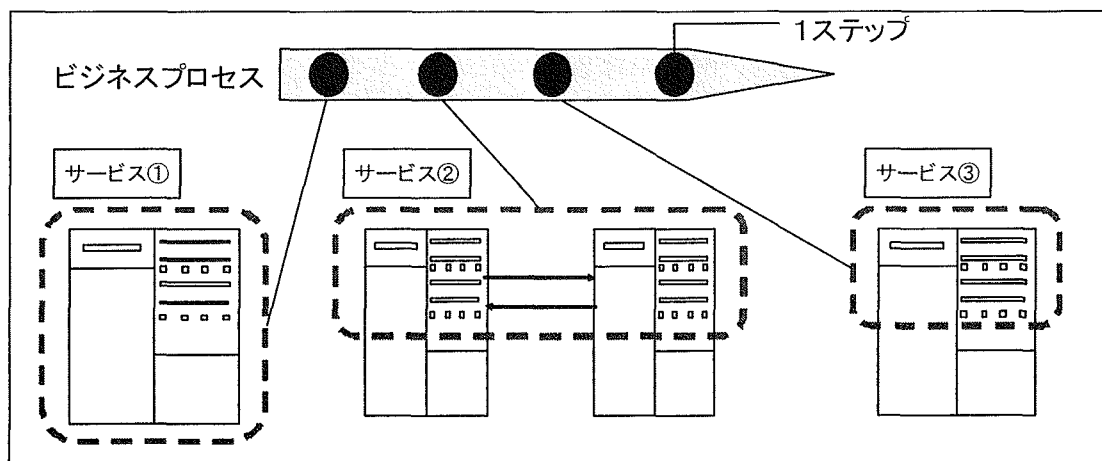
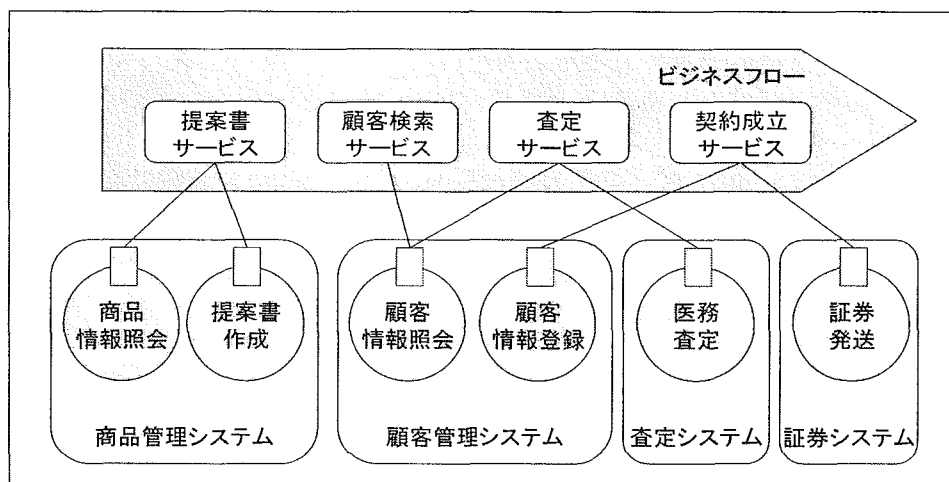


図 II-3 サービスとシステムのマッピング概念図

これを保険業界に置き換えると以下ようになる。

- ・提案書サービスは、商品管理システムの商品情報照会コンポーネントと提案書作成コンポーネントで構成されている。(図Ⅱ-3：サービス①の例)
- ・顧客検索サービスは、顧客情報照会コンポーネントでのみ構成されている。(図Ⅱ-3：サービス③の例)
- ・査定サービスは、顧客管理システムの顧客情報照会コンポーネントと査定システムの医務査定コンポーネントで構成されている。(図Ⅱ-3：サービス②の例)
- ・契約成立サービスは、顧客管理システムの顧客情報登録コンポーネントと証券システムの証券発送コンポーネントで構成されている。(図Ⅱ-3：サービス④の例)



図Ⅱ-4 保健業界におけるサービス概念図

### 3. SOA導入によるメリット

SOA導入によるメリットとして以下の3点が挙げられる。

#### (1) ビジネスプロセスの変更が容易

構成する単位、つまりサービスをビジネスプロセスの処理単位とすることで、それらの組み換えだけでビジネスプロセスの変更に対応することが出来、適応性や俊敏性を社内システムに取り込むことができる。

現在、取引先の変更、アウトソーシング、事業の拡大／撤退等々の様々な変化要因が存在するため、ビジネスプロセスは日々変化する。しかし、各プロセスを構成するエンティティの変化はそれほど多くはない。よってそれらをサービスという自立した単位で構成しておくことにより、ビジネスプロセスが変更されても、サービスの呼び出し方法だけを変更すれば対応が可能となる。

### (2) 資源の再利用性が高い

サーバーやメインフレーム等々のインフラに依存せず、どのようなアプリケーション、どのようなOS、どのような言語であってもサービス同士のコミュニケーションが可能である。

また、SOAによりシステムを構築していくということは、既存のシステムを整理していくことに繋がる。システムの整理が進んだことにより、再利用可能なサービスが増える。

### (3) サービス内での変更が容易

SOAではサービスの機能を定義するインターフェースとサービスの実装は分離されている。このため、インターフェースに定義されたサービス呼び出しのロジックは修正することなく、サービスの実装、つまりシステムコンポーネントを変更することが可能となる。この利点により長期的な視点に立つシステム計画とうまくシンクロナイズできるはずである。

第II章ではSOAの定義、基本概念、メリットを説明した。SOAの定義や基本概念は、具体的に想像することが難しい考え方であるが、次章からSOAを社内システムに導入する方法を具体的に説明することにより、SOAがどのような考えのものであるか、更に説明を行っていく。また、その説明によりSOAの導入によって、保険会社がどのようなメリットを享受するかを具体的に提示する。

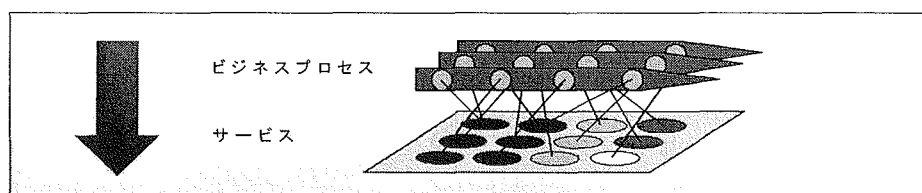
### 第Ⅲ章 SOA導入手法

前章ではSOAに関する基本概念を解説した。では、SOAを導入するためにはどのようなアプローチをすればよいのか。本章はその疑問に答えるべく、SOA導入の代表的アプローチ方法を3つ紹介する。それぞれにメリット/デメリットが存在するが、これを分析した上で比較検証を行い、「最も現実的なアプローチ方法」の提案を目的としている。勿論、導入手順の前提となることは言うまでも無い。

#### 1. トップダウンアプローチ

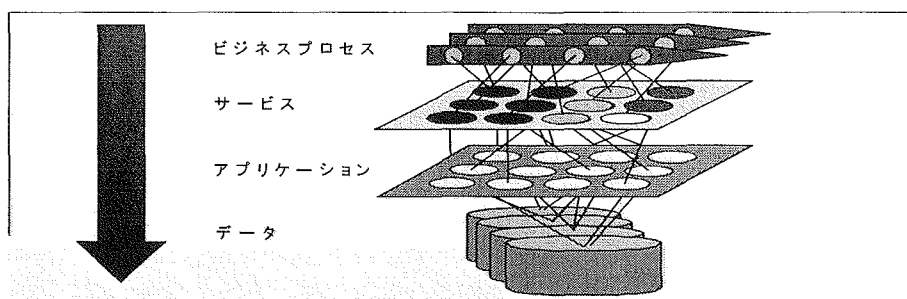
##### (1) 基本的な考え方

トップダウンアプローチは業務改革型アプローチとも呼ばれ、既存システムやシステムの処理プロセスを考慮せず、ビジネスの視点から全く新しいサービス群を構築する際に特に有効な手法である。



