

# SOAにおけるシステム方式設計 の進め方

## —SOAを実現していくために 必要なこととは—

### アブストラクト

#### 1. SOAは魔法の玉手箱なの？どのように作ればいいのか本当に知っている？

企業を取りまく環境は、競争力の強化、企業合併や分社化、法規制といった事象が随時発生するようになってきている。そのため、情報システムは常に迅速な変化を求められ、その範囲は複雑、多岐に渡っている。また情報システムの現状は、度重なる変更などにより、複雑・肥大化してきており、多くのバックログを抱えている。そのため、このようなビジネス変化に迅速に対応していくのが困難であり、想定外の期間とコストを費やしている。このような中、「変化に柔軟なシステム」、「コストのかからない方法」、「スパゲッティ状態から脱却」という観点からSOA (Service Oriented Architecture) によるシステム構築アプローチが注目を浴びている。しかしながら、SOAで構築された基幹システムの実現例は少なく、いざ開発しようとしたら、手順書もドキュメントも少ない。さらに、本当に効果がでるのか、ツールはそろっているのか、簡単なのか、生産性はどうか、本当に実プロジェクトに適用できるのか、など実装に至るまでの過程が明確にされていないのが実態である。

#### 2. 無いのなら・作って見せよう・体系図

では一体、SOAを用いたシステム開発はどのように進めるべきなのか。当分科会では、下記の3点をポイントに、SOAで必要となる作業項目の洗出しを行った。

- ・従来のシステム構築プロセスとSOAシステムの構築プロセスとの違いを明らかにする
- ・SOAシステムの構築において、検討が必要な事項（方式）を明らかにする
- ・SOAシステムの構築において、各工程・作業で作成が必要な標準ドキュメントを作成し、作業の完成イメージ（アウトプット）を明確にする

また、多様なSEが共通認識できるよう、作業プロセス、タイミング、言葉の定義を合わすべく、富士通のSDEM21を利用し、独自に「SOA作業体系図」を作成した。この作業により、SOAにおける具体的な全工程別カテゴリ別作業の「可視化」を実現できた。(図表1参照)

図表1 作業体系図

工程	システム方式設計	ユーザインタフェース	システム構造設計	プログラム構造設計	プログラミング	プログラムテスト	結合テスト	システムテスト	運用テスト
カテゴリ	(SA)								
A. 業務		B 業務システム仕様	B1 システム機能	B1SA システム機能要件の確認と機能概要定義	B1UI システム機能の仕様化				
B. 業務システム仕様				B1SA1 既存システムのサービス単位の見直し	B1UI1 スページの定義				
C. アプリケーション				B1SA2 システムサービスの粒度の明確化 ・サービス部品コンポーネントの粒度 各サービスを利用する業務の明確化	B1UI2 サービスバス機能活用の明確化 ・ユーザ機能、メディア活用の検討				
D. インフラ				B1SA3 SOAシステム構築方法の明確化 ・サービスバスの適用有無 ・ワークフローツール適用有無	B1UI3 各サービス機能の定義 ・サービス内の画面、プロセス、データの配置				
E. 運用・移行				B1SA4 サービスごとの基本利用パターンと選択基準の明確化 サービス単位のSLAを決定					
F. 開発支援				B1SA5 サービス構造の明確化(サービス構造図(詳細))					
G. PU管理									

「SOA作業体系図」を作成したことにより、SOAに特化した作業は上流工程に集中し、カテゴリは業務から運用までの広範囲に分散していることが確認できた。これは、SOA成功の背景には内部統制の実施・実現が必要不可欠であることを示唆して

いる一方、SOAで用いている開発技術は実は、既存技術の寄せ集めにすぎないことに起因している。さらにSOAはビジネス思考で業務寄りであると思われがちだが、実はインフラ、運用、プロジェクト管理に至るまで考慮しなければならない点があることが明らかになった。

