「コンポーネントベース開発(CBD)の実践」

WebSphere Business Components (WSBC)における コンポーネントとデザイン・パターンの適用

2002年 2月20日

日本アイ・ビー・エム株式会社
ソフトウェア事業部
永井 修

目次

WebSphere Business Components (WSBC)出現の背景

アプリケーション開発に費やせる時間が短くなった

Ready-To-Goのアプリケーションではビジネス・ニーズに柔軟に対応できない

レガシーアプリケーションを多種環境のシステムに統合したい

テスト済み部品を組み合せてアプリケーションができて、しかもそれがカスタマイズできたら…

コンポーネント技術

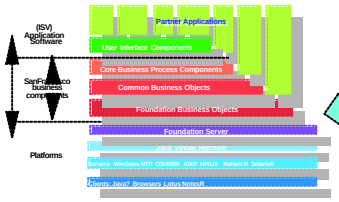
Javaの普及 EJB

WebSphere Business Components

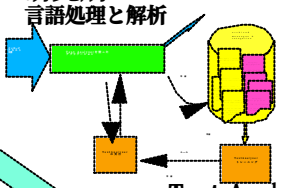
ビジネス・コンポーネント群
コンポーネント・サービス
ツール群 (開発支援用・構成支援用)

WSBC IBMのコンポーネント関連技術の集大成

SanFrancisco

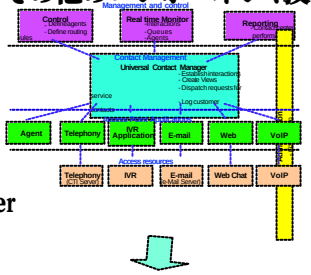


研究所
言語処理と解析



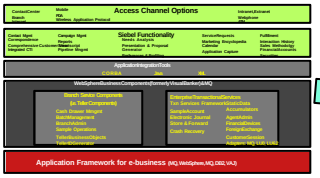
Text Analyzer

その他のコンポーネント技術

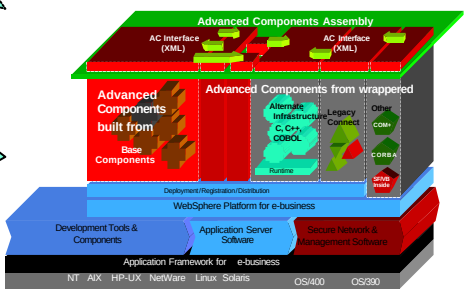


WebSphere Business Components Studio
Order Capture

Banking, Insurance, CRM Groups



WebSphere Business Components Composer
Bank Teller Business Components



WebSphere Buseinss Components

WSBC 製品ファミリー

- Base Components (BC)
  - クロスインダストリー向けの基本コンポーネント
- Advanced Components (AC)
  - Base Components を組み合わせた粒度の大きなコンポーネント
- Components Composer
  - 画面定義・フロー定義・ホストアプリケーション接続定義のコンポーザー、クロスインダストリー向け
  - CBTF: Banking Transaction Framework のリニューアル版
- Bank Teller
  - 銀行のテラー・アプリケーション、WebSphere Buseinss Components Composerで作成されたアプリケーション
  - Visual Banker のリニューアル版

WSBCが提供するコンポーネント BC - (V1.2.1 beta)

