

# 基幹業務のSOA化 ～WebSphere ESBで始めるSOA～

2006年7月12日

ゼネラル・ビジネス・サービス株式会社  
サービス統括 第3システム・サービス部  
恩田 将

# Table of Contents

1. SOAとは？
2. SOAとWebサービス
3. SOAとWeb2.0
4. ESBとは？
5. 基幹業務をSOA化する方法
6. WebSphere ESB紹介
7. まとめ

# 1. SOAとは？

# SOA?

Service Oriented Architecture ??

サービス指向アーキテクチャー ???

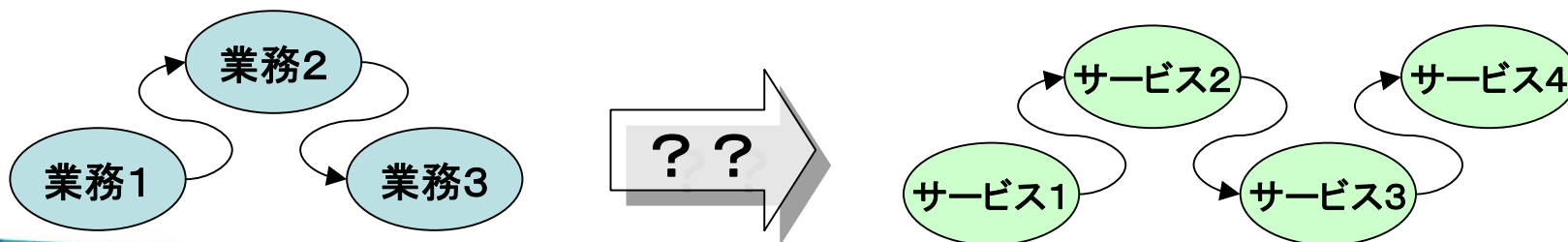
# 1. SOAとは？

「大規模なコンピュータ・システムを構築する際の**概念あるいは手法**の一つであり、業務上の一処理に相当するソフトウェアの機能を**サービス**と見立て、そのサービスをネットワーク上で連携させてシステムの全体を構築していくこと」

ーウィキペディア「サービス指向アーキテクチャー」より

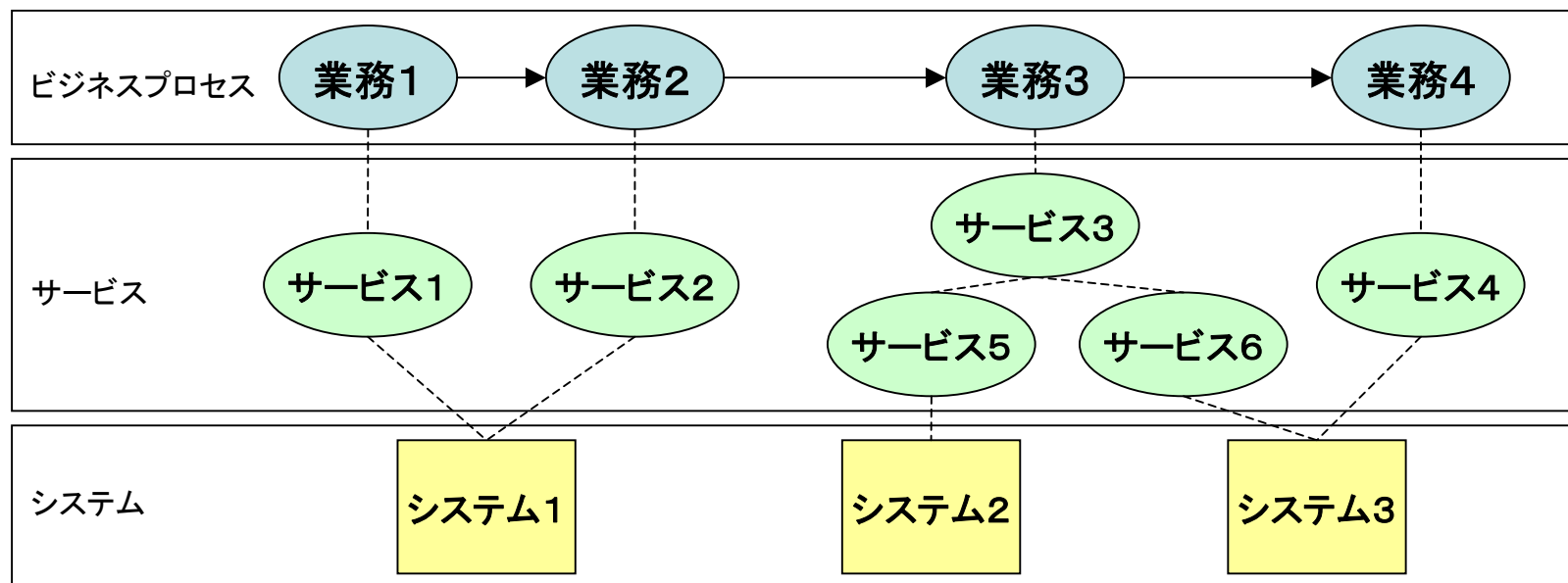
<http://ja.wikipedia.org/wiki/SOA>

- SOAはシステム構築の概念または手法であって、ある特定の技術や製品を指す言葉ではない
  - 「この製品を導入すればSOAが出来ます！！」はホント？
- 業務をサービスと見立て連携させる
  - サービスの切り出し方(粒度)に関する議論は様々。。
  - サービスの定義は業界、企業によってそれぞれ



