

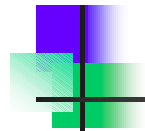
SOA概要説明

第 1 版

XMLコンソーシアム

SOA部会、B研究部会

SOA普及 ProjectTeam



《 目 次 》

Service-OrientedArchitecture	-----	1
1. 現在の IT 環境	-----	2
2. SOA とは	-----	4
3. SOA におけるサービスとは	-----	6
4. SOA の位置付け	-----	9
5. SOA のビジネス上での意義	-----	12
6. サービスのオーケストレーション	-----	14
7. ワークフロー	-----	16
8. SOA 実現化	-----	19
9. まとめ	-----	24
参考資料	-----	25
用語説明	-----	30

Service-Oriented Architecture

21 世紀を迎え、IT を含む様々な技術革新が引金となり、市場の多様化、変化のスピード化が急速に進展している。そのため、企業活動において、速やかな変化への対応と、同時に ROI を高めることが経営側に強く求められる時代になっている。

IT におけるこの 5 年間は、XML、Web サービスに関連する技術革新が顕著であり、特に、

- ①XML、Web サービス、BPM、SOA 等の技術の確立
- ②W3C、OASIS など標準化団体により進行中の意欲的な技術の標準化作業
- ③WS-I による Web サービスの相互運用性の確立
- ④UML、BPMN、BPEL と言ったビジネスプロセスを表現するモデリング記法や記述言語が確立
があげられる。

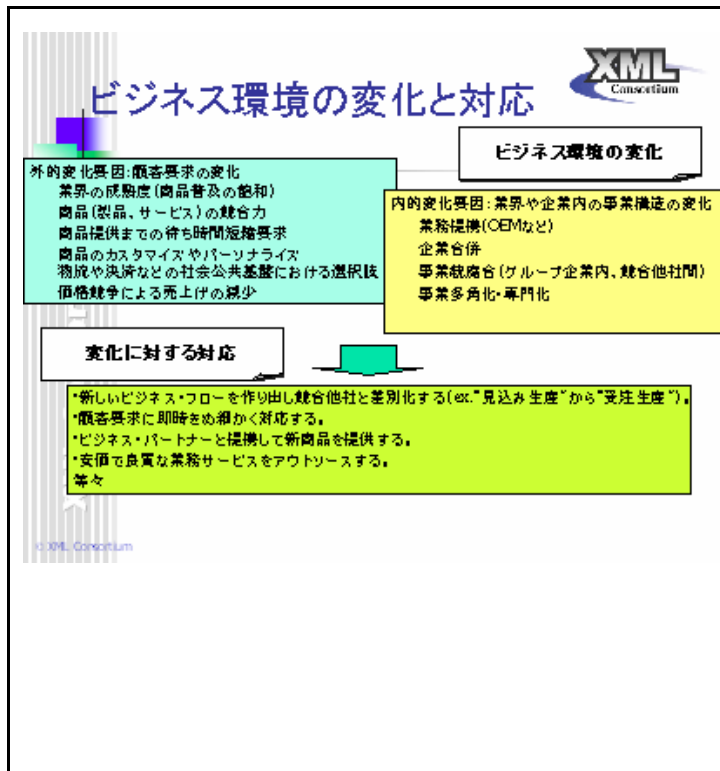
これによりに業界、企業、組織に分散するシステムを協調動作させることが容易になり、また、ビジネスプロセスの変革や改善とシステムの再編の効率が格段に上がることになる。

ビジネスイノベーションを見据えた経営のニーズに応え、必要とされるシステムを短期間で低いコストで備える必要がある。その IT 基盤の中核には、分散する各システムの機能をサービスとして部品化し、サービスを組み上げることでシステム間のプロセスを形成したり、既存システムを利用して新規システムを構築する SOA の概念がある。本資料は、SOA を中心に、その考え方、効果、実現の方法の概観を、今までに XML コンソーシアム部会活動で発表を行ったスライドを基に解説する。

説明は以下の順序で行う。

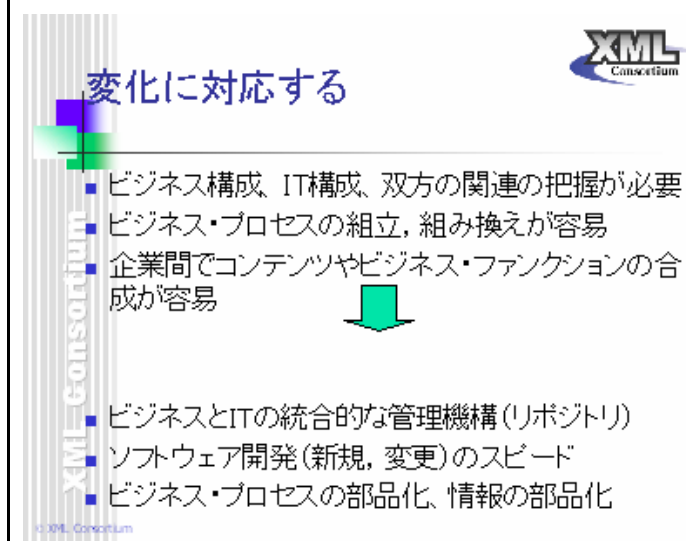
1. 現在の IT 環境
2. SOA とは
3. SOA におけるサービスとは
4. SOA の位置付け
5. SOA のビジネス上での意義
6. サービスのオーケストレーション
7. ワークフロー
8. SOA 実現化
9. まとめ

1. 現在の IT 環境



まず、IT を適用するビジネス環境について、考えてみる。

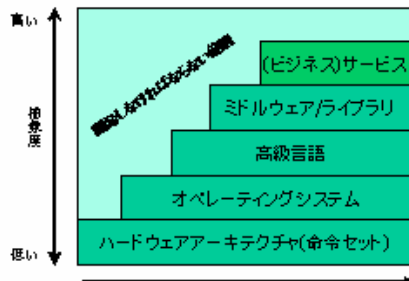
ビジネス環境の変化を大きく二つに分類する。それは、①外的要因による変化と②内的要因による変化である。外的要因による変化は市場の変化で、競争の激化、時間短縮要求、商品のパーソナライズ、価格競争などあり、内的要因には、企業構成の変更や、事業の多角化といった企業の構造の変化である。これらのビジネス環境の変化に対応し、企業として利益を得るためには、新しいビジネス・フローを作成し、顧客の要求に細かく対応する等の他社との差別化を行う必要がある。また、パートナーと提携しての新商品提供や、アウトソースにより良質で安価な製品提供するといった対応も必要となってくる。



そのため「ビジネスの環境変化に対応して、俊敏にシステムを変更する」ことが強く求められている。では、どのように実現するのであろうか？ そのためには、まず、1. ビジネスとシステムの構成と関連を把握する必要がある。2. 変化に対応するためには、構成されたビジネスのプロセスの組み替えに容易に対応できる必要がある。更に、3. 提携や委託といった方式でビジネス環境の変化に対応するためには、企業間においても、ビジネスの機能の合成も容易に行える必要がある。システム環境としては、ビジネスプロセスの観点からソフトウェアを部品化し、意味ある情報の単位に細分化されたデータを集め部品化する必要がある。また、ビジネスとシステムの統合的な管理の仕組みが必要となる。変化に俊敏に対応するために、当然ながら更なるソフトウェア開発スピードの向上が必要になる。統合的な管理機構を用い現状を把握し修正案を検討し、素早くソフトウェア部品とデータ部品を組み替えすぐさま運用が開始できるようになることが理想である。これらを実現するシステム環境のアーキテクチャとし

ソフトウェア開発のトレンド

■複雑さに迅速に対処する方法はより抽象度の高いレベルで開発を行うことである



て SOA が注目を浴びている。

これまでのソフトウェア開発は、どのように発展してきているのだろうか？かつては、ハードウェア・アーキテクチャの命令セットを意識してアセンブリ言語でアプリケーションを開発していた。その後、オペレーティングシステム、高級言語、ミドルウェア/ライブラリなどソフトウェアの実行環境と開発環境が用意され、アプリケーション開発は、より抽象度の高い開発に移行している。この進化が進むにつれて、アプリケーション開発で記述するコードの量が減り開発規模が小さくなり開発スピードが高まっている。また、抽象度が高くなると同時に、プラットフォームの依存性が低くなってきている。ライブラリといった特定の言語に依存する段階から、現在は、言語や実行場所に依存しないサービスが、更に上位に位置付けられている。この段階になると、粒度の違いはあるものの、ビジネスの観点で扱えるレベルにまで、ソフトウェア部品の抽象度をあげることが可能となる。つまり、自動化するビジネスプロセスの構成要素となりうるレベルにまで抽象化が進んでいると言える。

2. SOA とは

SOAとは

- 散在するシステムを、業務視点の機能(サービス)の集合と捉え、ビジネス環境の変化に対して、迅速にサービスを組み換える (Business Process Integration) ことで、柔軟に対応するシステム構築方針
 - アプリケーションをラップして **サービス**にする
 - 外からはサービスの **インターフェース**のみを意識する
 - ハードウェア/OS/ミドルウェア
 - 開発言語 等々を隠蔽
 - サービス自体の実装には依らない
 - **インフォメーションパス(ネットワーク)**にサービスをプラグインして統合を実現する
- 具体的には上記を実現するための
 - アーキテクチャパターン
 - デザインパターン
 - ベストプラクティス
 - 等々の集まりである

現在のビジネス環境及び技術の変化という観点から、背景を見てきたが、主題である SOA とは何かという具体的な内容について考察を行ってみる。現在のシステム環境では、システムが散在しているのが現実である。散在するシステムを業務の視点から機能の集合にとらえ、環境の変化に対応して迅速に機能を組み替えることで、柔軟に対応するシステム構築方針が SOA である。ここでいう機能は、それぞれが業務へのサービスであることから、ビジネスのサービスと言いかえることができる。システム的な観点より具体的な実装イメージで表現すると、アプリケーションをラップし、サービスの提供者として実現する。そして、外部から見た場合には、実装方法やハードウェア、OS、ミドルウェアに関係なくインターフェースのみを意識すればいいようにする。これらにより、サービスとして、1つの業務部品として扱うことが可能となる。

こういったシステム構築の実現を行うためには、アーキテクチャパターンやデザインパターン、ベストプラクティス等が必要であり、これらが SOA の構成要素である。

今までと何が違うのか

- SOAという概念は以前からある
 - 概念としてはシンプルで説得力がある
 - しかし、期待したほどの成果はあがっていない
- Corba, DCOM, EJB に足りなかったもの
 - 共通のインターフェース定義言語
 - 共通のインフォメーションパス
 - 柔軟なインターフェース記述
 - インターフェースとプログラミングモデルの分離

