XML Consortium

サービス設計のベストプラクティス

2007年5月15日

SOA部会

日本アイ･ビー･エム㈱　日力俊彦

本日の内容

- サービス設計ベストプラクティス WG活動報告
  - 当年度における当WGでの活動成果について、ご紹介します。

SOAに対する理解

- SOAの考え方(コンセプト)は浸透してきた(かな・・・)
  - 現実的な構築手法は未だ確立されていない
- サービス指向アーキテクチャーであるからには、"サービス"が重要(なんでしょ?)
  - どのようにして定義するか ?
  - どのようにして実装するか ?
- SOAではビジネスモデリングも重要(なんだよね?)
  - ビジネス上の目標/課題に対するトレーサビリティー

SOAをめぐる声・・・

SOAを巡り、現場で良く聞かれる声は・・・

- 皆様の声をお聞かせ下さい。。。

やっぱり・・・

- “サービス”って、よくわからない・・・
- “サービス”って、難しい・・・

・・・ですか？

サービス設計のベストプラクティス

- 当WGの活動目標
  - 目的：SOAにおけるサービスの捉え方及びその設計に関するベストプラクティス検討及び提示
    - 現時点において、SOAにおける“サービス”の捉え方に画一的な標準手法は存在しません。
      参加者のこれまでの経験/知識を踏まえて、
      “サービスとは”を再確認すると共に、
      その設計におけるポイントを検討します。

メンバー一覧

- メンバー（五十音順、敬称略）
  - 天野　富夫　日本アイ・ビー・エム(株)
  - 河原　正博　沖電気工業(株)
  - 竹内　拓也　(株)日立製作所
  - 中村　知義　(株)ジャステック
  - 何翁　径迪　(株) アイ・ティ・フロンティア
  - 日力　俊彦　日本アイ・ビー・エム(株)

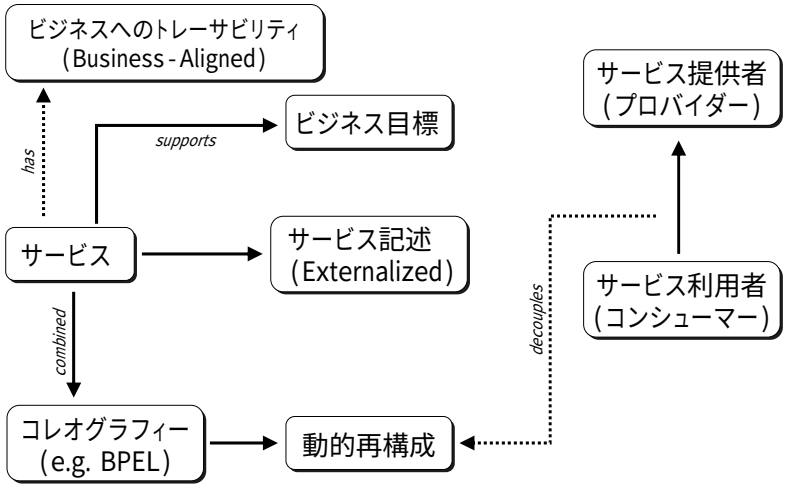
よく言われるSOAの解釈

- 基本は・・・
  - アーキテクチャーの原則である
  - 特定の製品やツールを指すものではない
- 特性は・・・
  - 疎結合の**サービス**の集まり
  - サービス間は標準化された**インターフェース**経由で通信
    - 複雑さ（実装）の隠蔽
  - サービスあるいはその組み合わされたものは**再利用可能**（なビルディング・ブロック）
    - LEGOブロックのイメージ
  - **Webサービス**がベースのキーテクノロジー