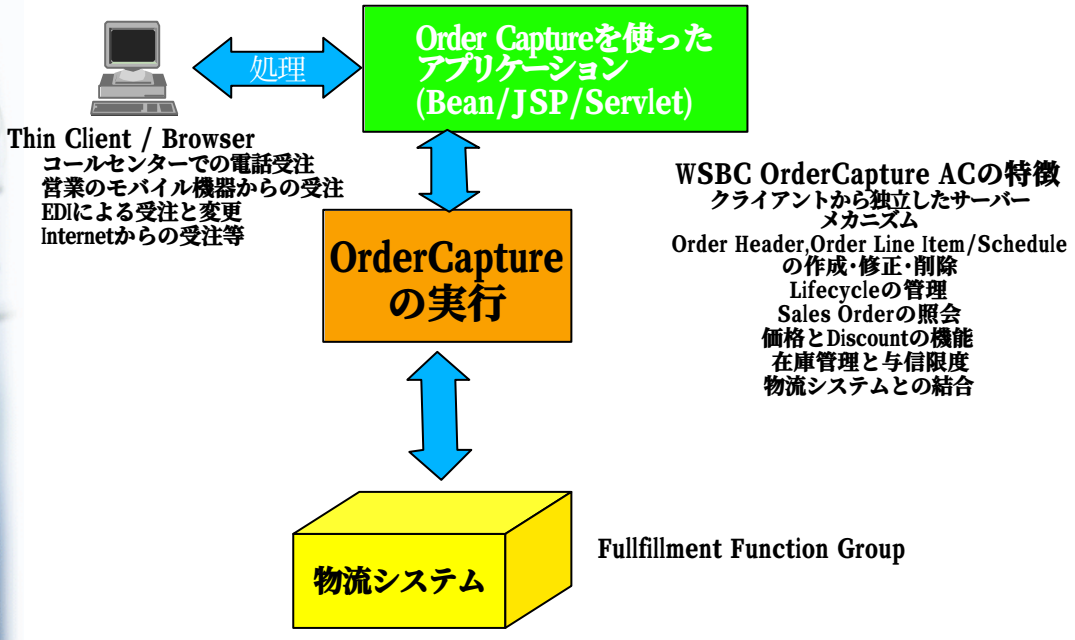
- SignatoryAbility
  - 人又はグループに対する承認レベルの定義
- ContactPoint
  - 住所、電話番号、メール・アドレス等連絡先の定義
- Party
  - 人やグループの定義
- Currency
  - 通貨と為替レートの定義
  - 通貨換算機能
- Natural Calendar
  - 労働日ベースのカレンダー
  - 労働時間算出機能
- Company
  - 企業内の組織構造の定義
- Unit of Measure
  - 計量単位とその数量
- Fiscal Calendar
  - 会計年度と会計期間
- Plan Definition
  - 活動のための計画
  - 他のアプリケーションとのやり取りの自動化などに使用
- Request
  - 他の部門に対する要求
  - 事前に定義されたプロセスに元づいて処理される
- Project Definition
  - プロジェクト
- など

WSBCが提供するコンポーネント AC - (V1.2.1)

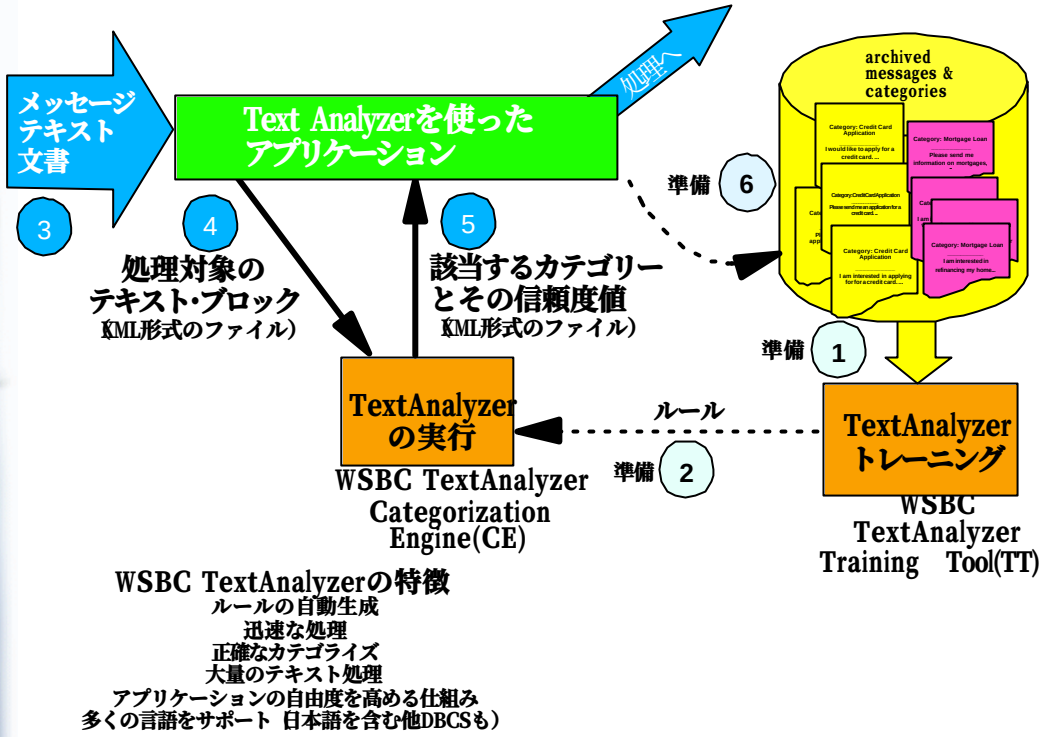
OrderCapture
- オーダー管理
- オーダー価格算出・与信限度額算出機能等
- OAGIS メッセージセットに基づく