SOA の考え方は、Web サービス技術が登場する以前から、アプリケーションを効率的に開発、運用する分散プログラミングの概念として存在していた。また、CORBA, DCOM, EJB など様々な技術が出現したもの、広く普遍的に使われるまでに至っていない。DCOM や EJB はそれぞれ Windows や Java など特定のプラットフォーム固有の技術であり、CORBA は十分な機能を持ち合わせる半面複雑であるために堅牢性を必要とする特定領域における利用に留まっている。しかし、それ以上に不足していたことは、開発者が慣れ親しんだ開発言語やプログラムスタイルを維持しつつ他種アプリケーションとの通信を可能とする協調的な思想であった。後発の Web サービス技術では異種アプリケー

	ション間の協調性、既存アプリケーションの有効利用を意識している。そのためインタフェースとプログラミングモデルの分離を原則にしている。Web サービスによる分散プログラミングの特徴は次の三つである。XML を用いたインタフェース定義言語の共通化。異種アプリケーションが参加するバス型結合を意識したXML によるメッセージング基盤の確立。同一種アプリケーションが存在する局所的な環境に配慮した柔軟なインタフェース記述。の3つが特徴といえる。
 ビジネスイノベーションを支える技術 Web Services  SOA BPM © XML Consortium - Webサービス - 相互運用性の実現 - ビジネス・プロセス管理 (BPM) - ビジネスの定義 - モデリング方法論 - プロセスの管理・維持のためのツールの集合 - サービス指向アーキテクチャ (SOA) - システム構築方針 - アーキテクチャパターン	SOA は複数の異種システムを協調動作させる言わばシステムのシステムを構築するアーキテクチャ「概念」である。ビジネス環境の変化にシステム環境が迅速に対応するためには、SOA と極めて親和性に優れたインターフェイス技術「道具」である Web サービス、ビジネスプロセスを構造的にモデル化しシステムとの関連を明らかにするビジネスプロセス管理 (BPM) 手法の組み合わせが現時点のベストプラクティスとして考えられている。例えば、販売会社のビジネス・プロセスには、受注管理、在庫管理、債権管理、売上管理など複数のアプリケーションシステムが存在し、電話での受注、請求書の発送、商品出荷など人手による事務処理も必要な場合でも、BPM では統合的にモデルで表現し、人手の業務、業務を支援するシステム機能、業務を自動化するシステムのプロセスを関連づけることが可能である。SOA のサービスはシステムの機能、自動化するプロセスのアクティビティとしてビジネスプロセス・モデルと対応づけることになる。

3. SOA におけるサービスとは

 SOAにおけるサービスとは ■ SOAで扱うサービスが持つべき性質 - 性質 1 個々のサービスの実装を隠蔽するインターフェースが定義され、機械処理可能な形式で記述されていること  WSDLで記述される広義のWeb サービスであること - 性質 2 ビジネスプロセスとして適切な粒度・抽象度を持つインタラクションやメッセージが定義されていること  インターフェース定義の意味・内容が可わる © XML Consortium	SOA は、サービスの統合で物事を実現することであり、サービスの集合体である。そのサービスの性質は、2つあり、1つ目は、サービスの実装を隠蔽したインターフェイスが定義され、機械処理可能な形式で記述されていることである。2つ目としては、ビジネスプロセスを実装できるレベルの粒度を持っていることが必要である。これらの性質を持ったサービスを組み合わせることでSOA 実現が可能となる。
 Webサービスの価値/本質とは？ ■ アプリケーション部品の呼出し ■ SOAP/HTTPを使ったワイヤーフォーマット/プロトコルレベルでの相互運用性確保 ■ なんらかの境界を超えてのアプリケーションの連携・統合 ■ インターネット/ファイアウォール、異なる分散プログラミング環境、異なるミドルウェア ■ アプリケーション部品の記述・リポジトリ ■ インタフェース記述レベルでの相互運用性の確保 ■ WSDLによるインタフェースの機械処理可能な記述(ワイヤーフォーマットにはこだわらない) ■ 抽象インタフェースと実装プロトコルの分離 ■ 部品を組み合わせでビジネスプロセスを実現する基盤 © XML Consortium	SOA と極めて親和性に優れたインターフェイスの技術である Web サービスの価値/本質とはどのようなものかについてもう一度考察してみる。まず、1 点目として、「アプリケーション部品の呼び出し」がある。ワイヤーフォーマット/プロトコルレベルでの相互運用性の確保を実現し、ファイアウォール越えや異なる分散プログラミング環境、異なるミドルウェア間でのアプリケーション部品呼び出しが可能となる点である。 2 点目として、「アプリケーション部品の記述・リポジトリ」がある。WSDL などの標準インターフェイスにより、インターフェイスと実装プロトコルを分離し、部品の組み合わせでビジネスプロセスを実現できる。 これら2点の特徴によりWeb サービスは、インターネット上でプラットフォームに依存しないサービス間連携を容易に実現することを可能とする。

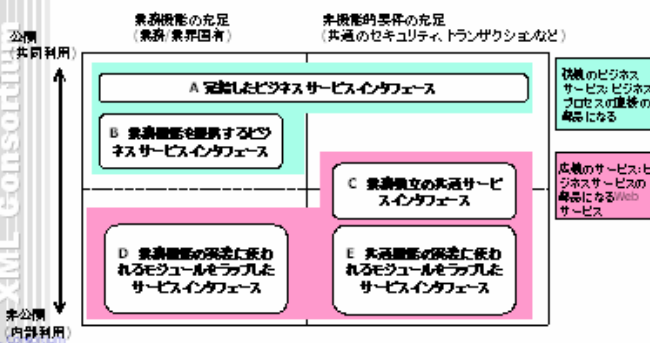
SOAにおけるインタフェース記述の意義

- 全てのアプリケーション部品(サービス)の一元的な扱い
 - 共通のサービス呼び出し機構 (WSIF)
 - SOAPでラップされたアプリケーション(通信プロトコルには依存せず)
 - REST Webサービス(インターネットWebサービス)
 - Javaコード
 - レガシーアプリケーション
 - ワークフローによるWebサービスの合成
 - BPEL4WSの記述はWSDLによるPortType等の定義を参照している
 - UDDI/WSDLによるサービスの検索

SOAにおけるインタフェース記述の意義は、「全てのサービス(アプリケーション部品)を一元的に扱えるようにする」という言葉に集約される。具体的には、EJBやServletなどJavaプログラム、C#などで作成された.NETプログラム、ワークフロー記述言語(BPEL4WSなど)で合成された複数のWebサービスなどである。Webサービス技術の利点は、プラットフォーム、開発言語、通信プロトコル、実装ミドルウェアを意識することなく、様々なアプリケーションのインタフェースをWSDLで記述できることである。

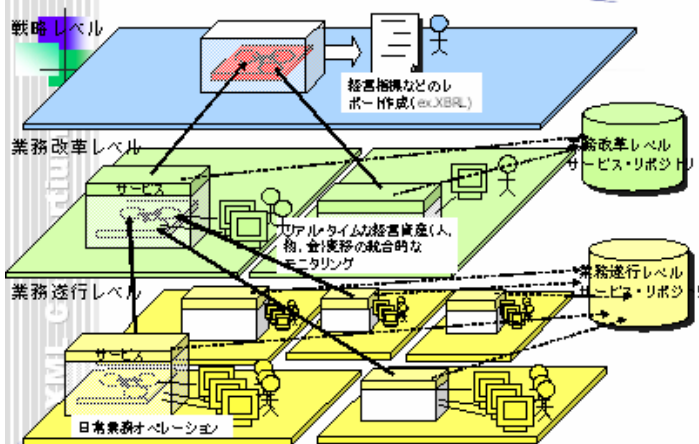
SOAで扱うサービスの分類(例)

- サービスの用途によって必要な粒度・抽象度は異なる



SOAの構成要素であるサービスの粒度はSOAを適用した設計において重要なポイントになる。用途によりサービスの粒度・抽象度は異なる。1つは、ビジネスプロセスの直接の部品となり、業務を遂行するためのインタフェースを定義したビジネスのサービスである。もう1つは、ビジネスサービスの部品となり、インタフェースを定義した業務から独立した共通のサービスやモジュールをラップしたものはビジネスサービスの部品となる。一般的には、Webサービスは、後者のビジネスサービスの部品となる。双方ともSOAで扱うサービスであるため、用途に応じて、粒度の考え方を考慮し、構築する必要がある。

SOAベース企業システム



次に、総合的にSOAベースの企業システムを考えてみる。

SOAベースの1つのシステム構成案として、業務遂行レベル、業務改革レベル、戦略レベルの三階層で構成することを考えよう。各層では、ビジネス機能や関連するビジネスプロセスの再利用性を考慮してシステム機能をサービスとして部品化する。

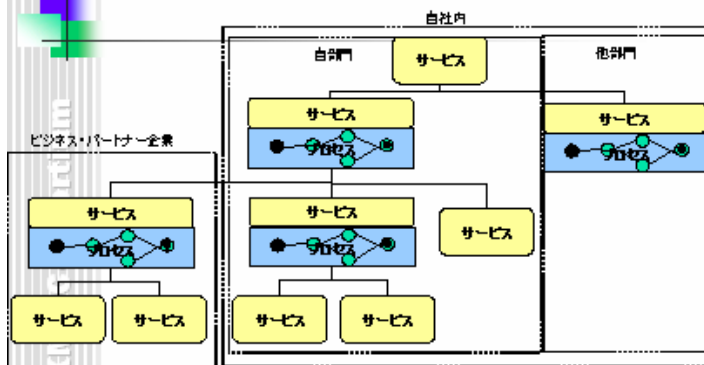
業務遂行レベルでは、日常の業務処理を実施するためのシステムであり、サービスから構成された個々の業務群である。業務改革レベルは、業務遂行レベルのビジネスサービスを組み合わせ構成し、ビジネスプロセスを構成する。そして、ビジネスプロセスの組み換えにより業

務改革を実施する。戦略レベルでは、業務改革レベルのビジネスサービスを利用し、全社レベルでの経営状況を監視/把握を行う。

このような情報システム環境を備える企業では、適宜、迅速に組織間のビジネスプロセスを統合・変更することができ、また、リアルタイムに経営状況を把握することも可能になり、経営に IT を直結することが可能となる。