図Ⅲ-1

まず、社内全体のビジネスプロセスを分析、可視化する。次にそれぞれのビジネスプロセスを最適化した上でサービスを定義し、最後にアプリケーション、データ項目の構造定義をしていく。ここで、サービスとアプリケーションは必ずしも1対1ではなく、N対Nの関係になることもある。つまり、複数のアプリケーションで一つのサービスを構成することもあれば、一つのアプリケーションで複数のサービスを構成することもありうるということである。さらにデータも最適化されているため、アプリケーションとデータはN対1で結びつくことになる。



図Ⅲ-2

##### (2) メリット/デメリット

トップダウンアプローチの最大のメリットは、高い投資対効果にある。社内全体の業務の可視化、最適化を行った上でサービス定義を行うため、サービス

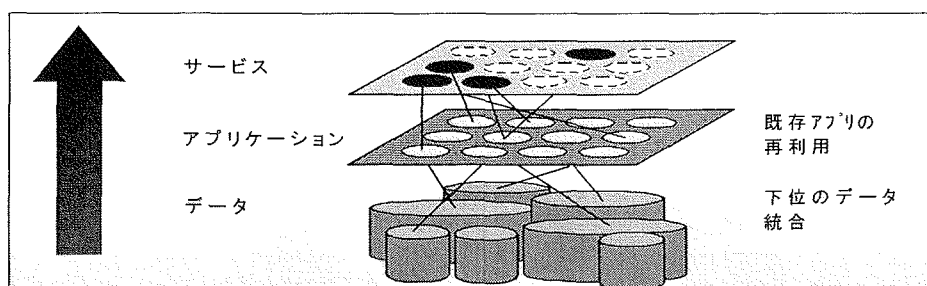
の再利用性は非常に高いものになる。そのため、システムの柔軟性も大幅に向上するとともに、システムのスリム化によるコスト削減効果を最大限に得られるため、自ずと高い投資対効果を得ることができる。

一方、デメリットは導入コストとリスクの大きさにある。膨大な業務処理を分析し、共通サービスを抽出する必要があるため、規模、つまりコストが非常に大きくなる傾向がある。また、社内組織的にも少なからず変化が発生するため、業務への影響とそのリスクも規模に比例して大きくなる傾向がある。

## 2. ボトムアップアプローチ

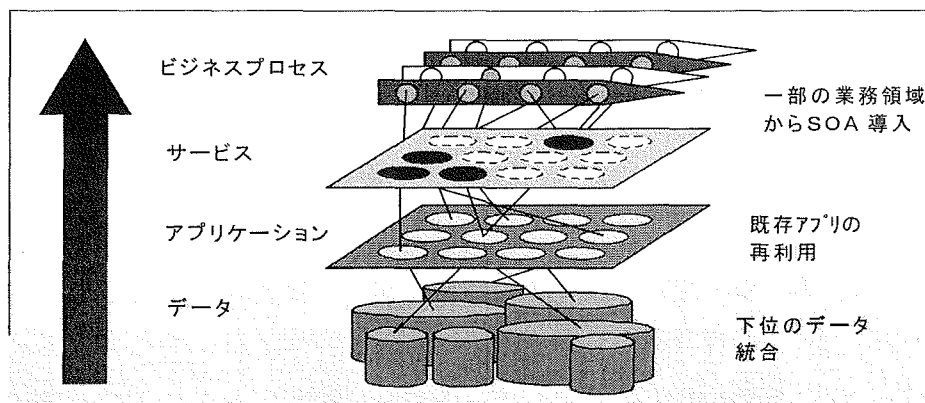
### (1) 基本的な考え方

ボトムアップアプローチは、基盤／開発環境整備型アプローチとも呼ばれ、前述のトップダウンアプローチとは対照的に、システム部門主導で現状のシステム構成の見直し／整備を基礎として、サービスを構築していく手法である。



図Ⅲ-3

まずシステムの最下層であるデータの統合／最適化を行い、次に、既存アプリケーションをアダプタやサービスという形でラッピングしていく。アプリケーションは既存のものを再利用する。最後に既存のアプリケーションが提供するインターフェースから、ビジネス要件を抽出し、定義されたビジネスプロセスの業務領域から徐々にシステム構築をしていく。



図Ⅲ-4

このように、利用できるサービスをシステム側から識別し、それらを組み合わせ再利用することで開発コストを抑え、かつフレキシブルな基盤構築／統合管理を進めることができるのが、ボトムアップアプローチの考え方である。

### (2) メリット/デメリット

ボトムアップアプローチのメリットは、既存システムの資源を最大限活用できることにある。既存資源を再利用することで、新しいサービスを短期間、且つ低コストに構築することが可能となる。

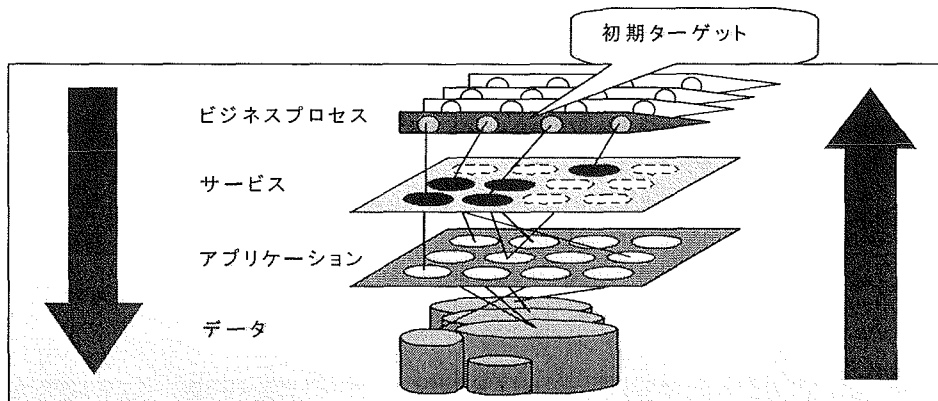
一方、データ層からサービスの構築をはじめていくため、最終的なサービスの粒度が下位層に依存してしまい、適切なサービスの粒度が決めにくいというデメリットが挙げられる。また、ビジネスプロセスの見直しをすることができないため、ビジネス上のインパクトは必ずしも大きなものとはならない。

導入コストの削減や、全システムを再構築する期間／コストが取れない場合、つまり、プラットフォームを構築した上で、段階的に各アプリケーションを実装していく場合に、特に適した手法であると言える。

## 3. 複合アプローチ

### (1) 基本的な考え方

複合アプローチとはトップダウンとボトムアップ両方のアプローチをバランス良く同時並行で進めていく手法である。



図III-5

構築対象を特定の業務に限定し、「システム変更」や「新規開発」という手段を通じて業務分析と、部分的なシステム構築を行っていく。これと並行して、システムの再利用性を高めるための基盤整備を進めていく。複合アプローチでは、一つ一つのサービスを一定の粒度に保つために、常に最終的に構築すべきシステムを見定めていくことがとても重要な要素となる。