TextAnalyzer
- 文書内のテキスト分析
- 分類機能
- 分析ルールの生成

Order Capture AC



Text Analyzer AC

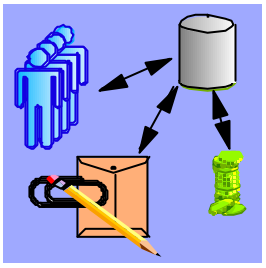


WSBC TextAnalyzerの特徴

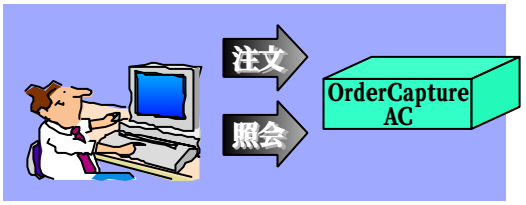
ルールの自動生成
迅速な処理
正確なカテゴライズ
大量のテキスト処理
アプリケーションの自由度を高める仕組み
多くの言語をサポート 日本語を含む他DBCSも）

2種類のコンポーネント

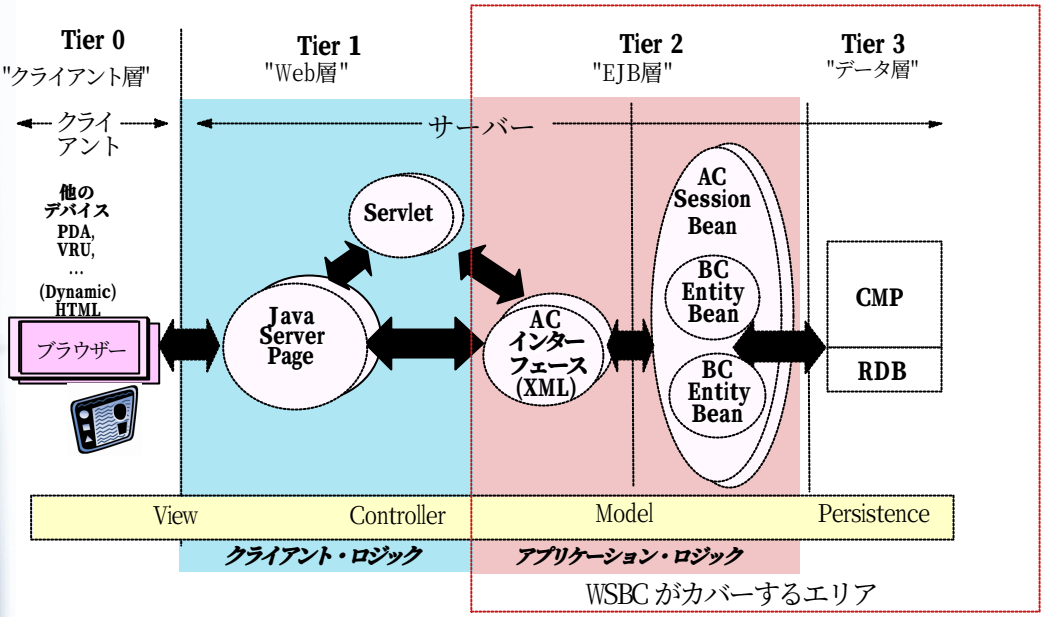
- Base Component (BC)
  - ドメイン内の個々の実体を表す
  - 相互に関係する単機能的なソフトウェア部品
  - 粒度が小さい



- Advanced Component (AC)
  - ドメイン内の機能を表す
  - それぞれが独立したソフトウェア部品
  - 粒度が大きい



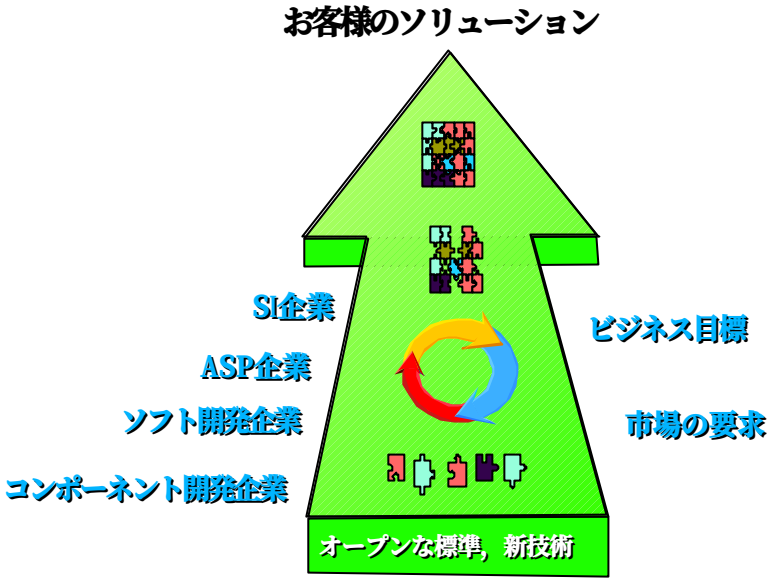
WSBC が提供するコンポーネントの範囲



目次

- 1. WSBC 概要
- 2. WSBC を使用したコンポーネント開発
  - UML ベースの開発
  - デザインパターンの適用
- 3. WSBCアーキテクチャー
- 4. まとめ