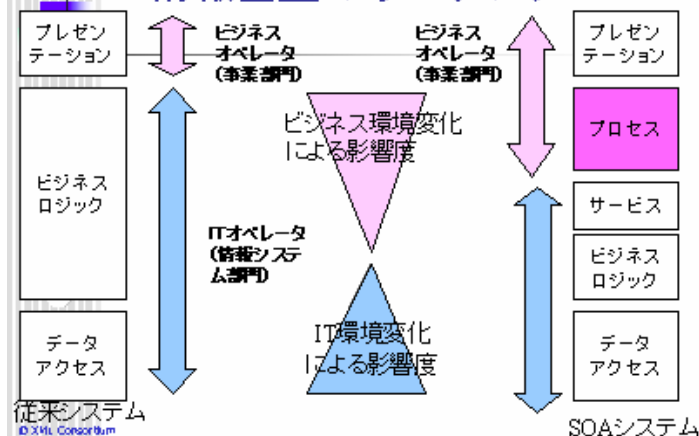
4. SOA の位置付け

ビジネス・プロセスとサービスの関係



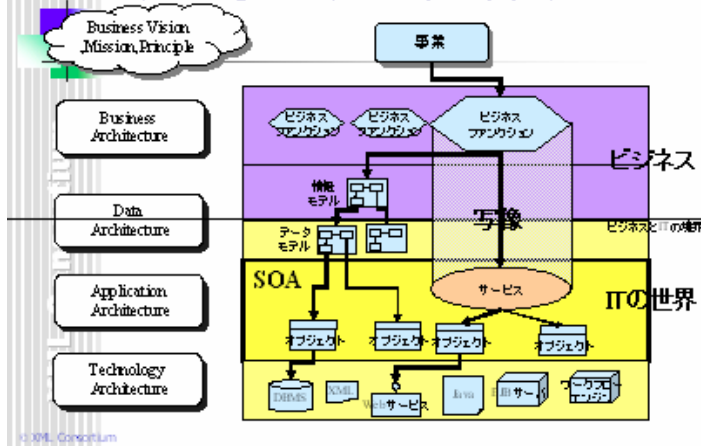
事業を構成するビジネスプロセスは、各組織や役割に分担された業務上のサービスの組合せと考えることができる。このサービスは、組織と同様に階層的に管理される。ビジネスプロセスは、組織階層と同様に垂直に統合されるものと、組織横断的に水平に統合されるものがある。SOA に基づく情報システムでは、組織間に跨るビジネスプロセスに合わせてデータや処理を統合するために、各システムのサービスを統合してプロセスを形成する。このプロセスで統合するサービスは、社内の部門内外、社外を問わない。

SOA情報基盤のオーナーシップ



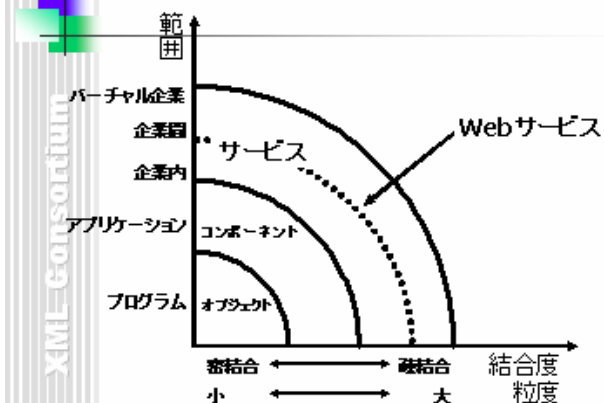
SOA による情報基盤におけるオーナーシップを考えてみる。従来のシステムの構造は、大きく、プレゼンテーション層、ビジネスロジック層、データアクセス層に分割をされている。クライアント PC などオペレーションに関連したプレゼンテーション層以外は、情報システム部門がオーナーシップを持つ。SOA によるシステムでは、従来のビジネスロジック層は、プロセス層、サービス層、ビジネスロジック層に分かれる。プロセス層は、分散する複数のシステムで実装されるサービスを統合する層である。サービス層以下は、従来どおり情報システム部門等が担うが、プロセス層は、サービスを統合するワークフロー管理機能が誰でも扱えるように簡便になれば事業部門に移譲することが可能になる。ビジネス環境の変化を知り戦略を検討する事業部門が、自らビジネスプロセスに合わせシステムを変更できるようになることで、より早くビジネス環境の変化に対応することができる。

ビジネスからITのトレーサビリティ



ビジネスプロセス・モデルに SOA のサービスを関連づけることでビジネス構造（業務体系）とシステムを関連づける。その目的は、ビジネス構造とシステム構造の関係を可視化することである。ビジネス構造の変更に伴うシステム環境変更規模の把握や、システム刷新による関連事業に与える効果（効率の向上、販売チャネルの増加）の評価が計画時に容易に行うことが可能になる。左図はトレーサビリティの一例を概念的に示したものである。事業（ビジネス）には、ビジョン（目標）と戦略がある。事業内部ではビジョン実現のため、戦略をガイドラインに、ビジネスファンクション（機能）を構成する。ビジネスファンクションは階層的に分解された木構造になる。ビジネスファンクションは、各組織に分担される。ビジネスファンクション内部では、業務活動で蓄積し利用する情報モデルがあり、IT の世界ではこれは、データモデルとなる。情報モデルまでは、IT を意識しないビジネスの領域である。ビジネスの領域であるビジネスファンクション内部のビジネスプロセスを支援するシステム機能、もしくは、ビジネスファンクションそのものを自動化するシステム機能に SOA のサービスが関連づけられる。サービスを構成するオブジェクト、更に、データモデルを実現するオブジェクトが、アプリケーションアーキテクチャになる。アプリケーションアーキテクチャは、それを支えるデータ、RDB、サーバ等のテクノロジーアーキテクチャの上で実現される。ビジネスから IT への写像されたアプリケーションアーキテクチャが、SOA の領域となる。ここで、サービスを実装するときの重要なポイントは、ビジネス上の機能とシステム上のサービスを対応付ける方法である。左図の例では、実際の業務で提供するビジネス機能と、対応するサービスの粒度を合わせ、マッピングしている。このようにすることで、新規事業の立ち上げ時のビジネス・フロー構成など、ビジネスプロセスとシステムの合成を同じ単位で考えることが可能と

サービスとコンポーネント

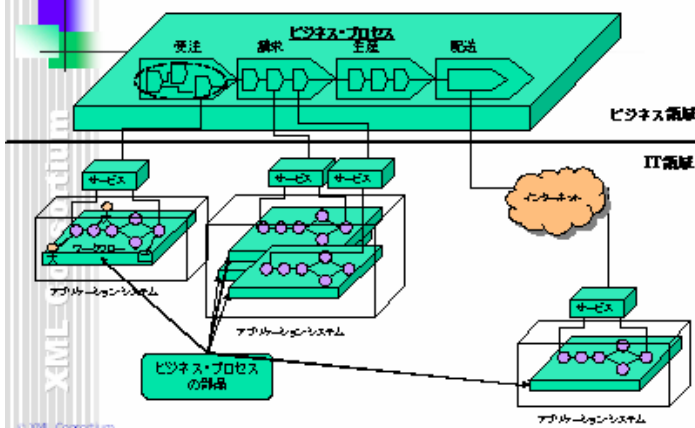


なる。

ソフトウェア部品と総称されるサービス、コンポーネント、オブジェクトの違いや関係はどうなっているのかを考えてみる。オブジェクトは、実行可能な最小単位であるプログラムを構成する要素である。その結合は特定の言語や実行環境に依存した密なものになり粒度は小さい。ソフトウェア部品と捉えたオブジェクトの主目的は開發生産性と品質の向上である。コンポーネントは、アプリケーション・システムを構成するプログラムである。Web の三層モデルのように、コンポーネントは複数のコンピュータに分散配置され処理を分担して行う。オブジェクトに比べ粒度は大きくなり、実行時に結合する遅延束縛(Late Binding)も実現され結合も緩やかになる。コンポーネントの主目的は、負荷分散と耐障害性確保である。

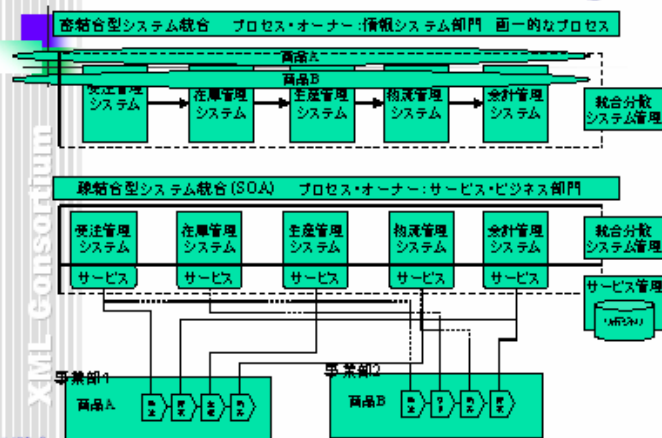
サービスの背景には、複数のシステムを組織化した大きなシステムの実現が背景にある。サービスは構成システムを抽象化したソフトウェア部品として位置付けられる。サービスは企業内、企業間のシステムで利用され、粒度も大きく、結合度合いもゆるい。結合するシステムの実装技術に依存しない疎結合が要求されるのは、異なる業務領域で運用されるシステム間の結合であるために互いに強く束縛することができないためである。

ビジネス・プロセス部品の組合せ



従来、現実のビジネスプロセスと IT 領域で設計するアプリケーション・モデルの関連が明確でないことが多く、ビジネスファンクションの追加やビジネスプロセスの変更に伴うシステム変更の規模や影響範囲を把握するために多くの時間と工数を必要とした。ビジネスプロセスは、ビジネスの機能で構成されている。ビジネスの機能の中で、システムで構築されるものは、SOA のサービスとしてビジネスプロセスの部品となる。つまり、SOA では、ビジネス領域と IT 領域のサービスが写像で表されるため、双方を連動させて管理することが容易になる。

ビジネスプロセス実現方式の比較



ビジネスプロセスを支援するシステムの実現には、2つの方法が考えられる。1つは、関連する情報システムを直列に結合しプロセスを形成する密結合型システム統合である。もう1つは、関連する情報システムそれぞれがサービスを提供し、別システムでビジネスプロセスに応じたプロセスを形成する疎結合型システム統合である。

密結合型システム統合では、情報システム部門がシステムとプロセスを管理する。種類のビジネスプロセスを共通化して、大量なトランザクション処理を実現する場合に向いている。基幹システムを直結するため、プロセスの変更はシステムの修正を必要とする。そのため、一つのビジネスプロセスの変更要求が他のビジネスプロセスに影響を与える可能性がある。疎結合型システム統合は、基幹システムの管理は情報システム部門が担当し、ビジネスプロセス毎のシステムのプロセスを実現するシステムは事業部が担当する。異なるビジネスプロセス毎に基幹システムのソフトウェア部品を組み合わせる。プロセスは事業部が管理するため、他の事業部や情報システム部門と調整することなく変更要求都度即時にプロセスを変更することができる。既存システムの機能や運用の体系を保持し、全体として、拡張性と柔軟性を維持し続けることがメリットである。

SOAのメリット

- 変化への対応
 - 迅速なビジネスプロセスの組み換え
 - ビジネスとIT間のトレーサビリティ
 - 事業者部門によるビジネスプロセス管理
- 変化の検出
 - 組織(企業内、企業外)を横断したビジネス・モニタリングの実現