サービスとは

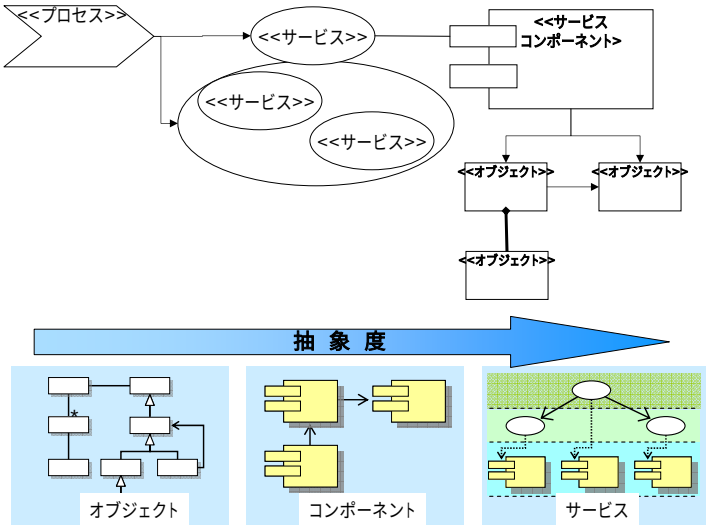
- サービスはSOAが定義する主要な要素の一つ
  - サービス提供者（プロバイダー）
  - サービス利用者（リクエスター）
  - サービスそのもの
- 業務の処理単位を論理的に記述したもの
- Well-Known, Well-Definedなインタフェースにより公開され、発見可能(discoverable)である
  - 必ずしもWebサービスである必要はない
- 再利用の単位
  - サービスのExtension Pointでもある

サービスとは(続き)



出典：http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/

サービスとは(続き)



SOAベースの開発プロセス例

- 従来の開発プロセス
- SOAベースになると・・・何か変わる？

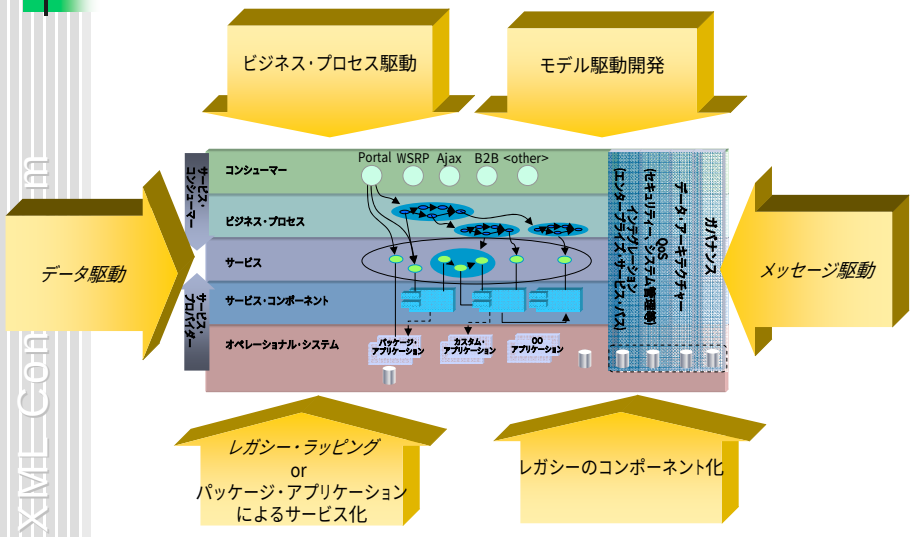
契約に基づく設計 (DbC)

DbC : Design by Contract

- ダイクストラが基本的な考え方を提唱
  - “A Discipline of Programming” by Edsger. W. Dijkstra (1976年)
- ソフトウェア・モジュールが満たすべき制約 (契約) を明確に定義
  - 事前条件
    - メソッドが呼び出される直前で、メソッドが仮定している条件
  - 事後条件
    - メソッド呼び出し終了直後に成立しているべき条件
  - 不変条件

DbCは、インターフェース規定のお作法を示している

サービス指向デザインのアプローチ



出典 : http://www-128.ibm.com/developerworks/webservices/library/ws-soa-soi/#N100B9

現実的なアプローチ（1/2）

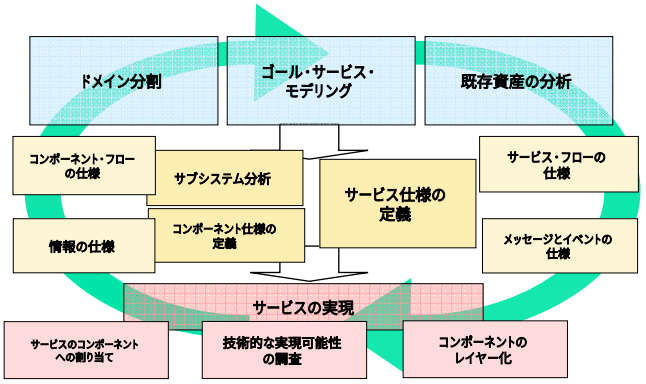
- トップダウンもしくはボトムアップによるアプローチ
- 機能を分析することにより、サービスの候補が識別される
- 非機能要求（NFR）をどのように実現するかによりサービスの粒度を検討する
- DbCによるインターフェースの分離
- BPELを使用したプロセスでは、フロー制御をサービスから分離する