コンポーネント・ベース開発のバリュー・チェーン



様々な開発者からの要求

- 「出来合いのコンポーネントを利用して、生産性を上げコストを下げたい…」
- 「出来合いのコンポーネントだけでは機能に不満がある。自分でコンポーネントを開発して利用したい・・・」
- 「我社（SI企業）は会計アプリケーションに精通している。これを是非コンポーネント化して、様々なプロジェクトで再利用したい・・・」
- 「今後は自社資産をコンポーネント化して外販していきたい。そのためにユーザーにやさしいコンポーネントを開発したい・・・」
- 「既存資産を有効活用してコンポーネントを提供したい・・・」
- ・・・

WSBCの用途

- **コンポーネント利用者（アプリケーション・アセンブラー）**
  - 「内部の仕組みには興味がなく、手間をかけずに求める機能を手に入れたい…」
    - → ACを利用
  - 「オブジェクト指向技術を活用して柔軟性の高いシステムを開発したい…」
    - → BCを利用
    - → BCSをベースに自社のBCを作成し、それを活用
- **コンポーネント開発者（コンポーネント・プロバイダー）**
  - 「オブジェクト指向技術を活用して汎用性の高いコンポーネントを開発したい…」
    - → BCSをベースに自社のBCを作成
  - 「ユーザーにやさしいコンポーネントを提供したい・・・」
    - → BCを利用してACを作成
  - 「既存資産を有効活用してコンポーネントを提供したい・・・」
    - → ACのラッピング機能を利用して、既存システムをACに仕立てる

開発ツール

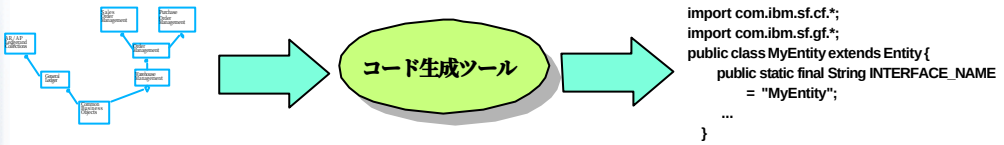
- BC 開発 (WSBC Code Generator)
  - UMLモデルからJavaコードを生成
  - AC で利用される BC を開発する際に有効
- AC 開発 (WSBC Code Generator)
  - UMLモデルからJavaコードとXMLスキーマを生成
  - スクラッチで AC を開発する際に有効
- AC アッセンブリー (AC Assembly Tool)
  - フロー・コンポジション (ノン・プログラミング)
  - AC を組み合わせてアプリケーション (または別のAC) を作成
- AC ディプロイメント (AC Deployment Tool)
  - XMLベースの AC ディプロイメント・ディスクリプターに基づき ACをディプロイ