ここまでのSOAの説明からSOAのメリットをまとめてみる。1つは、変化への対応が可能なことである。それは、迅速に変化に対応した組み換えが可能であることやビジネスとITのトレーサビリティによる連動性による。更に、オーナーとして、事業部門がビジネスプロセス管理を行うことも、変化への対応の行動が行える。次に変化の検出がメリットとしてあげられる。それは、企業の部署の壁をない中で、更に企業外を含めて横断したモニタリングを行うことで変化のタイミング、状況を検出することができる。

5. SOA のビジネス上での意義

変化に対応する

- ビジネス構成、IT構成、双方の関連の把握が必要
- ビジネス・プロセスの組立、組み換えが容易
- 企業間でコンテンツやビジネス・ファンクションの合成が容易



- ビジネスとITの統合的な管理機構(リポジトリ)
- ソフトウェア開発(新規, 変更)のスピード
- ビジネス・プロセスの部品化、情報の部品化

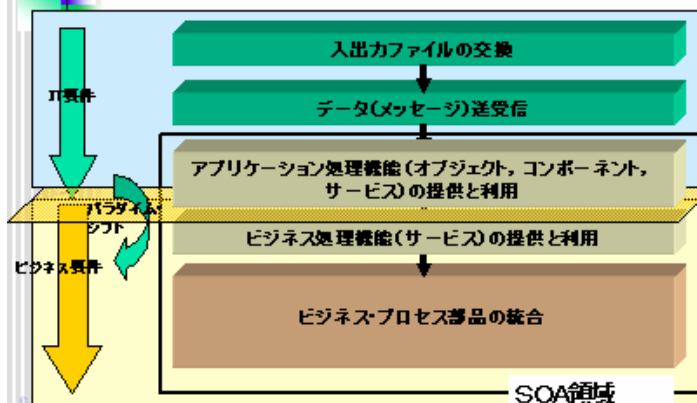
© XML Consortium



ビジネスと SOA といった観点で説明を行ってきたが、最初に説明を行った変化に対応をするといった部分を改めて説明を行う。

変化に対応をするためには、ビジネスの構成と IT の構成以外にトレーサビリティを実現する関連を知る必要がある。また、ビジネスファンクションから成り立つビジネスプロセスは、組み立てや組み換えが容易である必要あり、構成要素のビジネスファンクションは、企業内外を問わず合成が容易である必要がある。これを実現するためには、トレーサビリティを実現するビジネスと IT の管理機構が必要である。また、ビジネスプロセスの構成要素であるビジネスファンクションの部品化や情報を部品化する必要がある。これら、ビジネスと IT のトレーサビリティ、ビジネス機能の部品化、そしてビジネスプロセスの組み換えといった必要条件を満たすアーキテクチャが SOA である。

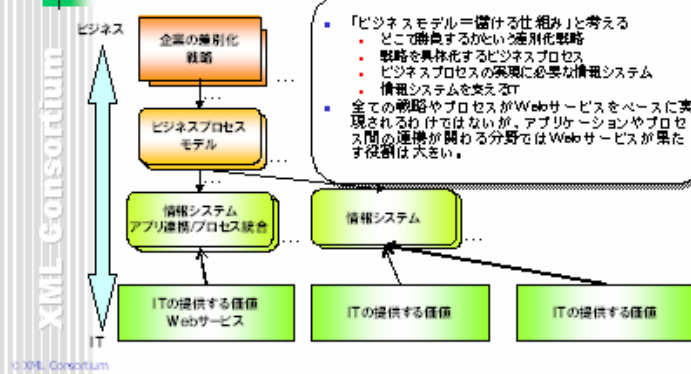
システム連携の進化



従来からシステム連携といった要件や実現方法は存在した。その移り変わりを見てみる。従来から、入出力データの交換や、メッセージの送受信といったデータのやり取りといった IT の要件に基づく考えで行われてきた。今後もこの連携は IT 技術としては重要であるが、ビジネスとの融合を考えた場合には、ビジネス要件にそって、アプリケーション機能の提供や利用、更に、ビジネスファンクションの提供や利用といった領域に進む必要がある。そして、ビジネスファンクションを融合することで、ビジネスプロセスの組み換えが可能となり、ビジネスと IT の完全なる融合となりえる。

SOA とは、ビジネス要件を中心に IT 要件を満たした領域であるアプリケーション部品からビジネスプロセス部品による融合を指している。

ビジネスからITに到る道



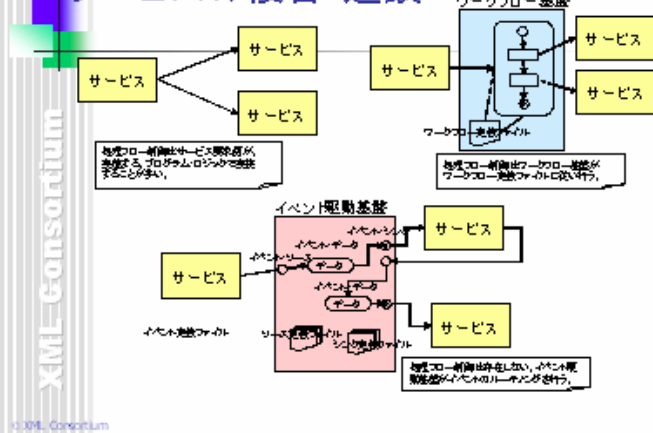
次にビジネスとITの筋道を考えてみる。まず、ビジネスモデルの基本は、ビジネスとして成り立つことであり、それは、つまり儲けることである。企業は、ビジネスとして成立することが絶対条件であり、実現するための差別化や、戦略の具体化を行う必要がある。これは、いくつものビジネスプロセスから構成されている。このビジネスプロセスを実現するために、情報システムは、アプリケーション連携や統合、独立した情報システムというもので構成されている。

この情報システムは、SOAのサービスとして提供されるビジネスファンクション、代表的には、Webサービスから構成される。このように、現在のビジネスとITの関わりは、企業戦略まで結びつく密接な関係になってきている。

システムのビジネスファンクションは、一概にWebサービスである必要はないが、戦略的でグローバルなシステム連携という意味では、現在、Webサービスは不可欠な技術として存在する。

6. サービスのオーケストレーション

サービスの複合・連鎖



ビジネスとITといった観点のSOAから、SOAの内部を見てみる。

まず、SOAのサービスは、複合や連鎖を行うことで連携を行うことが可能となる。このサービスの複合・連鎖にはいくつかのパターンを挙げることができる。

- ・ サービス要求側が処理フロー制御を行う。

⇒直接統合型

- ・ ワークフロー基盤が処理フロー制御を行う。

⇒ワークフロー基盤型

- ・ イベント駆動基盤によってイベントのルーティングを行う。

⇒イベント駆動基盤型

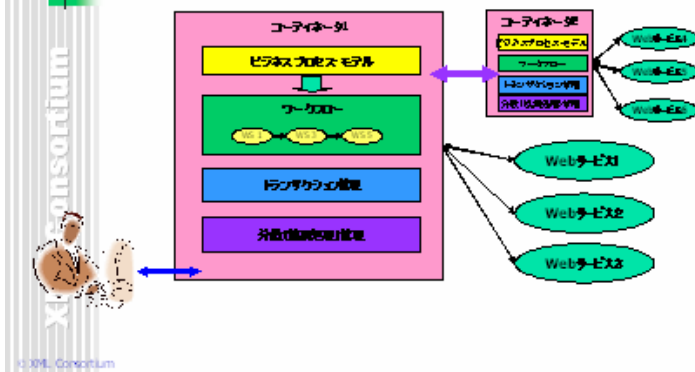
サービス複合パターンの特徴

	直接統合	ワークフロー基盤統合	イベント駆動基盤統合
同期性	同期	同期/非同期両方	非同期
迅速構築性	○	◎	△
可観性	△	△	○
拡張性 サービス追加	△ プログラム修正有り	○ ワークフロー修正	◎ 設定変更
状態監視容易性	◎	○	△
ビジネス・プロセス実装	○ ビジネス・ロジックで実装	◎ ワークフロー定義言語で記述	△ ビアビタ間のルーティングのみ
基盤確立初期コスト	◎	○	△
用途	拡張頻度の少ない 固定的な連携に向く	統合サービスの構築 やビジネス・プロセスの部品化に向く	不特定多数のプロセスの結合に向く

SOA システム構築パターンにおける3つのサービス複合パターンの特徴を表にまとめたものである。ニーズや要件に適合するよう、これらのサービス複合パターンを選択または組み合わせる必要がある。

特徴から判断できる用途としては、直接結合型は、組み換えが簡単に行えないものの他の実装コストのみで実現でき、同期処理も行えることから、組み換えが行われない固定的な連携の処理に向く。ワークフロー型結合は、ワークフローエンジンが必要なものの組み換えが行いやすい点から、統合サービスやビジネスプロセスの部品化の処理に向き、イベント駆動型は、独立性はあり、エンジンも必要がないが、状況の把握などに工夫がいることから、不特定多数のプロセスの結合に向く特徴がある。

ワークフロー基盤の実現



ワークフロー基盤には、ビジネスプロセス・モデルに従ってワークフローを定義、実行する機能ほかに、トランザクション管理機能、分散(協調処理)管理機能、また、他のワークフロー基盤との調整を行うコーディネータ機能が必要となる。

7. ワークフロー

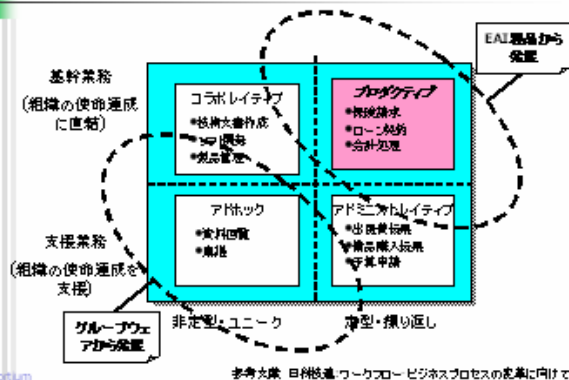
ワークフローの定義

- いろんな見方がある
 - コンピュータによるビジネスプロセスの促進・自動化
 - 個別のプロセスやアプリケーションを連携させる“のり”や“ベルトコンベア”
 - 昔ワークフローと言っていたものがビジネスプロセスマネジメントに進化した

SOA によるビジネスプロセスの実現にあたって、ワークフローについてももう少し見てみる。ワークフローとは、基本的には、ビジネスの手続きを自動化するか、手続きの処理手順を規定することにより、関係者の間を情報や業務が円滑に流れるようにすることである。もしくは、そうして作られた流れを意味するが、「コンピュータによるビジネスプロセスの促進・自動化」「個別のプロセスやアプリケーションを連携させる“のり”や“ベルトコンベア”」ととらえることにより、ワークフローはビジネスプロセスマネジメントへと進化したとも考えられている。