# 1. SOAとは？

- SOAはレイヤーアーキテクチャー



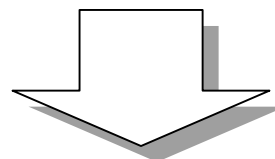
- レイヤーにすることで疎結合を実現
- 何よりも重要なのはビジネスプロセス
  - ビジネスモデル、ビジネスプロセスが明確でないのにSOAを実現しても・・・???

# 1. SOAとは？

## • SOA誕生の背景

### 【システム開発の課題】

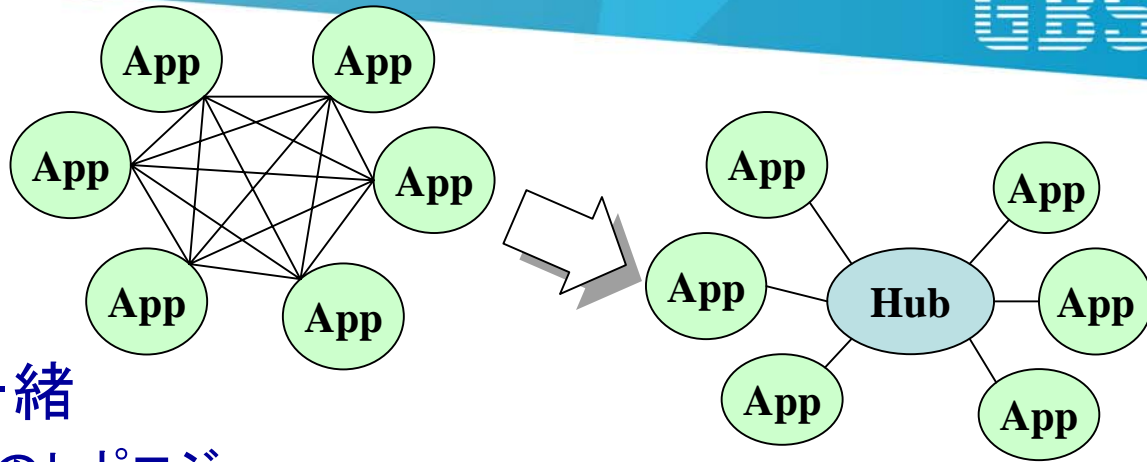
- ・ますますスピードアップするビジネス
- ・それを実現するためにITへの投資は必須
- ・しかし限りあるお金と時間
- ・仮にお金と時間があっても作っている間にビジネスが変化してしまう。。。
- ・せっかく作ってきた既存のシステム資産も出来るだけ再利用したい。。。



**既存資産を生かしつつ、いかに迅速に変化に対応するか？**

- SOAはその実現手段の一つ
- しかし、銀の弾丸ではない！
- SOAの考え方自体、新しいものではない

# 1. SOAとは？



## • SOAとEAIの違い

- 基本的な概念は一緒
  - ハブ&スポーク型のトポロジー
- EAIはそれぞれのアプリが使用している固有のプロトコルで接続し、メッセージ変換等を行う
  - 内部処理はベンダー固有かつ密結合
- SOAはオープンで標準化されているインターフェースでやり取りを行う
  - 代表的なものがWebサービス
  - 内部処理はSCA、SDOなど標準技術を使用
  - 疎結合

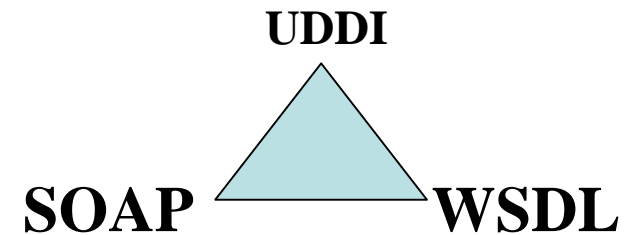
## 2. SOAとWebサービス

- SOA ≠ Webサービス

- SOAは概念、Webサービスは技術
- しかし、WebサービスはSOAを実現するための最有力手段
  - Webサービスを使用しているからSOAではない
  - Webサービス以外の方法でもSOAは実現できる