### (2) メリット/デメリット

次項にて論ずる。

#### 4. 評価

これら3つのアプローチについて、初期コスト／導入リスク／投資対効果の3点から評価する。

SOAに基づくシステム構築を実施するにあたり、トップダウンによるビックバン的な導入は、最大限の効果が導入当初から得られる反面、コスト／リスクが共に高く現実的では無いと言える。また、ボトムアップでは将来的なプロセス間連携や統合に向けた準備は進められるが、ビジネス上のインパクトは必ずしも大きなものとはならず、SOAによるシステム構築のメリットを最大限に得られるとは言えない。

実際にSOAに基づくシステム構築を行うにあたり、既存資源を活用しながら業務改革を進め、最低限の開発規模にてシステム構築を行っていくことが、プロジェクトのリスクを低減させ、コストをコントロールしやすいという面で最も現実的と言える。

手法	初期コスト	導入リスク	ROI
トップダウンアプローチ	HIGH	HIGH	HIGH
ボトムアップアプローチ	LOW	LOW	LOW
複合(トップダウン・ボトムアップ)アプローチ	LOW	LOW	HIGH

図 III-6

よって、IT研究会第3グループでは、SOAの導入手法としてトップダウンとボトムアップの両方のアプローチを、バランス良く同時並行で進めていく「複合アプローチ」が望ましいと結論付けた。

トップダウンとボトムアップの両アプローチによって、SOA基盤の構築と、限定的な範囲のアプリケーションの実装をスタートさせ、その後は実績を基に段階的に領域を拡大して、新たなサービスを配備する方法が、SOAに基づくシステム構築成功への現実的な近道であると考えられる。

## 第IV章 保険会社におけるSOA導入案

前章までで、SOAに関する基本概念、導入に対するアプローチ方法について解説した。この章では、「SOAに基づくシステム構築」について一つの例を挙げ、具体的なシステム構築の手順について解説を行う。ここでは、ストーリーに具体性を持たせるため、各保険会社アンケートの集計結果を元にモデル化した「A社」をおき、「A社」の抱える問題点／課題点の解決手段として「SOAに基づくシステム再構築」を選択するという前提で話を進める。

### 1. 「A社」の前提

導入手順の解説の前に、まずはこのストーリーの舞台となる「A社」の現状について、以下の通り整理する。SOAは業務とシステムが抱える問題に対する有効的な解決手法の一つであるということは、今までで述べてきたとおりであるが、導入前後の効果測定のために現状の整理は必要不可欠なものだからである。

#### <営業形態>

- ・古くから第一分野保険を主力としていた保険会社である
- ・第一分野保険の管理はメインフレームを使用している
- ・社会の動向に対応すべく、数年前より第三分野保険の販売を開始した
- ・第三分野保険については、技術革新が著しく、且つ「流行でもある」オープン系システムにて管理をしている

#### <経営サイドの方針>

- ・組織と業務の全社最適化によるコスト削減
- ・独自性を意識したビジネスモデル構築

#### <事務部門の抱える問題／課題>

商品競争力強化や顧客満足度向上のため、新規商品や業務改善案を検討してはいるが、開発コストや期間がネックとなり、対応が見送られることや、仮にシステム対応されたとしても100%の要望を満たされないことによる不満を抱えている。

⇒自社システム開発の機動性の低さから、経営から与えられているミッションを達成できない

#### <IT部門の抱える問題／課題>

構築当初から、綿々と保守開発を行ってきたことから、システムの肥大化／複雑化を招いており、継続的な保守を考えると多くの問題点を抱えている。これにより事務部門から挙げられる多くの業務要件を満たすためには、膨大なコスト、長期的な時間が必要となるため、開発を見送る、もしくはスコープを縮小するケースも多くなってきている。

このように、「A社」は多くの保険会社が同様に抱えているであろう問題に四苦八苦している、正に「ありふれた」保険会社である。昨今の「競争力強化」のための商品の複雑化や、市場動向に敏感に対処するための短期的な新規商品の市場投入、「CS向上」のための多くのサービスの展開により、システムは肥大化／複雑化の



一途をたどっている。過去、事務負荷の軽減により多くの恩恵を受けてきた営業／事務部門は、現在、システムが「足枷」となり、自信のある商品やサービスの展開を足踏みしなければならない状態となってしまうている。

このような現状を鑑み、経営サイドはIT部門に対して現状打破を強く要望し、IT部門は「システム再構築」を選択するのはごく自然な流れであると言えるであろう。

## 2. 構築に向けた検討

一言に「システム再構築」と言っても、事はそれほど単純な話ではない。システムの全面再構築を行うためには多くのリソースや期間が必要であり、その他の開発案件はストップ、若しくは大幅に減少せざるを得ない。当然ながら、経営層はそのようなことを了見できるわけもなく、IT部門にとって頭が痛いところである。

このような状況を踏まえ、「A社」IT部門は新基幹システムの構築と、既存システムのパイロット移行を基本方針に検討を進めることとした。この時、IT部門の上層部の目に留まったのが「SOA」であり、再構築の検討が予め行われていた新契約システムを「パイロット移行」の対象とすることで、方針が固まったのであった。

## 3. 「A社」新契約システムの現状と問題点

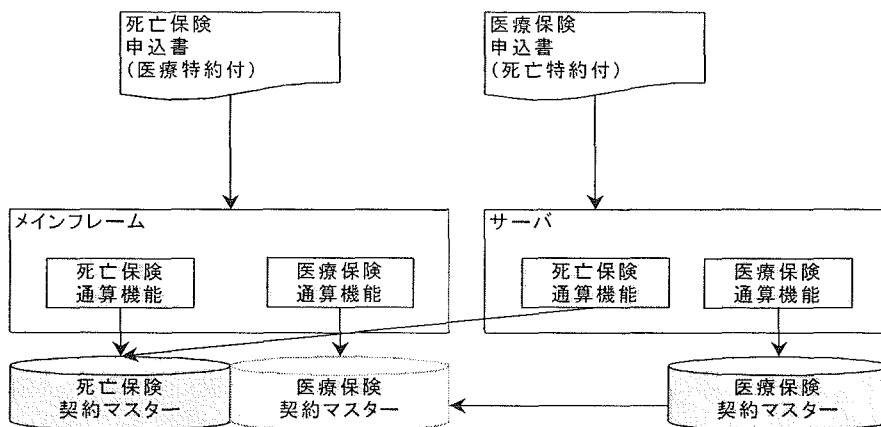
パイロット移行の対象システムの決定により、具体的な移行計画の検討となる。ここでは新規システムの青写真を描くためにも、「A社」新契約システムの理想と現実のギャップ分析について、述べておく。

このギャップ分析の結果が、SOA導入に基づくシステム再構築にて解決すべきターゲットとなり、これを如何にして解決できるのかが、「SOA」導入／評価のポイントとなる。

### (1) システム機能の重複

「A社」では第一分野保険をメインフレームで、第三分野保険をサーバーで管理しているため、自ずと事務機能に対してシステムが2系統存在することになっている。

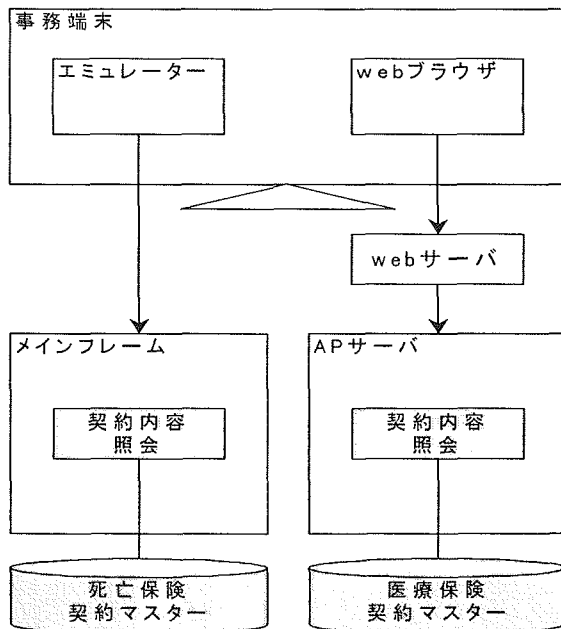
以下の図IV-1を例として見て頂きたい。新契約加入時の通算チェックについて、第一分野／第三分野に共通する保証部分を通算／限度額チェックする必要があるため、メインフレームにも第三分野商品の通算機能を、サーバーにも第一分野商品の通算機能を重複しておいている。この「同機能」を両システムに実装しなければならないことから、仕様変更時に二重の開発をしなければならないという、開発効率の低下を招いている。(図IV-1参照のこと)