実際には、ワークフローの解釈/表現はさまざまである。

ワークフロー利用形態と製品



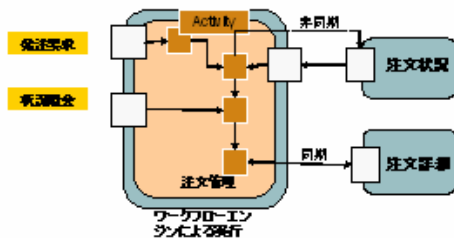
一概にワークフロー製品と言われても利用形態によりセグメント分けされる。

ワークフローの利用形態は、 基幹業務 / 支援業務

非定型・ユニーク / 定型・繰り返し、といった二次元の軸によって、コラボレイティブ、アドホック、プロダクティブ、アドミニストレイティブといった4つの製品タイプにセグメント分けされる。資料回覧、稟議などのアドホック・タイプの製品はグループウェアから発展し、保険請求、ローン契約、会計処理などのプロダクティブ・タイプの製品はEAI製品から発展してきたものである。

Webサービスベースのワークフローとは

- Webサービスベースでビジネスプロセスを組み立てるXMLボキャブラリ(e.g.BPEL4WS)を使用
 - できなかったプロセスはWebサービス
 - 部品として使うプロセス/アクティビティもWebサービス

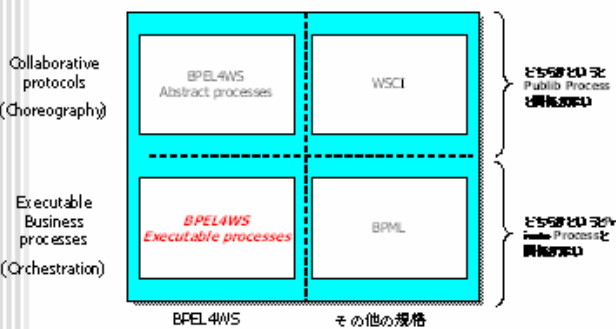


それでは、Web サービスベースでのワークフローとはどういうものかを見てみる。Web サービスベースのワークフローは、BPEL4WS 等の Web サービス・XML 技術を利用して、ワークフローを定義し実行することでビジネスプロセスを実現する。

例えば、発注要求・状況照会などのビジネスプロセスを実行するにあたって、一連のワークフローの定義・アクティビティをWeb サービスベースで実装することができる。

また、ワークフローを構成する注文状況確認や注文詳細確認等のプロセスについても Web サービスで実装することもできる。

オーケストレーションとコレオグラフィ



Web サービスワークフローを考える場合、オーケストレーションとコレオグラフィを考える必要がある。IEEE Computer Vol36 によると、BPEL4WS や WSCI・BPML は左図の様に分類される。主にオーケストレーションはアプリケーションの起動シーケンスや制御フローの実行可能なシナリオを定義するプライベートプロセス的なものと考えられ、コレオグラフィはアプリケーション同士が連携する際の上位プロトコルを定義するパブリックプロセス的なものと考えることができる。

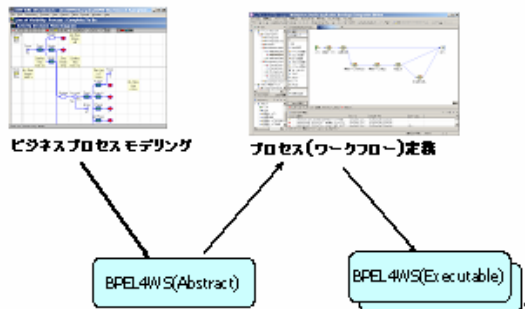
ワークフロー記述言語: BPEL4WS

- Business Process Execution Language for Web Services
 - マイクロソフトのXLANGとIBMのWSFLを統合
 - OASISのWSBPEL TCで標準化活動中
 - PublicとPrivate両方のプロセスを実行可能

BPEL1.1の仕様は <http://www.ibm.com/developerworks/webservices/library/ws-bpel> を参照

BPEL4WS とは、ワークフローを記述する言語であり、ワークフローを構成するパブリック、また、プライベートプロセスの全てを網羅する形で標準化が進められてきた。標準化にあたっては、OASIS のWEBPEL TCが中心となり、マイクロソフトのXLANG、IBM のWSFL の双方の規格を統合する形で策定が進められ、現在のステータスはVersion1.1 (2005 年 2 月) となっている。

ワークフロー(プロセス定義)作成手順



© XML Consortium

ビジネスプロセスのモデリングと IT プロセス設計は目的が異なる。ビジネスプロセスのモデリングは、現実の複雑な業務を、詳細を省き大枠で捉えることを目的にしている。ビジネスプロセス・モデルは実行することはできない。IT プロセス設計は、実行可能なワークフロー定義を作成することが目的である。データの定義や処理フローの記述、例外処理の条件指定などプログラミング・レベルの記述が要求される。ビジネスプロセス・モデルは、システム化するための有効な入力情報になる。しかし、モデル駆動的なアプローチで IT プロセスのワークフローに自動的に変換することはできない。

また、ビジネスプロセスが明示的なワークフローにより実現されることは部分的で、その他はデータの共有やサービスの共有、バッチプロセスの実行名で実現される。ビジネスプロセス・モデルからワークフローの設計は、モデル記述を複数のワークフロー記述に分解し詳細化することで行う。

BPEL4WS を用いる場合、ビジネスプロセスモデリングを実施した結果が、Abstract Processes として記述され、プロセス (ワークフロー) 定義については、Executable Processes として記述される。

ビジネストランザクションの実現

課題

- 生存時間の長いトランザクション
 - リソースを長時間ロックし続けることはできない
- トランザクションに参加する Web サービスの実装はさまざま
 - Proprietary な Coordination プロトコルは使用できない

解決策

- 補償 (Compensation) トランザクションの導入 (BPEL4WS)
- オープンな Coordination プロトコルの採用
 - WS-Coordination
 - WS-Transaction

© XML Consortium

ビジネスプロセス・ワークフローを Web サービススペースで実現する場合、ビジネストランザクションを実現する必要性が生じる。それは、ビジネスプロセス・ワークフローによっては、トランザクションを長時間にわたり何らかの形で維持する必要が発生し、その際、ビジネスプロセスを共有する企業間・システム間でオープンなプロトコルを用いる必要がある。

これらは、BPEL4WS や WS-Coordination、WS-Transaction 等の標準規格を用いて実装することで解決を図ろうとしている。

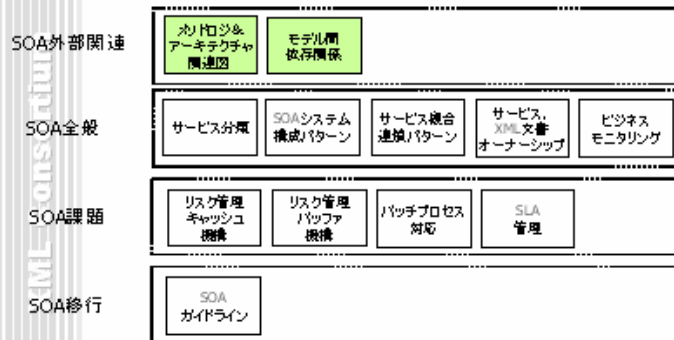
(参考資料)

※ 1 : WS-Coordination

※ 2 : WS-Transaction

8. SOA の実現化

アーキテクチャパターン、デザインパターン、ベストプラクティス

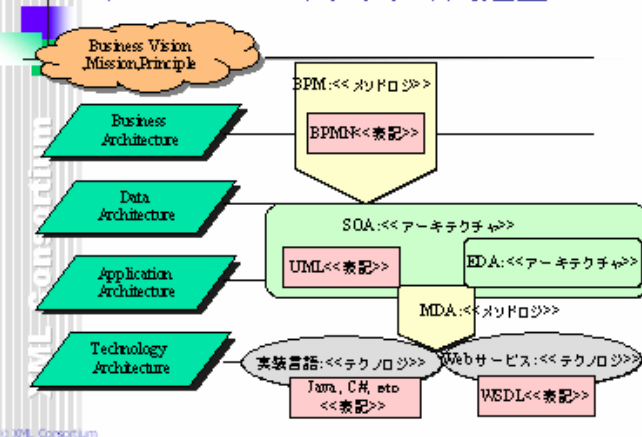


SOA を実現するためには、SOA の構成要素であるアーキテクチャパターン、デザインパターン、ベストプラクティスについて述べる。説明は、SOA 外部関連、SOA 全般、SOA の課題、SOA 移行という4つの視点の中で、課題以外について解説する。まずは、SOA 外部関連から解説を行う。

(参考資料)

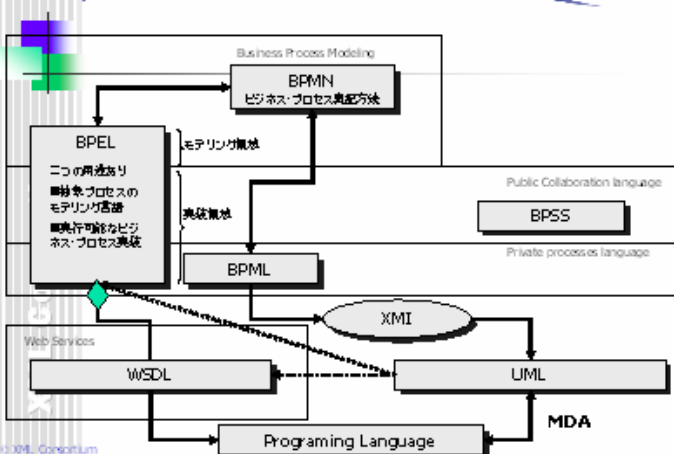
※ 3 : SOA の課題

メソッドロジ&アーキテクチャ関連図



ここでは、SOA を取り巻くメソッドロジ・アーキテクチャについて述べる。まず、上位のビジネスアーキテクチャレイヤにおける代表的なメソッドロジとしてBPMがある。BPMは、データアーキテクチャ/アプリケーションアーキテクチャレイヤにおけるSOAを紐づく位置付けにある。SOAにはEDA(イベント駆動型アーキテクチャ)が包含される。アプリケーションアーキテクチャレイヤからは、メソッドロジのMDAを介して、最下位のテクノロジーアーキテクチャレイヤのインプリメンテーションが生成される。テクノロジーアーキテクチャには、Webサービスや実装言語が含まれる。