### 3. 実際に・作ってみよう・SOA

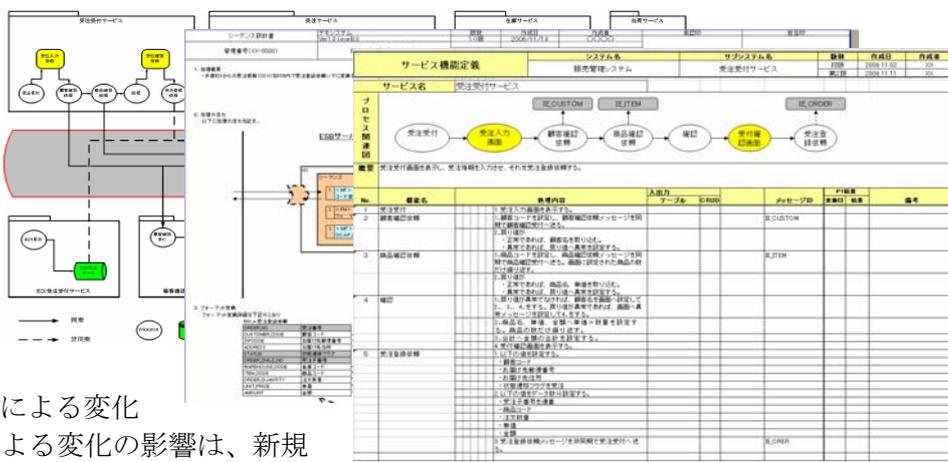
前述の「SOA作業体系図」により一応の作業の可視化を実現したが、さらに踏み込んで、昨年度分科会の受注業務モデルをベースに実装作業を行ってみた。この実装作業によりSOA開発に必要な作業とその具体的なアウトプットを明確にすることができた。(図表2および3参照)

図表2 作業ポイント・アウトプット一覧

工程名称	作業ポイント	アウトプット
システム方式設計	・サービスの抽出、・実装サービスの決定 ・サービスに対するデータ配置の検討と決定、	システム化事務フロー、ER図、ユースケース図、 サービス構造図、インターフェース一覧
ユーザインターフェース設計	・サービス機能定義、 ・サービスインターフェース設計	サービス機能定義書、画面遷移図、画面設計書、 テーブル定義書、インターフェース定義書
システム構造設計	・サービスバス、・シーケンスの詳細定義	アプリケーション構造図、シーケンス定義書

今回はサービスバスに富士通の Interstage Service Integrator (ISI) を使用し、実装作業を行った。またモデルシステムに変化を発生させることで、「変化に強い」というSOAコンセプトを検証した。結果は下記のとおりである。

図表3 SOA開発ドキュメントイメージ



- (1) 新規サービスの追加による変化  
新規サービス追加による変化の影響は、新規サービス内に閉じることができ、開発箇所を局所化できる。またサービスバスにエンドポイントを追加するだけで、既存サービスをそのまま再利用できるため、迅速かつ安全に対応することができる。
- (2) サービス内の業務手順の変更による変化  
サービス提供側と利用側間で各々に発生する業務変更の影響は、サービス間が同期型通信である場合、片方の業務変更の影響が相互に波及してしまうが、サービス間が非同期型通信で接続されている場合は相互に影響することはないため、柔軟かつ安全に対応することができる。
- (3) 異種プラットフォームサービスからのサービス接続追加による変化  
サービスバスの「文字コード変換機能」を使用して吸収することができるため、各サービスでのアプリケーションの追加や修正を行う必要がなくなる。
- (4) フォーマット仕様が異なるサービスとの接続追加による変化  
サービスバスの「フォーマットマッピング機能」を使用して吸収することができるため、各サービスでのアプリケーションの追加や修正を行う必要がなくなる。

これらによりSOAは、単に「変化に強い」ではなく、同期型よりも非同期型の方がより変化に強いなど、より具体的に評価できた。また、SOAに対する期待は実現可能だということを証明できた。

### 4. おわりに

SOAでのシステム開発は、激変の時代に企業が勝ち残るための1つの重要なキーワードである。

当分科会の成果物である「SOA作業体系図」は、開発から運用・保守に至るまでの全ての具体作業が一望できることで、「どのタイミング」で「どんな作業をする」かが瞬時にわかる。さらに実装検証を行ったことで有用性を証明しているが故、即時に実際のプロジェクトで適用できることを約束する。

ユーザ企業やベンダー企業を問わず、多くの開発者にこの「SOA作業体系図」を活用いただき、エンドユーザと共にWin-Winの関係になれることを期待する。