図IV-1

(2) 一本化されていないユーザーインターフェース

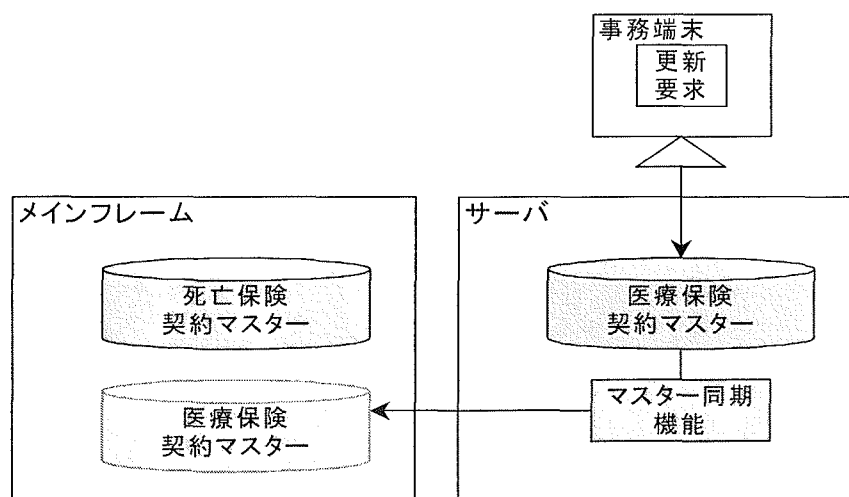
前述のとおり、第一分野保険と第三分野保険でシステムだけではなく、インフラについてもメインフレームとサーバーとで分かれていることもあり、ユーザーインターフェースがメインフレーム=エミュレーター、サーバー=webブラウザと、統一されていない。「契約内容照会」というオンライン照会を例にあげると、第一分野保険の契約内容をオンライン照会する場合はエミュレーターを使用し、第三分野保険の契約内容をオンライン照会する場合にはwebブラウザを使用することになる。これが事務方の非効率化を招いている。



図IV-2

### (3) マスターDBの2重管理

メインフレーム／サーバーでの機能重複に伴い、この「機能」にて使用するマスターDBを双方のインフラにて管理することが必須となってしまっている。サーバーに設置されているDBが本来のマスターであるが、当然ながら、メインフレーム側のDBのデータ鮮度を保つことも機能要件となるため、マスターの同期機能を構築せざるを得ない状況となっている。これらは単純に、「マスターDBの二重管理」と言うデータ管理リスクのみではなく、サーバー側のDBレイアウトを変更した場合に、この機能とメインフレーム側のDBの拡張を行わなければならないと言う、開発工数の増大をも生んでいる。



図IV-3

## 4. 導入

### (1) 新契約業務の要素分解とサービスの定義

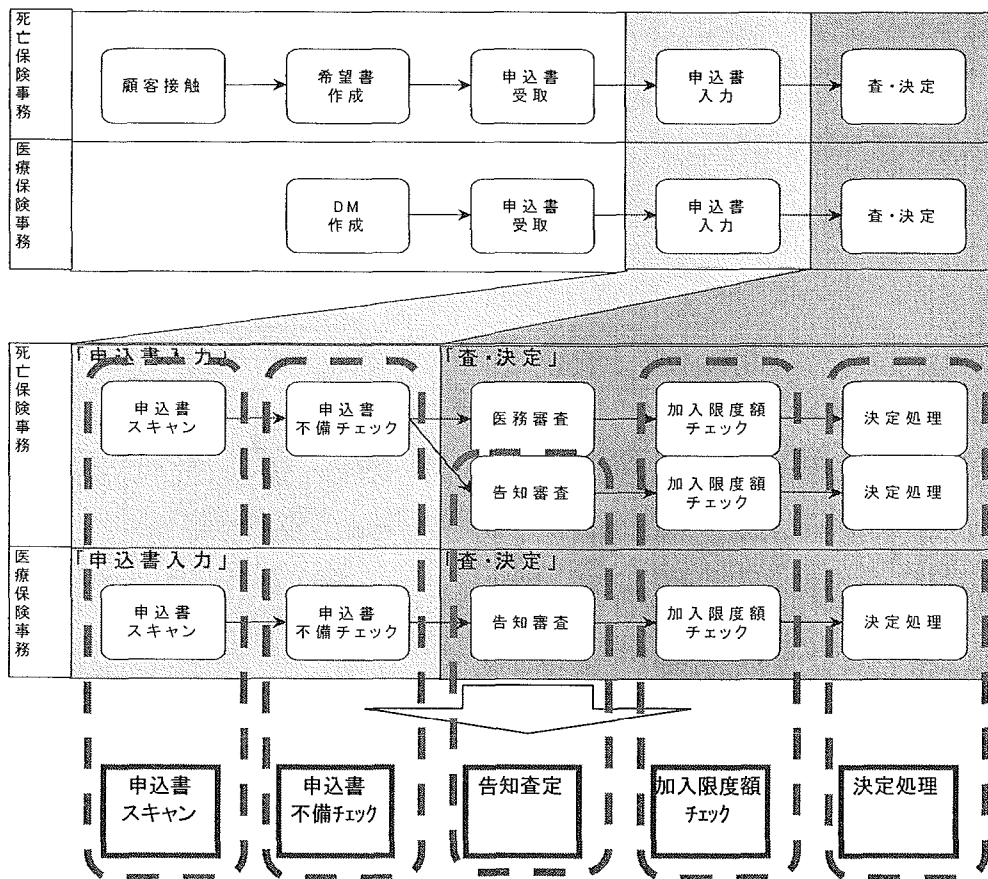
新契約システムを利用する「新契約業務」について、業務要素単位への分解と、順序化（フロー化）を行う。「A社」の場合、図IV-4のように、まず新契約業務を「死亡保険事務」「医療保険事務」に分解した上で、各々の事務フローを細分化している。例えば「申込書入力」という業務は「申込書スキャン」と「申込書不備チェック」という2つの業務に分かれている。

業務要素の分解の次は、分解された要素の統廃合を行う。このステップが最終的な「サービスの定義」に繋がることを念頭におくことが重要である。業務の最適化という観点でも、システムの重複機能の排除という観点でも、同種の業務要素が複数存在する事実は弊害となるため、分解された業務要素の重複分のみでなく、その他の業務要素自体に類似点がないことを一つ一つ確認するのが重要である。場合によっては要素分解のステップに立ち返り、要素の粒度を変更した上で、このステップを進めていくことになる。

全ての業務要素の最適化が終了した後存在している一つ一つの要素が「サービス」となり、この粒度が今後、「A社」にて維持されるべき「サービスの粒度」と

なる。

また、この手順で最も重要なのは、事務部門主導にて行われること、若しくはIT部門の主導であったとしても事務部門の全面的な協力を得られる体制を構築した上で臨むことである。