SOA,BPM領域技術マップ

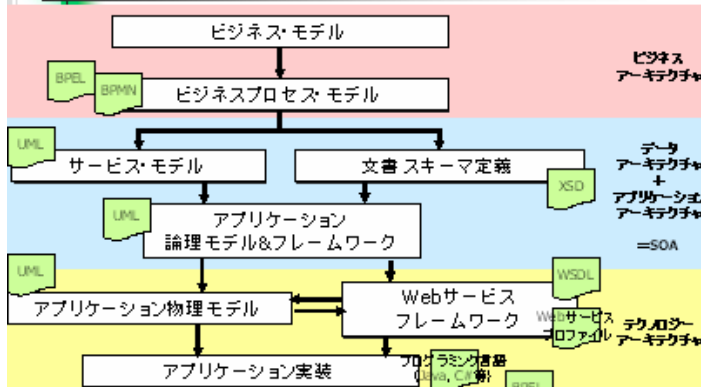


SOA・BPMを実現するに当たって、現状では、左記の技術(規格)が登場しており、それぞれの技術(規格)はXMLを用いるため、相互に連携が可能である。

(各用語については、用語説明を参照)

モデル間依存関係

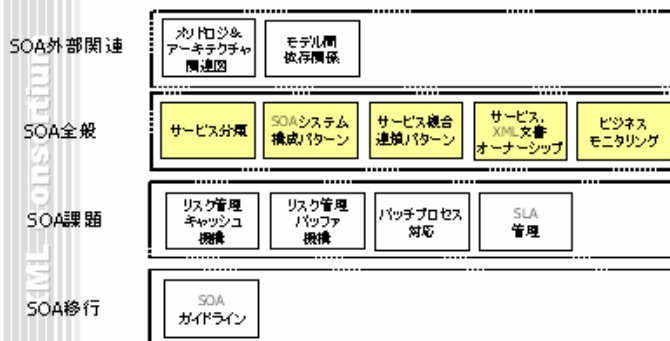
モデル仕様変更の影響関係が矢印



アーキテクチャとして、ビジネス、データ、アプリケーション、テクノロジーアーキテクチャに分けて考えてみる。

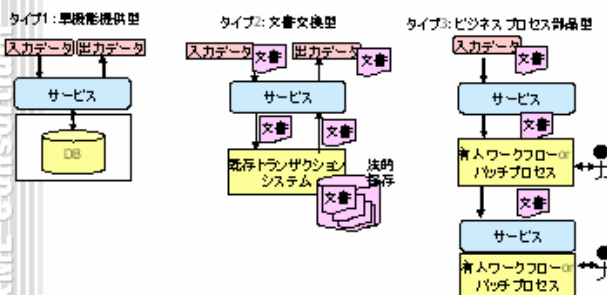
各アーキテクチャレイヤに属するモデルは、図のとおりである。ビジネスモデルは、ビジネスプロセスモデルになり BPEL や BPMN 等で表現される。この領域がビジネス領域であり、データアーキテクチャレイヤ以降は、IT 領域に分類される。IT 領域であるアプリケーション論理モデルやスキーマ定義等は、ビジネスと実装をつなぐ領域として SOA の領域となる。テクノロジーアーキテクチャでは、アプリケーションとして実装が行われる。ビジネスモデルの仕様変更は、トップダウン的に全てに影響し、粒度は、トップダウン的に細くなる。

アーキテクチャパターン、デザインパターン、ベストプラクティス



SOA 全般に関わるカテゴリにおいて、サービスの視点から、サービスのタイプやサービス間のメッセージ構造、更にビジネスモニタリングなどについて解説する。

サービスの分類



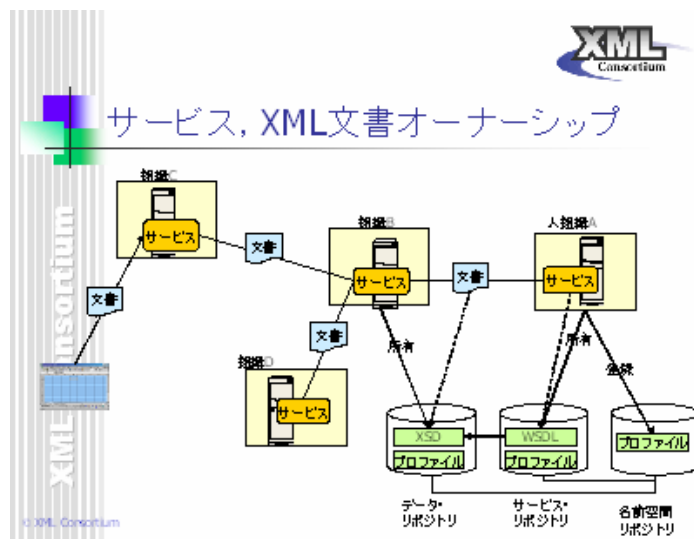
サービスをパターン化してみると、「単機能提供型」、「文書交換型」、「ビジネスプロセス部品型」の3つに分類される。

1つ目の「単機能提供型」は、単一サービスとしての利用パターンである。データベース検索などシンプルな機能のサービス化が典型的な例となる。

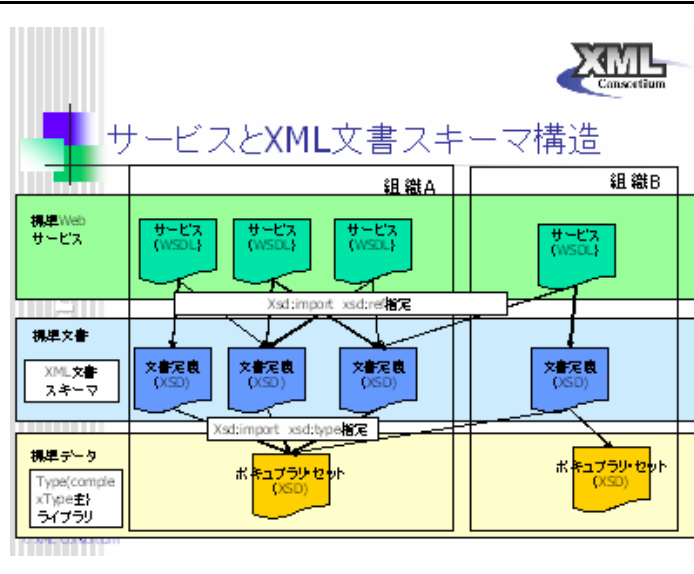
2つ目の「文書交換型」は、ある程度まとまった機能を持ったシステムのサービス化またはサービス連携のパターンである。生命保険や自動車保険の見積り依頼を行ない、見積算出システムで処理を行ない、見積回答を行うといった文書交換を伴うようなシステムが、典型的な例

となる。

3つ目の「ビジネスプロセス部品型」は、複数のサービス連携においてワークフローを伴うパターンである。受注・与信・在庫引当・出荷指示といった一連のプロセスにおいて、各機能がサービス化されており、人為的な処理を含め、業務全体としてワークフローを伴うシステムが典型的な例となる。



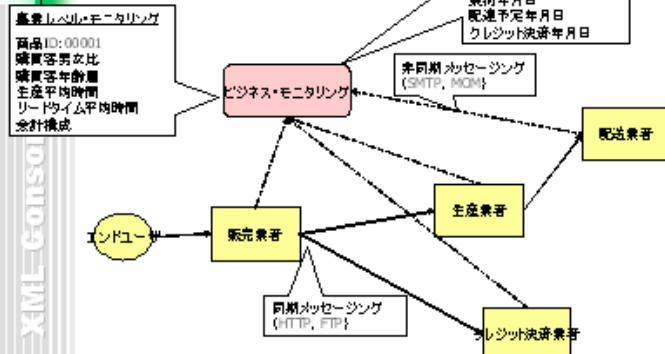
SOA では、各アプリケーションが高度なビジネス機能をサービスとして公開し、他のアプリケーションが利用できるようにする。サービスは、WSDL を使用して、サービスがサポートするすべてのメッセージ内容を詳細に指定する。また、WSDL は、XML スキーマ定義(XSD)を使用して、各メッセージの内部構造とメッセージ要素の制約をドキュメント化する。WSDL、XSD は、それぞれがサービス・リポジトリ、データ・リポジトリを使用することにより、サービス間のやりとりが大幅に軽減され、サービス間でのXML 文書送受信がスムーズに行われる。



SOAのサービスとして標準的とされるWeb サービスでは、提供するサービスの内容をWSDLにて記述する。このWeb サービス上でやりとりされる標準的な文書構造は、XSD を用いて定義することが可能である。そして、その標準文書内で使用されるデータのスキーマも、XSD で定義することができ様々な文書定義から参照され、共通的なボキュブラリセットとして使用される。

これらの文書定義や標準データ(ボキュブラリセット)は、組織間で共有することが可能である。

ビジネス・モニタリング

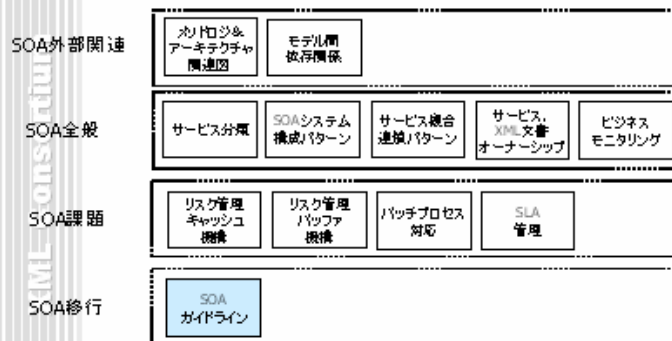


SOAにおけるモニタリングは、システムの異常監視ではなく、ビジネスの観点からプロセス監視を行うことを指す。

販売業者、生産業者、配送業者、クレジット決済業者が携わる一連のビジネスプロセスを、事業レベルや業務モデルといった様々な観点からモニタリングでき、業務プロセスに影響を与えることなく、モニタリング情報を集約することが可能となる。また、モニタリング対象の構成を自由に組み替えることも容易となる。