WSBC 開発ツール
UMLに基づき Java (EJB)、XML を生成

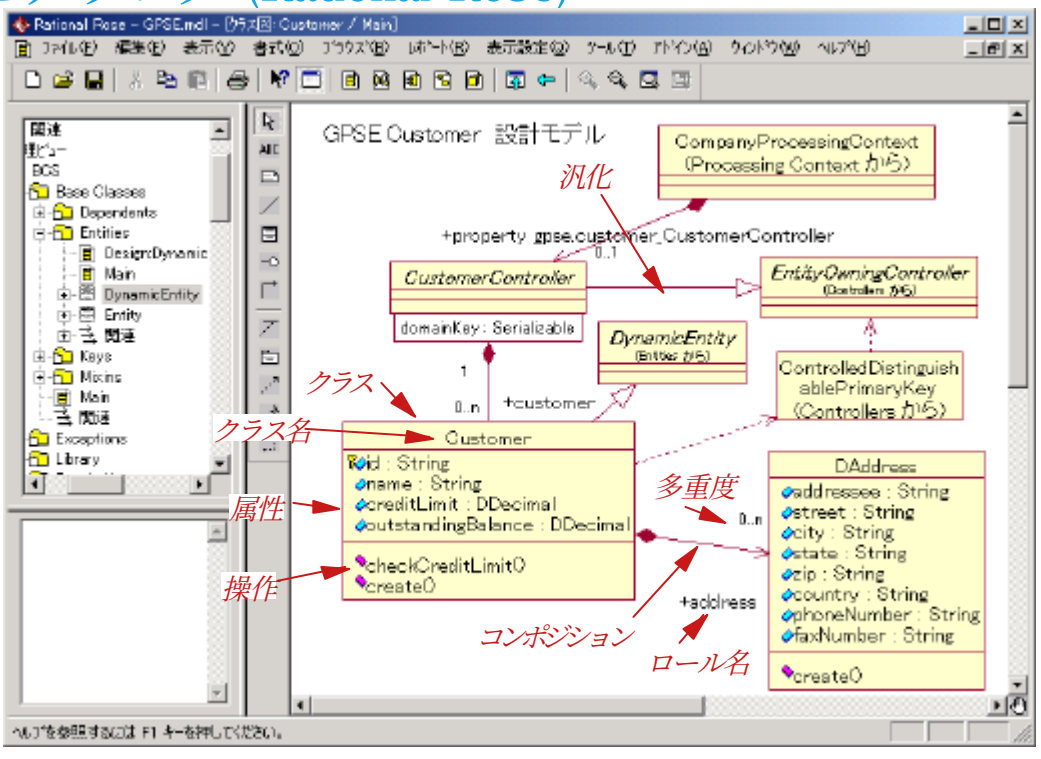
UMLモデル → 生成 → 完成

- 繰り返しのコーディング作業をなくす
- エラー発生率の減少
- システム・プログラミングから開発者を解放
- アプリケーション・ロジックに専念
- 一貫性の保持
- パワフルなベース・クラス、デザイン・パターン

生産性・品質・柔軟性の向上



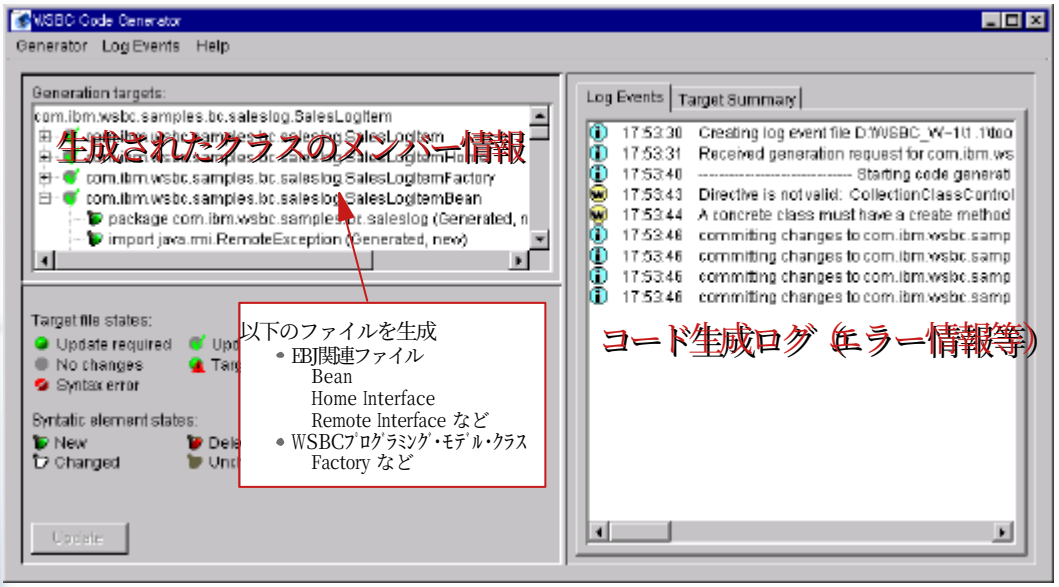
BC 開発ステップ 1
モデリング (Rational Rose)



BC 開発ステップ1 (続き)

- UML = 統一モデリング言語 (Unified Modeling Language)で表記
- クラス図をもとにコードを生成する
- WSBC が提供するコンポーネントを利用
  - Java コードレベルの再利用ではなく、デザイン・レベルでの再利用を可能にする
  - 2種類の方法
    - 汎化 (継承)による利用
    - 集約またはコンポジションによる利用
  - 例)
    - EntityOwningController : 顧客台帳や受注台帳のような「複数インスタンス」を管理する機能をあらかじめ実装している