- なぜWebサービスが重要なのか

- 製品、言語を超えた標準仕様
- 通信プロトコルを問わないため、HTTPを使ってFireWallを超えて企業間連携も可能
- WSDLがあればサービスの呼び出し処理は自動構成可能
- 周辺技術の発展により実運用が現実化 (WS-\*)



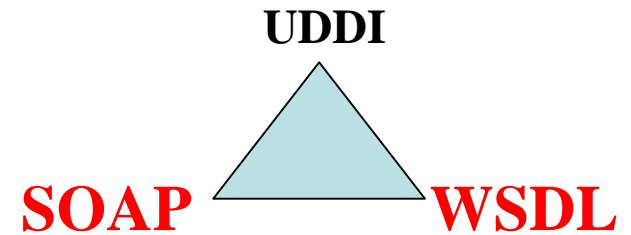
## 2. SOAとWebサービス

- SOAP

- サービス間の呼び出し、応答のプロトコル
- XMLで定義されており、ヘッダー・本文といった手紙のテンプレートのようなもの
- 実際の通信プロトコルに制限はなく、HTTP、SMTP、JMSなど様々なプロトコルに乗せることができる

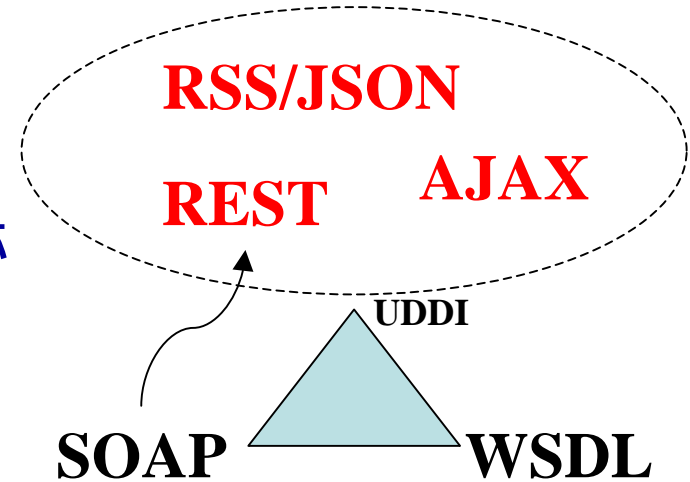
- WSDL(Web Services Description Language)

- XMLで定義されたWebサービス記述言語
- Webサービスがどのサーバーのどのポートにあり、どの操作に何を渡して何が返ってくるのかが全て記述されている
- WSDLを元にWebサービスを呼び出すために必要なモジュールを全て自動生成できる



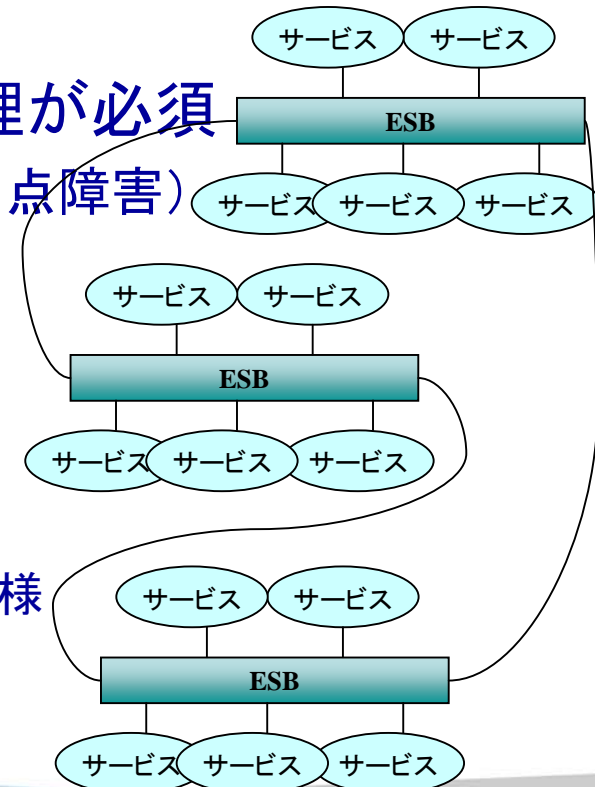
### 3. SOAとWeb2.0

- Web2.0とは
  - 次世代Webにまつわる概念の総称
  - BuzzWord??
  - 主にB2Cの世界での現象
- マッシュアップ (MushUp)
  - 既存のサービス同士を繋げて新しいサービスを作り出す
  - 例: GoogleMapsを使ったサービス
  - 誰でも使えるAPIを提供
  - ん?