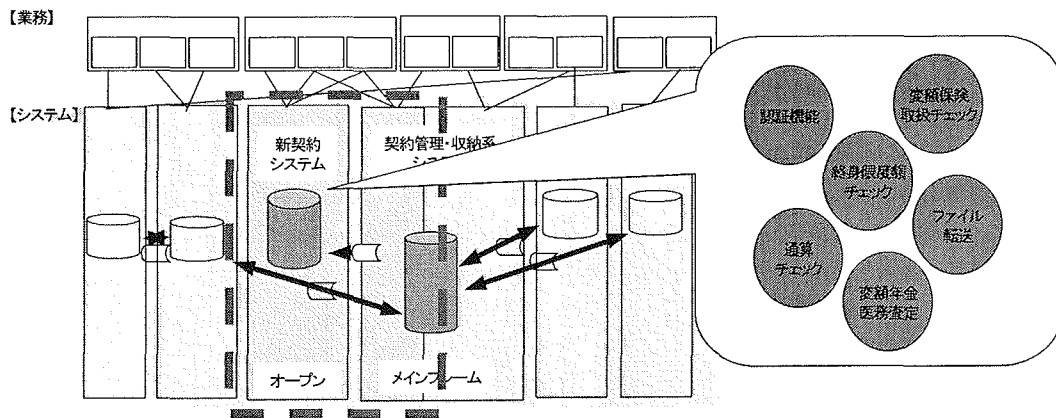
図IV-4

## (2) 現行システムのコンポーネント化

業務面からのアプローチが終了した後は、システム面からのアプローチとなる。このステップでは、現行システムに実装されている機能を分析、再利用可能な単位に分解することから始まり、重複機能を統廃合し、最終的には各機能が可能な限り疎結合となるコンポーネントとするところまでを行う。

ここで重要なのは、新契約システムのための機能分解ではなく、最終的に全システムを移行することを視野に入れた上で、他システムとの重複機能の洗い出しを行うておくことにある。このステップの範囲に含めるシステム数が多ければ多いほど、作成されるコンポーネントの再利用性は高くなるという点を忘れてはならない。

(図IV-5 参照のこと)

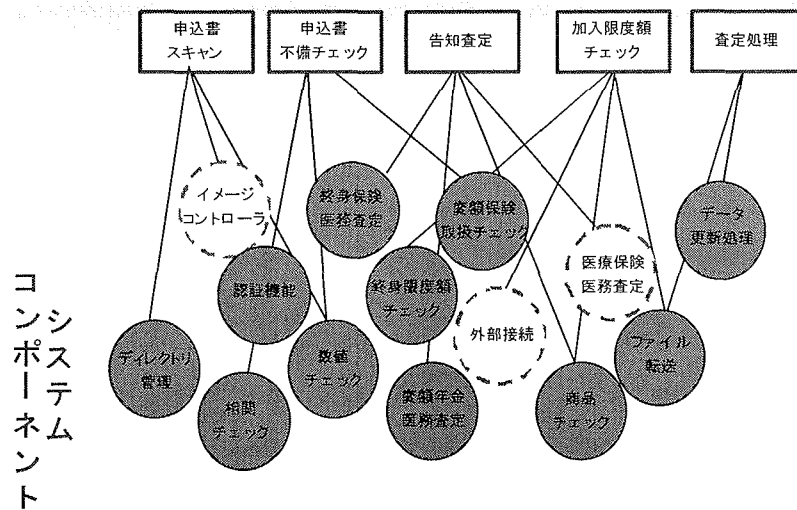


図IV-5

### (3) 「サービス」のシステム設計

このステップでは、「サービス」と、それらを実現する「コンポーネント」のマッピングを行う。一つ一つのコンポーネントの組み合わせを行い、「サービス」を実現するサブシステムを構成するようにする。この作業を行う中で不足するコンポーネントが発生した場合、新規コンポーネントを構築するの可否かについての検討如何で、サービスの粒度に変更が生じる、またはコンポーネントの分類に変更が生じるため、適切に判断することが重要になってくる。

サービス



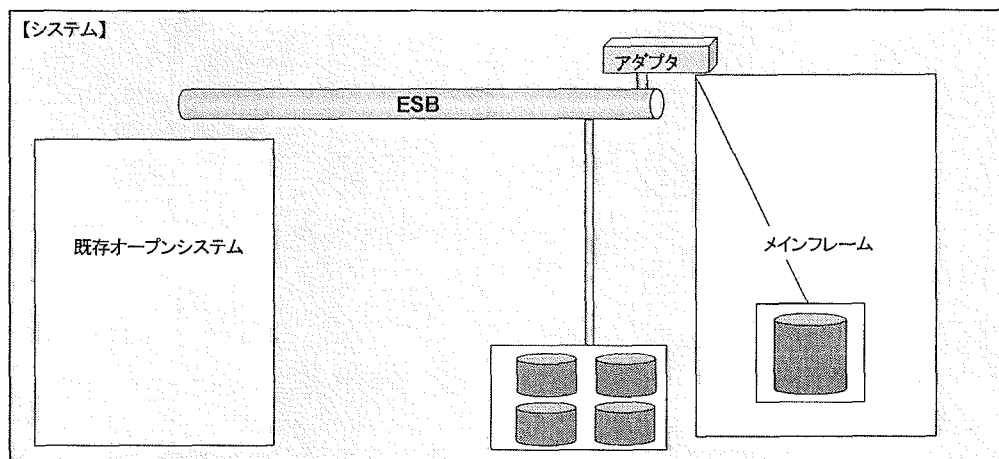
コンポーネント

図IV-6

### (4) インフラ基盤の構築

ここからはいよいよSOAの実装が始まる。第一ステップとして行うのは、SOAという概念に不可欠である、インフラ/言語に左右されないシステム間通信を実現するための、「交通整理役の導入」である。

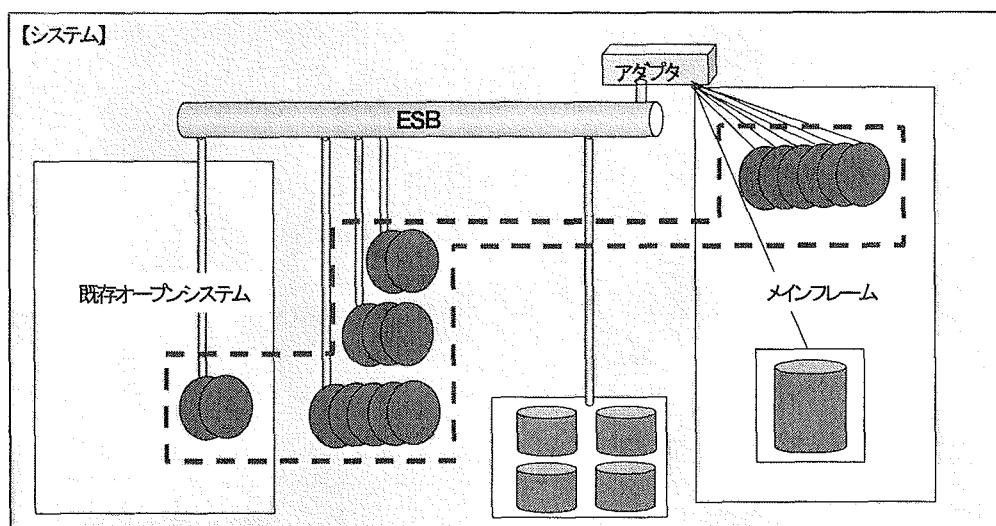
ESB (Enterprise Service Bus) の実装と、接続のためのアダプタの設置がこれにあたる。また、「A社」の場合、手順には載せていないが、DBの再構築も同時に実施するスコープとしているため、このステップにてDB環境の構築も平行して行っている。