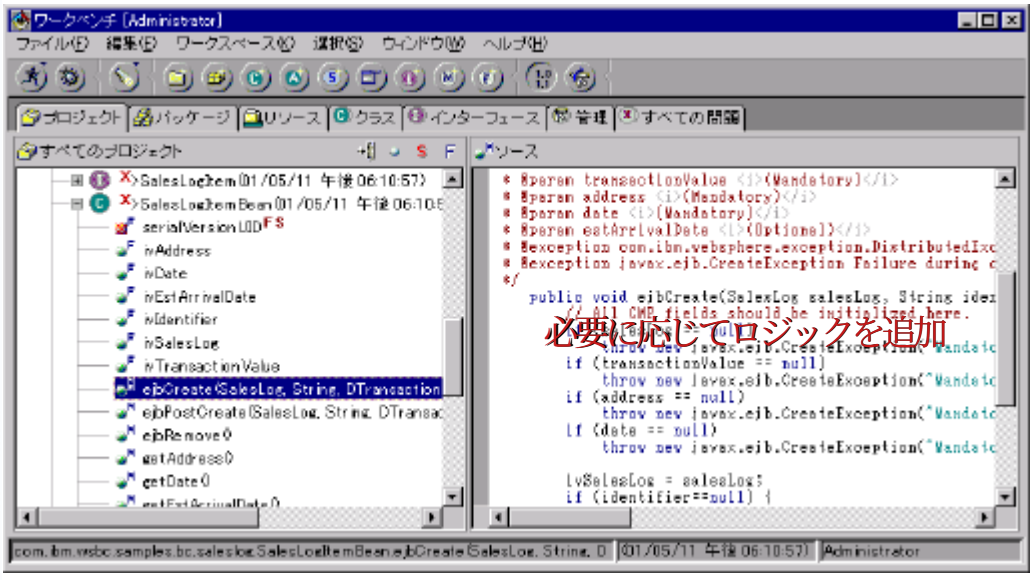
BC 開発ステップ2
コード生成 (WSBC コード・ジェネレーター)



BC 開発ステップ2　続き)

- 生成対象クラスを選択し、WSBC コード･ジェネレーター を起動
  - Rational Rose のプラグ･インとして稼動
- EJBとして稼動するのに必要なコードを生成
- Rose上に指定された指示に基づき、WSBCデザイン・パターンに則ったコードを生成

BC 開発ステップ3
コード完成 (VisualAge for Java, または WebSphere Studio Application Developer)



BC 開発ステップ3　続き)

- Roseで表現しきれないビジネス・ロジックを実装
  - 例)checkCreditLimit メソッド(与信限度額をチェックするメソッド)のロジックを実装

コンポーネントとは？

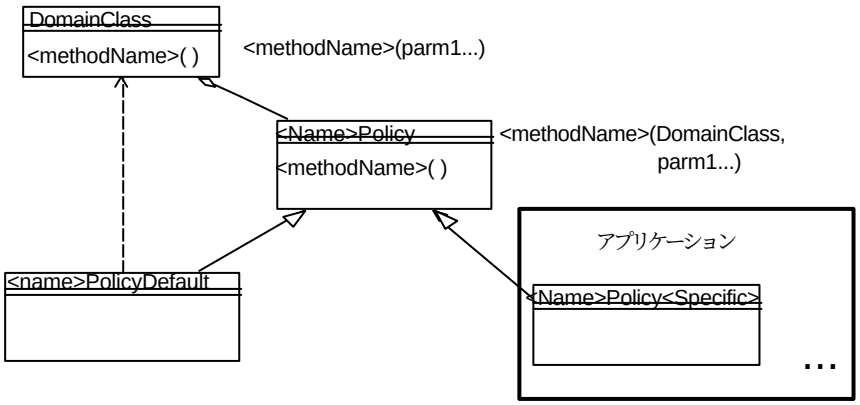
- WSBCにおけるコンポーネントの定義
- 以下の性質を持った、論理的につながりのあるソフトウェアの実装のパッケージ
  - 独立して開発・配布・導入ができる
  - 提供するサービスがインターフェイスとして明確に定義されている
  - 必要とする他のコンポーネント及びそのサービスについて明確に定義されている
  - インターフェイスとそれをサポートする実装が切り離されている
  - インターフェイスを通じてのみ操作される
  - カストマイズによって機能変更が可能