このビジネスプロセスをリアルタイムにモニタリングするというアプローチは、高いROIを達成するテクノロジーとして注目されている。

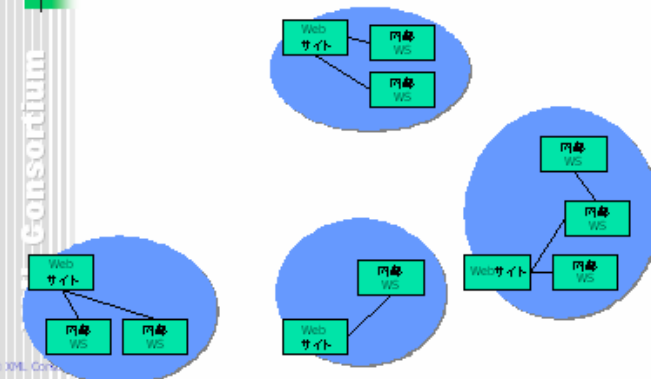
アーキテクチャパターン、デザインパターン、ベストプラクティス



次に、システムのSOAへの移行について解説する。

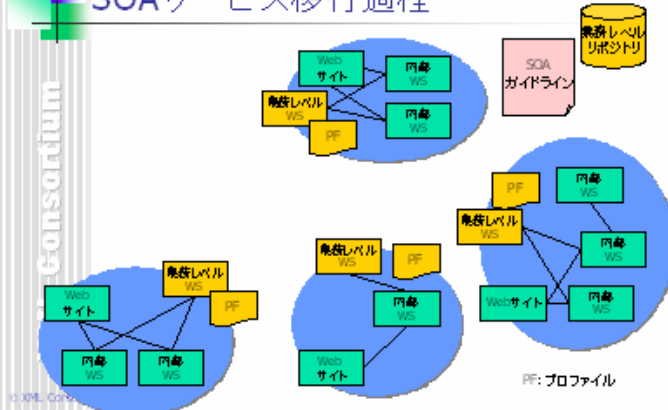
実際の移行にあたっては、ガイドラインの策定が重要となる。

SOAサービス移行過程



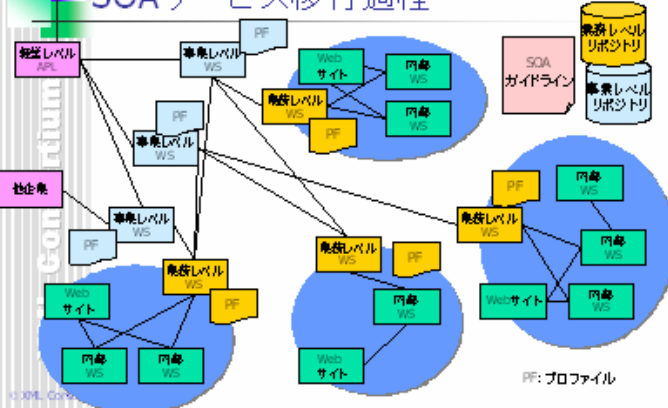
SOA サービスへ移行する前段階として、まず点在するWebサイトに閉じた形で、サイト内部でのWebサービスを使用している状態を仮定する。

SOAサービス移行過程



SOA ガイドラインを元に、Web サイト内の Web サービスを組み合わせることで業務レベルの Web サービスおよびプロファイルを定義し、業務レベルリポジトリに登録する。

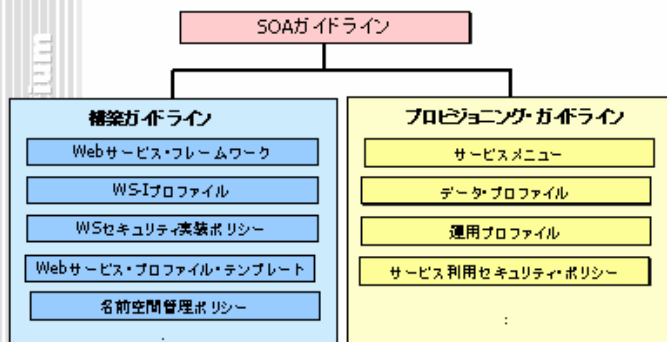
SOAサービス移行過程



最後に、業務レベルの Web サービスを組み合わせることでさらに上位である事業レベルの Web サービスおよびプロファイルを定義し、事業レベルリポジトリに登録する。事業レベルの Web サービスは、経営レベルのアプリケーションや他の企業から利用される。

このように、レベル、リポジトリを定義したガイドラインを元に SOA へ移行・構築していくものと考えられる。

SOAガイドライン



移行過程で記載を行った SOA のガイドラインには、SOA を構築するためのガイドラインとサービスを提供するためのプロビジョニングガイドラインに大別される。

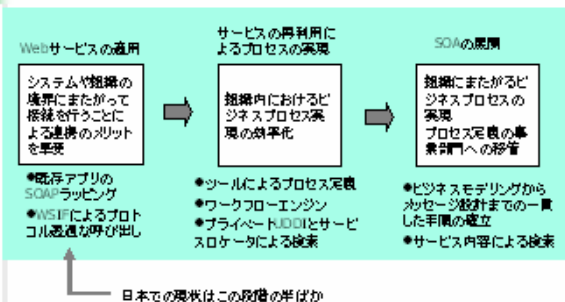
SOA 構築においては、サービスのフレームワークや基本ともいえる WS-I プロファイル、セキュリティポリシー等が必要となる。

次にプロビジョニングガイドラインでは、サービスのメニューや提供するための運用プロファイルやセキュリティポリシー等が必要になる。

ガイドラインでは、規模、内容によって内容は異なるが、構築する場合と提供する場合における意味合いの違いを認識して作成される必要がある。

9. まとめ

SOAへの発展



SOAの発展は、まずは、「Webサービスの適用」段階からはじまり、「サービスの再利用」段階、「SOAの展開」段階への発展が考えられる。現在の日本ではWebサービスの適用段階と言える。「Webサービスの適用」段階では、WSDLによるインターフェイス定義とアプリケーションのラッピングが行われ、「サービスの再利用」段階では、ツールによるプロセス定義やワークフローによる結合、そしてUDDIによる検索が行われる。「SOAの展開」段階では、ビジネスモデリングからメッセージ設計まで一貫した手順が確立され、サービスの内容、レベルによる検索が行える。SOAは、このような段階を通じて、発展していくと考えられる。

SOA活用のために成すべきこと

1. ビジネスプロセス統合計画の作成(AsIsとToBe)
2. 既存システムのサービス化
3. サービスリポジトリの導入
4. ビジネスプロセス統合基盤(ツール・体制)の整備
5. ビジネスオーナー支援の実現

SOAの発展・活用にあたって以下のポイントが重要となる。

(1) ビジネスプロセス統合計画の作成

現在のビジネスプロセスの把握・定義、将来あるべき姿のビジネスプロセスの策定・定義と、段階的移行(統合)計画の作成。

(2) 既存システムのサービス化

既存のシステムをサービス化し、プロトコル透過にそのサービスを利用可能とすること。

(3) サービス・リポジトリの導入

サービスのメタ情報を一元管理し整合性を保つ為のリポジトリを導入すること。

(4) ビジネスプロセス統合基盤の整備

ビジネスプロセスを統合するための、ツールやインフラを整備すること。

(5) ビジネスオーナー支援の実現

企業戦略に基づいてビジネスプロセスを定義したビジネスオーナー自身が、ビジネスプロセスの現在の状態をモニタリング、または市場の変化に対応すべく、ビジネスプロセスを迅速に変更できるような支援機能を実現すること。これらにより、SOAは実際の場面で活用され、発展を遂げていく。

参考資料

※ 1. WS-Coordination

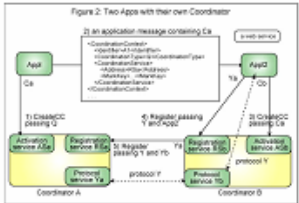


Figure 2: Two Apps with their own Coordinator

WS-Coordination は、分散アプリケーションの動作を調整するプロトコルを提供する、拡張可能なフレームワークを定義している。このような調整プロトコルを使用して、分散トランザクションの結果が一貫した合意に至る必要があるアプリケーションなどを含む多数のアプリケーションをサポートしている。

アプリケーションサービスは、

WS-Coordination で定義するフレームワークを使用することにより、アクティビティを他のサービスに伝播させる必要があるコンテキストを作成を可能とし、さらに、調整プロトコルに登録できる。既存のトランザクション処理、ワークフロー、およびその他の調整用のシステムは、このフレームワークを使用することにより、独自のプロトコルを隠蔽でき、異種環境で運用できるようになる。

Coordination 用のプロトコル自体は WS-Transaction によって定義される。

※ 2. WS-Transaction

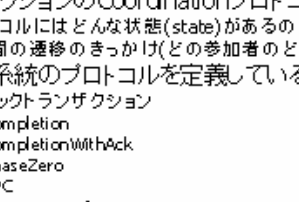


Figure 2: Two Apps with their own Coordinator

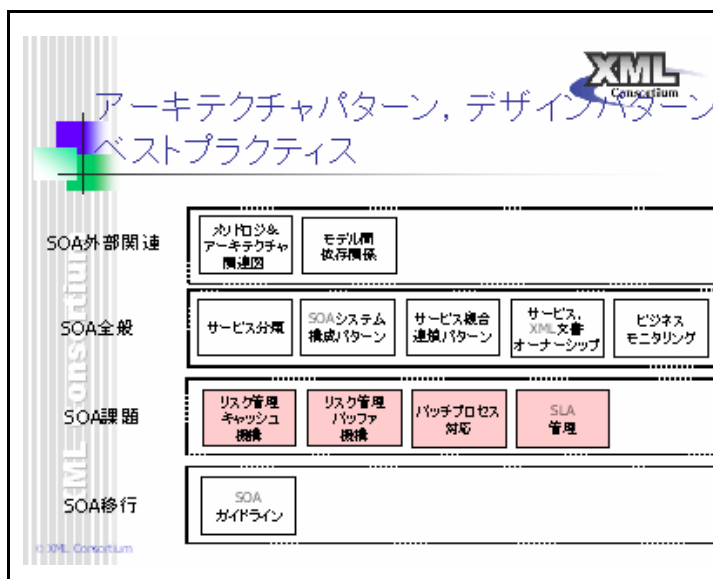
WS-Transaction では、WS-Coordination 仕様に記述された拡張可能な調整フレームワークで 사용되는調整の種類について記述している。WS-Transaction では、2 つの調整の種類 “アトミック トランザクション (AT: Atomic Transaction)” および “ビジネス アクティビティ (BA: Business Activity)” を定義する。開発者が、分散アクティビティの結果が矛盾なく一致する必要があるアプリケーションを作成しているときに、これらの調整の種類のいずれか、または両方を使用できる。

アトミック トランザクション ?制限された信頼ドメイン内で短期間実行されるアクティビティを調整するために使用する。このようなアクティビティは、すべて実行されるかまったく

実行されない（“all-or-nothing”）という属性を持っているので、アトミック トランザクションと呼ばれている。

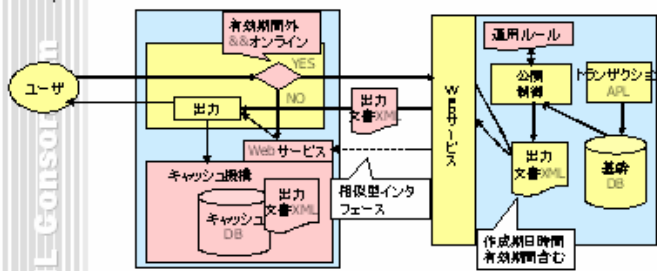
ビジネス アクティビティ プロトコルは、長期間にわたり実行され、ビジネスの例外を処理するためにビジネス ロジックを適用したいアクティビティを調整するために使用する。長期間の実行では、動作を一時的なものにし、他のアプリケーションから隠ぺいするために、データリソースをロックすることが禁止される。代わりに、動作は即時適用され、永続的なものとなる。ビジネス アクティビティ仕様では、既存のビジネス プロセスやワーク フロー システムが専用のメカニズムをラップして、信頼関係の境界や異なるベンダの実装にまたがって相互運用できるようにするプロトコルを定義している。