図IV-7

#### (5) サービスの実装

インフラ環境が完成したら、次のステップとして、(3)にて定義されたシステムコンポーネントの実装を行う。各システムコンポーネントを適切なインフラ環境に設置し、これらをESBに定義することで、ESBに接続する全てのインフラ環境にて利用可能となる。

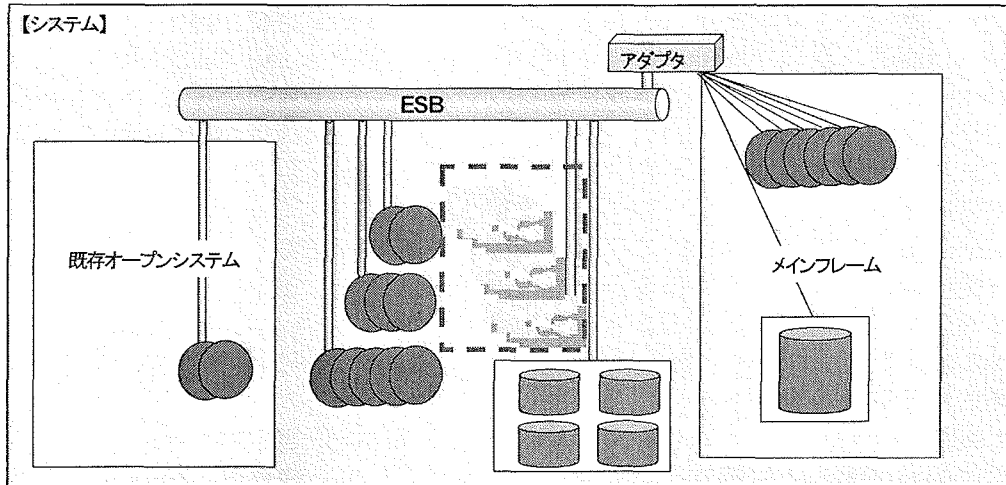


図IV-8

#### (6) BPELエンジンの実装

システムコンポーネントを実装したことによって、業務機能の実現可能な環境は

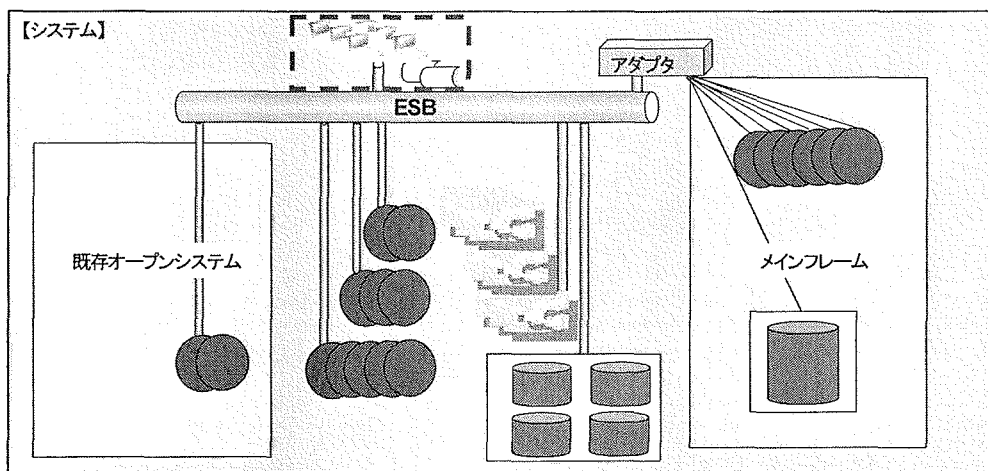
整い、このステップではそれらのシステムコンポーネントを適切に選択／コントロールするための管理機能（コリオグラファ）の実装ステップとなる。一般的には「BPELエンジン」と呼ばれるソフトウェアの導入をすることとなる。



図IV-9

(7) インターフェースの実装

いよいよ最後のステップとなるインターフェースの実装である。外部システムインターフェースやGUIがこれにあたる。既にシステムコンポーネントとコリオグラファの実装が済んでいるシステムにおいては、GUIはビジネス・ロジックとは完全に分離したものとなり、単に、サービスを利用するための窓口となる。



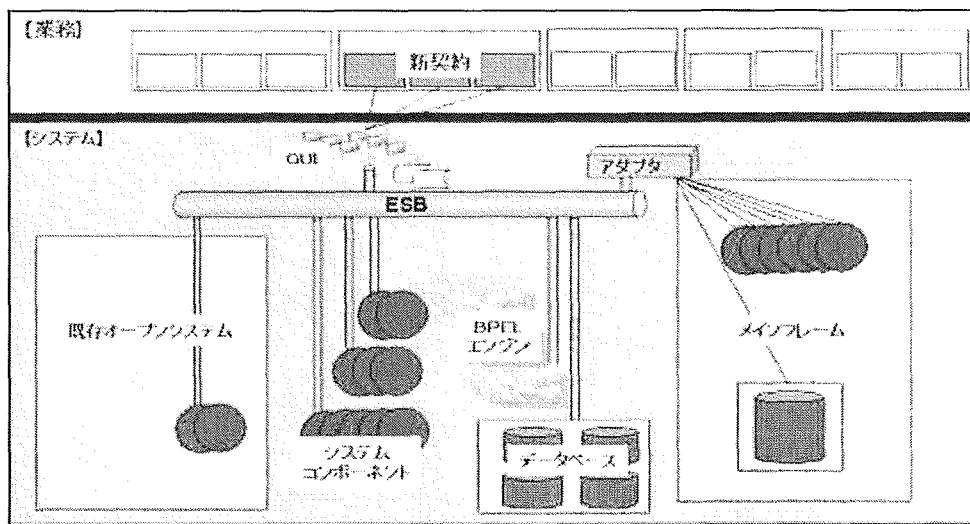
図IV-10

## 第V章 SOA導入後のシステムについての検証

### 1. 部分的なSOA導入による効果

複合アプローチやボトムアップアプローチにより段階的なSOA導入を行った場合、SOAによる新システムと既存システムの並存期間が発生する。よって、ここではSOA導入後に「A社」の抱えていた問題点／課題点がどのように解決されたのかを検証しながら、一部分のシステムにSOAを導入した場合の効果について考察する。

図V-1はSOA導入により新契約システムを再構築した「A社」のシステム構成である。新契約システムが持つ機能と新契約システムが必要とするメインフレームでの機能は全てシステムコンポーネント化されESBに接続されている。メインフレームとESBとの通信はアダプタを介して行われており、システムコンポーネント間の通信プロトコル等の違いはここで吸収される。また、それらのシステムコンポーネントが利用するデータも正規化され、これもまたESBに接続されている。そして、BPELエンジンがサービスとしてのシステムコンポーネント群を操作し、新契約業務機能を実現する。さらにGUIや外部システムインターフェースもESBに接続されており、BPELエンジンによる一連の処理を呼び出したり、サービス単体を呼び出したりすることも可能となる。ここで再構築されたシステムはすでに『新契約システム』という枠組みではなく、新契約という業務とつながっているのはGUIのみであり、システムと業務とのつながりは完全に疎結合となっている。