WSBC デザイン・パターン

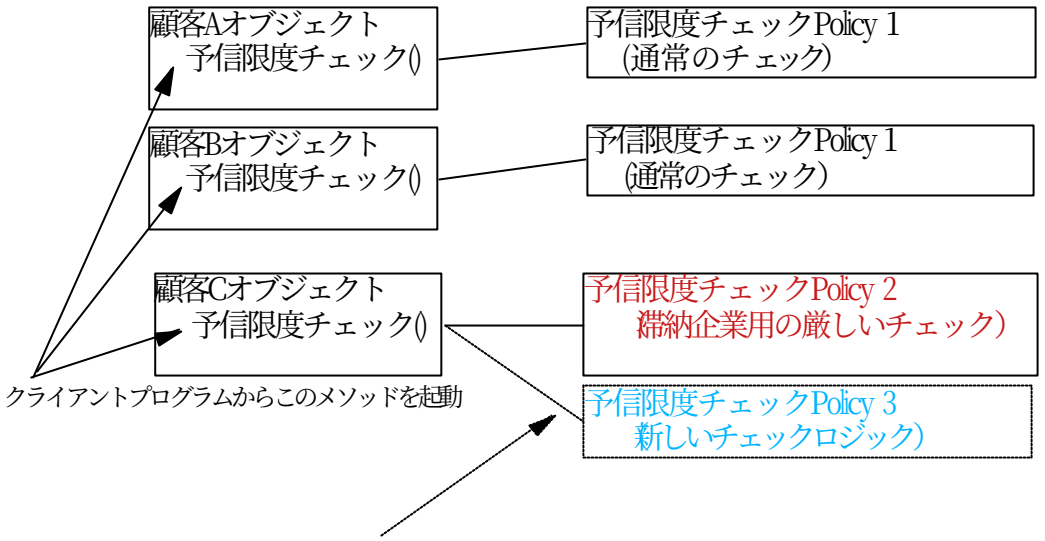
- 拡張性の高いコンポーネントを作成するためのパターン
- ビジネス・ロジックの柔軟性を高めることに主眼をおく
- UMLモデリングでパターンを指定すれば、コード生成機能が自動的にそのパターンを実装したコードを生成
  - Policy
    - 既存コードにまったく影響を与えずにロジックを切り替える
    - Strategy Pattern の応用
  - Property Container
    - クラスの再定義なしに属性を追加
    - BCにベース・クラス (DynamicEntity)として実装されてる
  - Extensible Item
    - クラスの再定義なしに継承関係を変更できる
    - 振る舞いが進化するビジネス・オブジェクトを表現
  - . . . 等

デザイン・パターンの例
Policy

- Strategyパターンに類似
  - 一群のアルゴリズムを定義し、それぞれのアルゴリズムをカプセル化し、カプセル化したアルゴリズムを交換可能にする
- オブジェクトのアルゴリズムを、オブジェクトから分離して管理することが可能
  - アルゴリズムを変更することでフレームワークを拡張
  - 状況に応じてアルゴリズムを切替え (プラグイン・プラグアウト)



Policyの適用例 顧客の予信限度チェック)

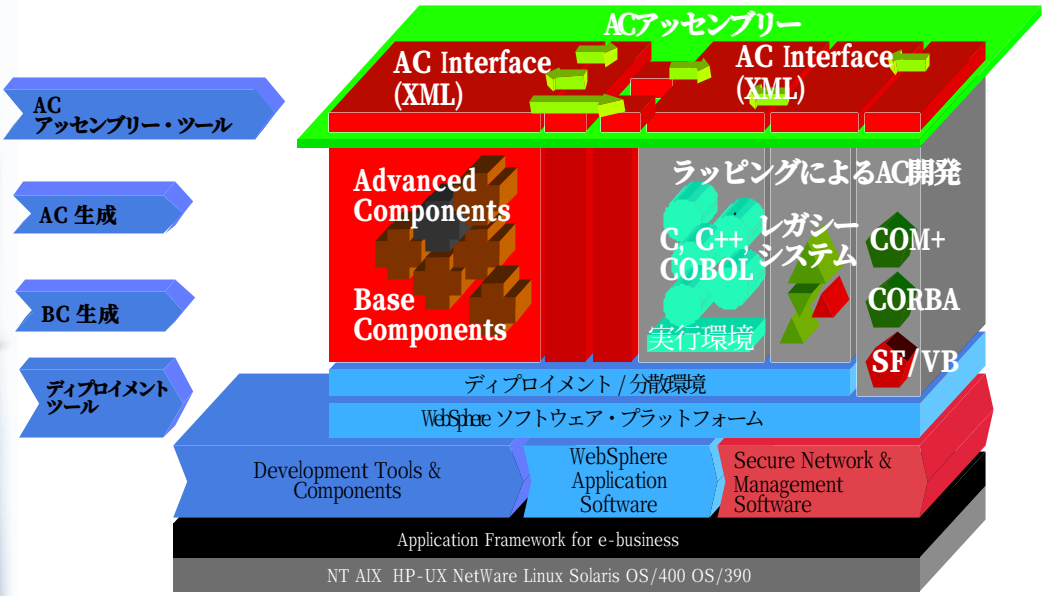


新しいチェックロジックが必要になれば、そのロジックを実装したクラスを作成し、対象顧客オブジェクトと紐付けるだけですむ。既存のコードに影響なし 容易な拡張)
従来の手法だと、既存プログラムを改造し、 If文を追加して、そこにロジックを埋め込む)既存データへの影響がないかをテストしなければならない。

目次

- 1. WSBC 概要
- 2. WSBC を使用したコンポーネント開発
  - UML ベースの開発
  - デザインパターンの適用
- 3. WSBCアーキテクチャー
- 4. まとめ