## 4. ESBとは？

- ESB (Enterprise Service Bus)
  - SOA実現のための実装基盤
  - 標準インターフェース (Webサービスなど) を使用してサービスをつなぐ論理的なバス
    - ESB同士の連携も可能
  - ESBは分散処理、フェイルオーバー処理が必須
    - ESBが止まれば全てが止まってしまう (単一点障害)
  - ESBはサービス間のMediator (仲介役)
  - ESBの標準はまだ存在しない
    - SCA (Service Component Architecture)
      - IBM、BEA、Oracleなどが共同で発表
      - 言語に依存しないサービスを作るための仕様
    - JBI (Java Business Integration)
      - JavaによるSOAのための仕様



## 4. ESBとは？

- ESBを導入するわけ

- パフォーマンス的にどう考えても遅くなりそうなのに何故？
- 1対1でそのままつなげた方が早いんじゃない？

→サービスとの間に一枚挟むことで、サービス間の直接的な依存関係を断つ(疎結合)

→サービスの共有化、再利用性の向上を図る

## 5. 基幹業務をSOA化する方法

- なんだかんだ言ってWebサービス化しなきゃダメ？
  - WebサービスだけがSOAの手段ではない
  - とはいえ、Webサービス化することがSOAへの近道であることは確か
  - って言ったって、どうやって。。。。
- System iならば簡単にプログラムをWebサービス化できます
  - WDSC(WebSphere Development Studio for Client)のWebサービス作成ウィザードを使用
  - 裏ではAS/400 Toolbox for JavaのPCMLを使って、JavaからプログラムをCallしている

## 5. 基幹業務をSOA化する方法

# DEMO

## 6. WebSphere ESBの紹介

- IBM WebSphere Enterprise Service Bus v6.0.1
  - IBMのESB実装でありSOA基盤製品
  - 標準インターフェース(SOAP/HTTP, SOAP/JMS)による接続をサポート
    - WebSphere Adaptersを使用してデータベース、フラットファイルでの接続も可能
  - SCA、SDO等の標準技術を実装し、サービスの共有化、再利用性の向上を実現
  - 開発はWebSphere Integration Developer(WID)で行い、ノンプログラミングで定義が可能
  - WAS NDの上位製品であるため、クラスタリング、フェイルオーバー、セキュリティ等の技術はそのまま使用できる

## 6. WebSphere ESBの紹介

- WebSphere ESBのメディエーション(仲介)機能

- XSL変換

- サービス間のデータ構造をマッピング、変換処理

- データベース・ルックアップ

- メッセージ・フィルター

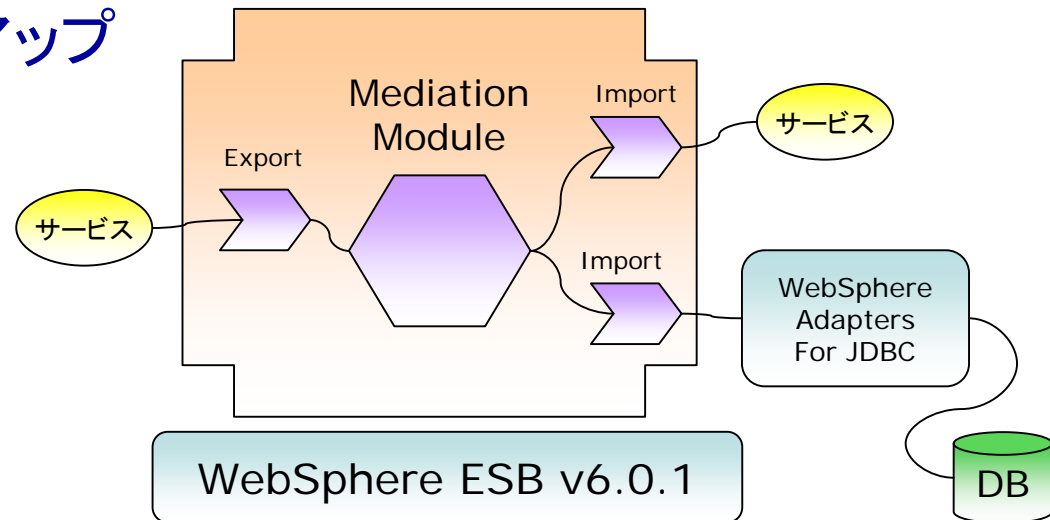
- 条件分岐

- メッセージ・ロガー

- その他

- ビジネスプロセスは

WebSphere Process Serverで定義



## 6. WebSphere ESBの紹介

# DEMO

## 7. まとめ

- SOAは変化に迅速に対応するためのアーキテクチャ
- SOAとWebサービスは別物だが、WebサービスはSOA実現のための最有力手段
- 基幹業務といえどもWebサービスに対応することがSOA化への一番の近道
- System iであればWDSCを使ってプログラムを簡単にWebサービス化可能
- WebSphere ESBはSOAを実現するための基盤
- WebSphere Adaptersを使用すればWebサービス以外のサービスも接続が可能

ありがとうございました