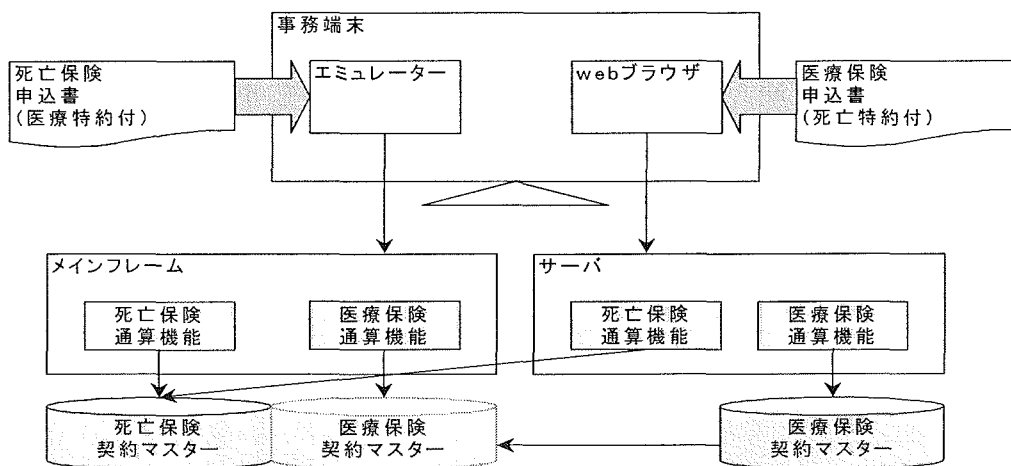


図V-1 A社のシステム構成図(新契約システム再構築後)

次に、「A社」の新契約システムが抱えていた問題点／課題点がどのように解決されたのかを検証する。図V-2はSOA導入前の「A社」の新契約システム構成図であり、先に述べた通り、以下の3つの問題を抱えていた。（図V-2参照のこと）

- ・ 機能重複による開発効率の低下
- ・ 一本化されていないユーザーインターフェースによる事務効率の低下
- ・ DBの2重管理によるデータ管理リスク





図V-2 「A社」新契約システム構成図(SOA導入前)

一方で、図V-3はSOA導入後の「A社」の新契約システムである。旧システムが抱えていた問題点／課題点と対比すると以下の効果が確認できる。

(1) 機能の収斂による開発効率の向上

旧システムでは事務機能に対してシステムが2系統存在していた為、新契約加入時の通算チェックがメインフレームとサーバーの双方に配置されていたが、図V-3の通り、新システムでは事務機能とシステム基盤の間にサービス層が配置されたことで、事務端末はサービス層にあるサービスを呼び出し、そのサービスはE S Bを介することでシステムの物理的な配置を意識することなく、通算チェック機能呼び出すことができるようになった。これにより、メインフレームには死亡保険通算、サーバーには医療保険通算が重複することなく配置され、例えば通算日数や限度額の変更の際も、最小限のシステム修正で対応可能となった。修正モジュールが2つから1つになったこと、二重開発による煩雑なプロジェクト管理が不要となったことなどを考慮すると、開発効率は倍以上になったと考えられる。

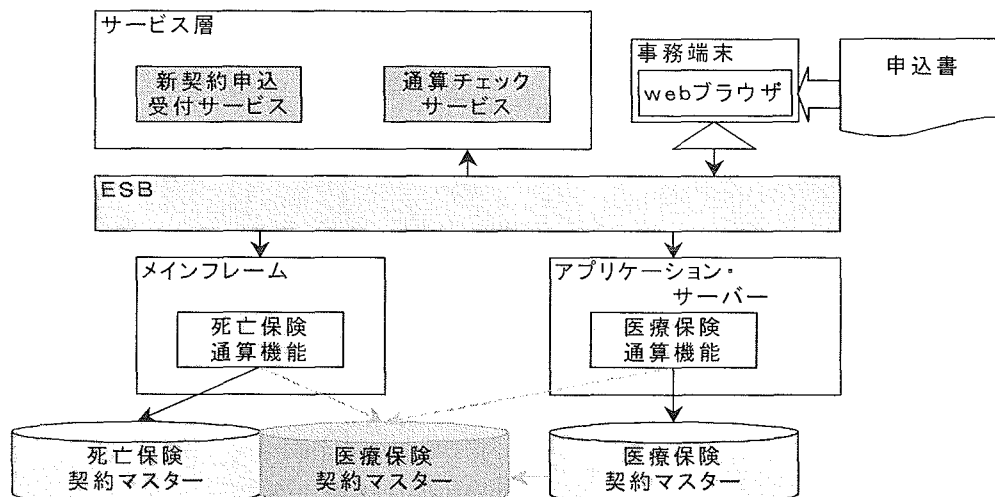
(2) ユーザーインターフェースの統一による事務効率の向上

事務端末のGUIがWEBに統一された為、第一分野保険の契約内容をオンライン照会する場合はエミュレーターを使用し、第三分野保険の契約内容をオンライン照会する場合にはWEBブラウザを使用するという面倒なユーザーオペレーションは不要となる。さらにユーザーインターフェースとビジネスロジックが分離されたことでユーザーインターフェースのレスポンスも向上し、異なるインフラ上にあるデータを一覧化させたり、混在表示させたりとユーザーにとってより使いやすい構成が実現され、その結果として事務効率が向上する。

(3) データの最適化によるデータ管理リスクの低下

機能が収斂されたことにより、メインフレームに医療保険契約マスターを配置す

る必要はなくなった。これに伴い、データの同期機能も不要となる。さらにデータをモデル化し最適化を行ったことで他システムからも利用しやすいデータとなり、システム毎に独自データベースを構築して同期を行うなど、新たな重複の発生を防ぐことができる。また、昨今では個人情報保護の観点から契約情報などの個人情報を保持する場合は、相応の管理が必要とされるが、データを極小化することでこれらの管理コストも削減可能となる。



図V-3 「A社」新契約システム構成図（SOA導入後）

以上より、SOA導入によって「A社」の新契約システムが抱えていた問題点／課題点はほぼ解決された。しかし、A社のシステム全体を見た場合には一部分の問題が解消されたに過ぎず、異なるアーキテクチャーの混在による管理コストの増加などにより、実質的なコストダウンや開発期間の短縮には繋がっていないことも考えられる。現実的には、SOA導入による恩恵を最大限に享受できるのは、社内のシステムの大部分が再構築された後となるであろう。

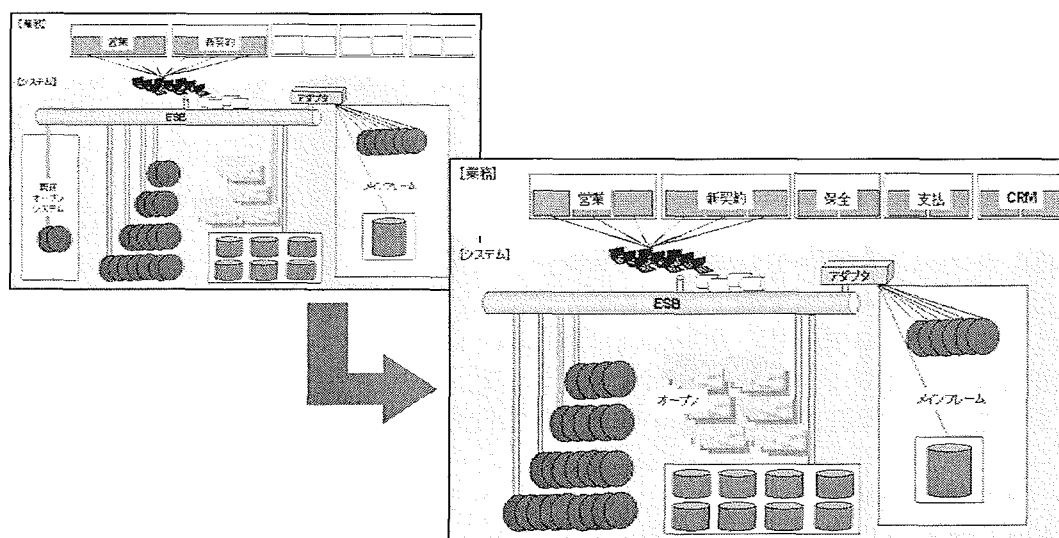
## 2. 漸進的な移行

SOAを導入していくにあたり、現実的かつ効率的な方法は漸進的な移行方法である。

新契約システムへのSOA導入により、ESBやBPELエンジン、メインフレームとのアダプタなど、必要となるインフラ基盤はすでに構築されている。ここで、新システムと現システムは相互補完的に並存するが、アーキテクチャー上は、巨大な現システムは小さな新システムの一部となる。未対応のシステムについては、再構築や追加機能開発の際、構築済みの部分はそれを再利用し、まだ存在しないシステムコンポーネントやデータベースを新規構築すればよい。既存システムに無い機能のみを補完する為、新機能構築については比較的短期間で構築することが可能である。