※ 3. SOA の課題



SOA における課題について解説する。課題としては、障害としてのキャッシュやバッファ機能や、バッチプロセスへの対応、更に SLA への対応があげられる。

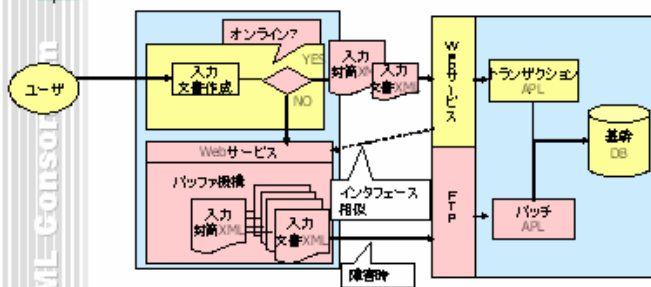
キャッシュ機構 (照会・参照WS障害対応)



まず、照会や参照を行う場合、障害がある場合に備えたキャッシュ機構が必要となる。

Web サービスを提供する側においても、運用規定は存在する。その場合に、利用者としては、その運用規定外の場合の対応方法といった点が問題となる。こういった場合には、条件によっては、既にキャッシュされたサービスを読み出し対応するといった仕組みが必要となる。ただ、この場合にも、キャッシュのタイミング等の問題点が存在する。

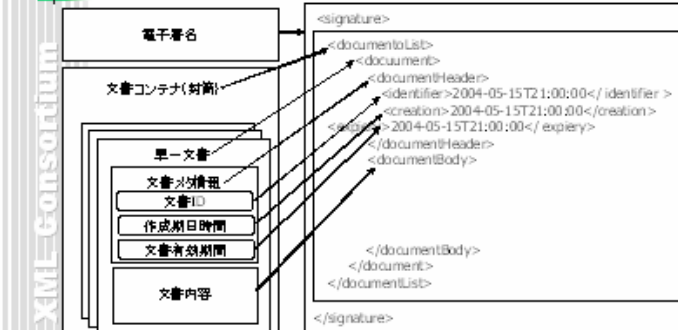
バッファ機構 (更新WS障害対応)



照会や参照の場合には、キャッシュしたものを利用するといった方法が考えられるが、更新する場合には、トランザクションの問題もあり、同じようには行えない。この場合には、一度バッファに保存するイメージで処理を継続し、オンライン時に更新を行う方法や、障害が発生している場合には、データの別送信から、バッチアプリケーションでのデータ更新といった方法が考えられる。

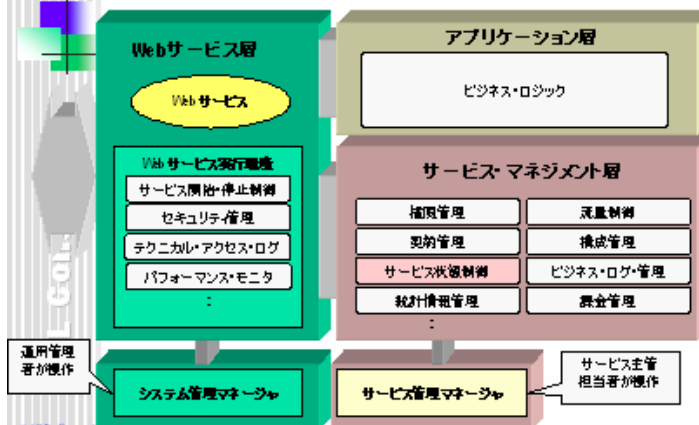
ただし、この場合には、最初に記載したトランザクションの処理を如何にするのか、どこでバッファリングを行うのかといったアーキテクチャが必要になる。

障害対応文書構造例



障害が発生した場合には、障害対応文書が必要になる。このスライドに記載した例では、日付、有効期間そして内容の通知を行っている。文書構造以外にも、こういった文書の取り扱いについてもシステムの対応が必要になる。

SLA運用・管理アーキテクチャ



状態遷移で説明した内容は、アプリケーション層とは別にサービスマネジメント層として必要になってくる。その内容としては、サービス管理マネージャへの通知及び情報の蓄積が必要である。一方、Web サービス層では、実行状況を把握する必要があり、提供者として、Web サービスシステム管理者が操作を行える必要がある。

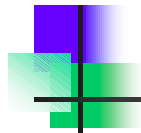
但し、Web サービスの粒度、SOA のレベルによっては、これらの管理が必須な部分とアプリケーションレベルで管理できる部分があるため、構築時の検討が必要になってくる。

用語説明

用語	説明
BPEL (Business Process Execution Language for Web Services)	XML ベースのワークフロー記述言語。複数の Web サービスを連携させることで、複雑なプロセスフローを定義することができる。 ビジネスプロセスを記述・設計するための言語としての側面と、ビジネスプロセス・エンジン（オーケストレーション・エンジン）などと呼ばれる実行環境で、Web サービスによるプロセスを実行するスクリプト言語としての側面がある。
BPMN (Business Process Modeling Notation)	実行可能な BPEL4WS の生成を可能にする内部モデルとしても利用できるため、ビジネスプロセス設計とプロセス実装間のギャップに標準化されたブリッジを作る。BPMN は、Business Process Diagram (以下、日本語ではビジネスプロセス図)を定義する。この BP 図は、ビジネスプロセスオペレーションの可視化モデルを作成するため、特別に仕立てたフローチャーティング技術をベースにしている。ビジネスプロセス・モデルは、アクティビティ（例：作業）やそれらの実行順序を規定するフロー制御などからなるグラフィックオブジェクトのネットワークである。
BPML (Business Process Modeling Language)	この仕様は WSCI をさらに活用するための抽象／実行イメージプロセスを表現するビジネスプロセスモデリング仕様。BPML は、最終的に BPML プロセスとして実行される企業内でのプロセスの実装と、WSCI によって定義される企業間のインターフェイスに変換される。BPML は、異なるプラットフォーム間でビジネスプロセス定義を交換することを可能にするもので、IBM が開発した WSFL や Microsoft の Xlang と競合するもの。
ebXML BPSS (Business Process Specification Schema)	商取引を行うビジネス・パートナー間で交換するメッセージの順番や、メッセージ自体といったビジネスプロセスを記述するための仕様。あくまでもビジネス・パートナーの間でやり取りする処理の流れ、いわゆる「パブリックフロー」を定義するもので、パートナー内部で実行する「プライベートフロー」とは関係ない。このことから ebXML BPSS は、取引における一種の契約情報を表現する手段と見なすこともできる。
BPSS (Business Process Specification Schema)	ebXML におけるビジネスプロセス定義のためのスキーマ。 BPSS では、複数のビジネス・パートナー間の取引をビジネスコラボレーションと呼ぶ。 このビジネスコラボレーションの定義に必要な、ビジネストランザクション、ビジネスドキュメントフローといった要素や概念などを規定しているのが BPSS。
XMI (XML Metadata Interchange)	メタモデルを記述する為の文法 MOF(Meta Object Facility)によ

	って作られたメタモデルを交換するための標準仕様
WSDL (Web Services Description Language)	Web Service が提供する機能を記述するための、XML ベースの言語仕様の 1 つ。この WSDL は、同様の目的でマイクロソフトが開発した SDL/SCL と、IBM が開発した NASSL (Network Accessible Service Specification Language) を統合したもの。Web Service によって実現される分散オブジェクト環境において、各 Web Service オブジェクトがどのような機能を提供するのか、その機能を利用するために、どのようなパラメータを渡す必要があるかなどを標準的なデータ仕様で記述できるようにし、Web Service 同士のコミュニケーションを可能にする。 ・ service : port 要素の集合を定義する。 ・ port : 具体的なネットワーク・アドレスと binding 要素を関連づける事によって定義する。 ・ binding : prototype 要素で定義された操作と具体的なプロトコルとデータ・フォーマットを定義する。 ・ port Type : 抽象的な操作の集合を記述する。 ・ message : 交換されるデータを抽象的に記述する。 ・ types : 交換されるデータの記述に使用されるデータ型を定義する
UML (Unified Modeling Language)	統一モデリング言語で、オブジェクト指向で業務を分析したり、システムを開発したり、ソフトウェアモジュールを設計したりする際のダイアグラムの描き方・記法を定めた米国のオブジェクト技術標準化団体 OMG(※)の標準。 オブジェクト指向分析・設計の標準表記法 (※) OMG=ObjectManagement Group
リポジトリ	用語としては、貯蔵庫、倉庫という意味。システムにおいては、システムが活用すべきデータ格納を一般的に意味する。
SLA	サービスの提供側の QoS に関する取り組み。提供者は、サービスの品質を明確にして、顧客と締結する契約を意味する。
WS-Transaction:Completion	コーディネーターに対してアトミック トランザクションをコミットするか、またはアボートするかを通知するために使用するプロトコル。
WS-Transaction:Completion With Ack	Completion プロトコルと同義。Completion との違いは、コーディネーターがトランザクションを破棄する際に、トランザクション参加者から承認メッセージ (Notified メッセージ) が必要となる点である。
WS-Transaction:2PC	コーディネーターが、トランザクション上のデータ更新において、コミット処理を準備フェーズとコミットフェーズの 2 フェーズで管理するための調整プロトコル。

WS-Transaction:PhaseZero	2PCを行なうために、他のトランザクションが終了している状態であることを確認するためのプロトコル。このプロトコルがトランザクションが終了（=PhaseZero）の状態になっているのを認識してから、コーディネータは2PCを開始する。
WS-Transaction:OutcomeNotification	コーディネーターに対して、トランザクションがいつ終了し、トランザクションがコミットしたかアボートしたか、その結果通知をやり取りするプロトコル。



スライド出展：XML コンソーシアム テクノロジー部会活動発表資料より抽出

- 2004 年 1 月 22 日 XML コンソーシアム Day
2004 年 2 月 6 日 XML コンソーシアム大阪セミナー
「Service Oriented Architecture 概説」 牧野 友紀 (日本ユニシス)、
天野 富夫 (日本アイ・ビー・エム)
- 2004 年 5 月 17 日 コンソーシアム Week 「SOA 実現化の考察」 牧野 友紀 (日本ユニシス)
- 2004 年 5 月 17 日 コンソーシアム Week 「SOA とワークフロー」 天野 富夫 (日本アイ・ビー・エム)
- 2004 年 5 月 19 日 コンソーシアム Week 「テクノロジーMAP(SOA)」 XML テクノロジー部会

作成者： 芦田尚人 (ブレイニーワークス)
稲荷教司 (PFU アクティブラボ)
倉沢良明 (キヤノン)
永田明 (サンモアテック)
森本信次 (日本オラクル)
山本倫生 (東日本電信電話)

監修： 天野富夫 (日本アイ・ビー・エム)
牧野友紀 (日本ユニシス)
XML コンソーシアム BI 研究部会、SOA 部会