現実的なアプローチ（2/2）

- トップダウン・アプローチ
  1. 業務分析/要求管理を行う
  2. ビジネス・プロセス・モデリングを行う
  3. 発見/定義されたサービスを実装
  4. フロー（BPEL等）によってサービスの結合を行う
- ボトムアップ・アプローチ
  1. 既存の業務分析から行う
  2. サービス候補を発見/抽出する
  3. 抽出された候補をサービスとしてラップする
  4. フロー（BPEL等）によってサービスの結合を行う
- ミート・イン・ザ・ミドル　（折衷案）
  1. トップダウン・アプローチ 及び ボトムアップ・アプローチによってサービスを発見/定義
  2. サービスを実装、ラップする
  3. フロー（BPEL等）によってサービスの結合を行う

サービス設計アプローチ例

- ビジネス分析により導出されたビジネスコンポーネントをもとに、SOA実装へ繋げる為のメソドロジー例



出典：http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/

サービス抽出の手順例

1. ビジネス・コンテキストの作成
2. システム・コンテキストの作成
3. ビジネス・プロセス図の作成
   - As-Is
   - To-Be
4. システム・ユースケースの作成
5. サービス候補の抽出
   - ビジネス・プロセス図におけるシステム境界に着目
   - シーケンス図等を活用して、候補の妥当性をチェック
6. サービス粒度の評価
   - サービス間の依存関係
   - 非機能要件
7. サービス候補の評価
   - アンチパターン
   - 非機能要件
8. サービス仕様の策定
   - サブシステム分析
   - コンポーネント仕様定義
   - メッセージ仕様等の定義

サービス候補抽出時のアンチ･パターン

- パターン#1：サービス増殖パターン
  ビジネスの視点で整理されていないFine Grained（細粒度）なサービスを公開することでサービスの増加のみを招く.
  - できるだけCoarse Grained（粗粒度）のサービスを公開する。
    Fine Grained（細粒度）なサービスもあり得るが、これはビジネス上のNeedsを検討した上で採用されるべき。
- パターン#2：サイロ･パターン
  サービスが特定のアプリケーションの為の局所的なものとなっているパターン.（他での再利用が考慮されていない）
  - ミート･イン･ザ･ミドル アプローチでのサービス抽出を考える。

サービス候補の評価例

- 基本的な考え方
  - サービスとして公開する／しないは、再利用可能性を前提（合目的性／再利用可能性）
    - サービス候補として抽出されたものを全て公開する必要はない
  - 全体を管理可能であること（≠サービス増殖パターン）

サービス候補の評価基準例

このサービスは･･･

- 合目的性･･･ ビジネス上の目的やゴールの指標にあっているか？
- ステートレス性･･･ リクエスト間で状態の維持、情報を必要としない？
- 外部仕様（発見可能性）･･･ 明確なインターフェース、サービスの記述を持っているか？
- 再利用可能性 ･･･ それを必要とする他のプロセスの要求を満たしているか？
  上位のレベルの他のプロセスで再利用可能？

サービス評価から仕様策定へ

- **サービス仕様策定**
  - サブシステム分析/コンポーネント仕様定義
    - 静的分析と動的分析
    - クラス図、シーケンス図等を活用
  - インターフェースの定義
    - メッセージ／パラメーター
  - WSDLの定義
    - メッセージ
    - ポート・タイプ (Operation)
    - バインディング
    - サービス
- **サービス実装へ繋げる**
  - サービス・レイヤリング
    - 非機能要件を考慮

まとめ

- SOAにおけるサービスをどのように捉え、設計していくかはまだこれからの領域
  - 次年度活動で、より洗練されていきます
- 出来るだけ多くの方々との経験と知識の共有が必要
- ご興味のある方 (あるいは持った方) は是非、当部会へ飛び込んで来てください
  (参加者募集中！)