さらにインフラ基盤が整備されている為、初期導入時のような案件の規模は不要

となり、膨大なIT資産を有効に利用しながら新アーキテクチャへ移行できるだけでなく、最終的な新システムへの全面移行を目指して、漸進的に移行していくことが可能である。



図V-4 漸進的な移行

### 3. SOA導入による会社組織への影響

SOAを全社的に導入したことによる会社組織全体への影響を、「A社」の各部門への影響を例に階層的に検証すると、以下の通りとなる。

#### (1) IT部門への影響

システムの肥大化／複雑化により、膨大なコストと期間を浪費して事務部門の業務要件を満たしてきたが、システム機能が最適化され、データベースが正規化されたことにより開発コストと期間が極小化された。

#### (2) 事務部門への影響

開発コストや期間がボトルネックとなり、企画した新規商品や業務改善案をプラン通りに実現できていなかったが、システム対応が迅速かつ低コストで実施できるようになり、さらにSOA導入プロセスの中で複雑なビジネスプロセスが可視化／簡素化された為、生産性が向上した。結果として商品競争力が強化され、顧客満足度が向上した。

#### (3) 経営サイドへの影響

最もボトルネックとなっていたIT部門の開発コストや期間が改善された為、事務効率が向上し、結果として重要課題である『組織と業務の最適化によるコスト削減』、『独自性を意識したビジネスモデル構築』が実現された。

## 第VI章 SOA導入への提言

### 1. 組織全体での取り組み

SOA導入はIT部門のみでの実施は不可能である。先の導入手順でも述べたとおり、SOAの導入には、ビジネスプロセスが可視化されていることが前提であり、サービス定義も業務視点で行う。よってIT部門だけでなく、経営サイドや事務部門の協力が必要不可欠である。

### 2. 長期的な計画と継続的な実施

SOAの導入には長期的な計画と継続的な実施が必要である。SOA導入はソフトウェアのパッケージを購入し、自社向けのカスタマイズをして半年で導入というものではない。海外の事例では、5年かけてSOAを導入したという例もある。SOAを導入する過程で目に見えた効果を期待しがちであるが、一部の業務にSOAを導入しただけでは、もたらされるメリットも一部分となる。最大限のメリットを享受できるのは全社的にSOAが導入された時である。

### 3. 管理体制の整備

SOA導入によってシステムを構築した後、「サービス」を管理する部門を明確にし、全社的な統ルールを作成し、これに則りサービスをコントロールしていく必要がある。サービスをコントロールする上で重要となるポイントは以下の通りである。

- ・サービスの管理による独立性／再利用性の維持

サービスの独立性が維持できない場合、ある部署で定義した「サービス」の業務内容が、他部署で定義した「サービス」と重複するなど、既に存在しているサービスを誤って二重に定義してしまい、運用／管理コストに悪影響を及ぼす可能性がある。

- ・統一されたサービスの粒度の維持

一つの事務部門がサービスの定義を担当すると、自部署の業務に偏ったサービスばかりが出来てしまい、サービスの再利用性低下を招く可能性がある。このような事態を避ける為、サービスの粒度を一定に統一する必要がある。

- ・「サービス」からのシステム性排除

システム部門のみがサービス定義を担当した場合、業務視点からはかけ離れた粒度のサービスを定義してしまい、サービスの再利用性を著しく低下させる可能性がある。よって、「サービス」からは可能な限りシステム性を排除することが望ましい。

このように適切にコントロールされていないサービスの再利用性は非常に低くなり、SOA導入のメリットである柔軟なシステム構築やコスト削減効果に深刻な影響を及ぼす。そのような状況を避ける為、決められたルールに則り全社的にサービスをコントロールしていくことは必須である。

あえて一言に纏めるならば、「組織全体」で「やり遂げる」ことである。

## おわりに

今回の研究メンバーは全員、IT部門にて何かしらシステム開発に関連する業務に携わる人間であり、IT研究会という名目からも、SOA (Service Oriented Architectures) という題目からも、研究開始当初は「技術」に関連するテーマであると言う想定のもとに進んでいた。当然、7割方は技術的な研究であったが、残りの3割は会社の枠を超え、自社システムの抱える問題点を共有し、強いては保険業界の行く末を睨み、ビジネスに対して「SOAがどのような価値を創造できるのか？」という話に、いつの間にかメンバーの焦点が合っていた。

単純に「技術」のみの話で終わってしまっていたら、ひょっとするとSOAに関する理解と、SOAの有用性について、メンバーの認識は高いものにはならなかったかもしれない。そしてメンバーの意識がビジネスにまで向いたからこそ、SOAとはある特定のサービスを提供するのではなく、今後のビジネス戦略において、IT部門が受けるであろう「様々なリクエストを迅速に対応可能とするシステム環境構築のために最適な手法である」という結論にたどり着けたと自負している。

SOAという考え方そのものは決して新しくはない。この方法論に技術がついていけなかつただけである。SOAを支える技術基盤が急速に進歩している今、いよいよこの「夢物語」が実現可能なレベルに到達しつつある。今後、より多くの保険会社がSOAを導入し、「研究」としてではなく「現実」として、その有用性が実証されていくことを期待すると共に、その時に当事者として1年間の研究の成果を発揮できればと思ってやまない。

最後にこの場をお借りして、1年間の研究会を支えていただいたIT委員の皆さまと、技術支援を頂いた日本アイ・ビー・エム株式会社の方々、また、アンケートにご協力いただいた各社ご担当の皆様にお礼を述べさせていただき、この論文を締めくくることとする。

<参考文献/資料> ※敬称略

書籍名	著者
SOA—サービス指向アーキテクチャー	日本BEAシステムズ株式会社
日経コンピュータ 2006.2.6号	矢口竜太郎 田中淳 著
これだけ知って欲しい SOA第一回 <a href="http://www.bcm.co.jp/site/soa/soa01/html">http://www.bcm.co.jp/site/soa/soa01/html</a>	アクセンチュア株式会社 野本義博
これだけ知って欲しい SOA第二回 <a href="http://www.bcm.co.jp/site/soa/soa02/html">http://www.bcm.co.jp/site/soa/soa02/html</a>	アクセンチュア株式会社 立花良範
SOA実現に向けて 基礎からわかるSOA	日本アイ・ビー・エム株式会社 (技術支援) 米持幸寿 著
SOA大全	Dirk Krafzig, Karl Banke, Dirk Slama 著 /山下眞澄 訳
SOA入門 <a href="http://www-06.ibm.com/jp/software/websphere/developer/soa/">http://www-06.ibm.com/jp/software/websphere/developer/soa/</a>	日本アイ・ビー・エム株式会社 牧野あすか
Webサービスとは・SOAって? <a href="http://www.microsoft.com/japan/msdn/architecture/way/03/">http://www.microsoft.com/japan/msdn/architecture/way/03/</a>	マイクロソフト株式会社 成本正史