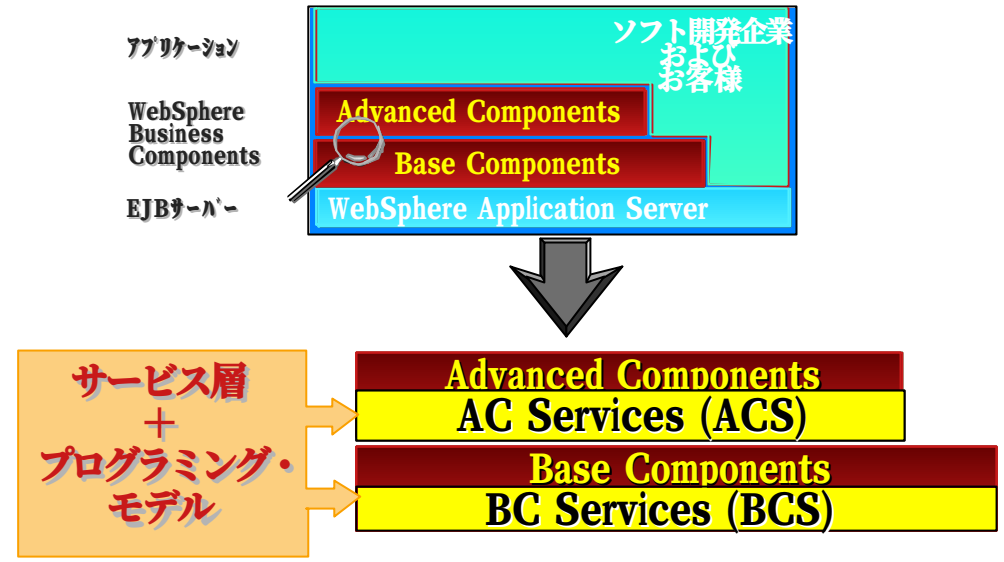
WSBC アーキテクチャー 全体図



BCとAC ～比較～

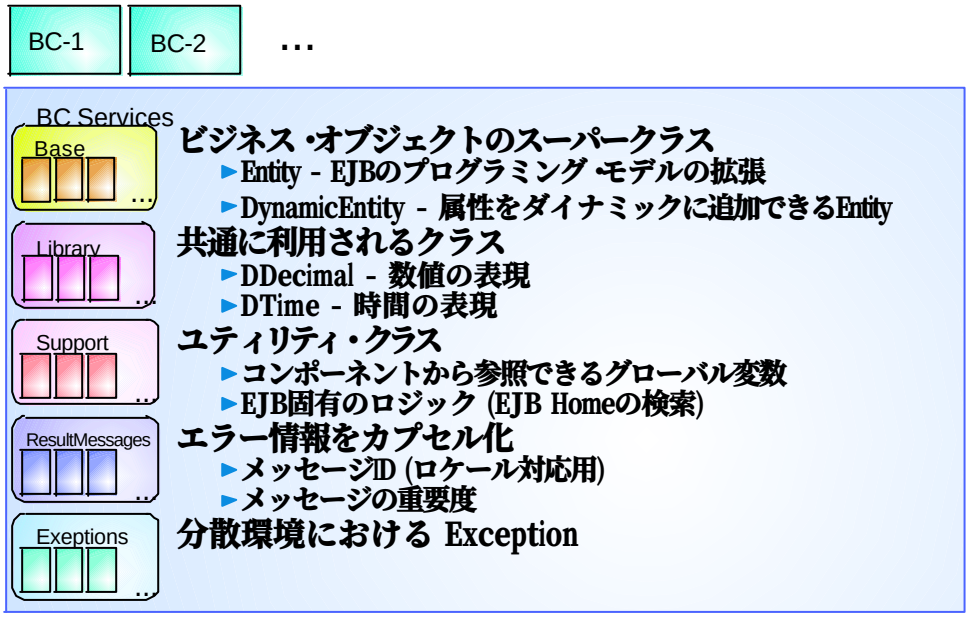
	BC	AC
表現しているもの	実体	機能
例	製品、顧客、など	受注管理
利用形態	EJBプロトコルによる会話	XMLによる会話が可能
難易度	プログラミング・モデルがやや複雑	プログラミング・モデルが容易
カスタマイズ	ホワイト・ボックス(ソース公開) ■ 伝統的なオブジェクト指向による拡張 (継承・集約・委譲) ■ 豊富な拡張メカニズム Policy , ExtensibleItem, PropertyContainer, など)	ブラック・ボックス ■ 構成設定機能による外側でのカスタマイズ ■ Policyメカニズムを用いたロジック置換
実装	1つ以上の CMP Entity Bean	インターフェイス＝1つ以上の Stateless Session Bean 実装＝あらゆる実装が可能 (BC、EJBのEntity Bean、COM/CORBAアプリケーション、レガシー・アプリケーション)

コンポーネント・サービス



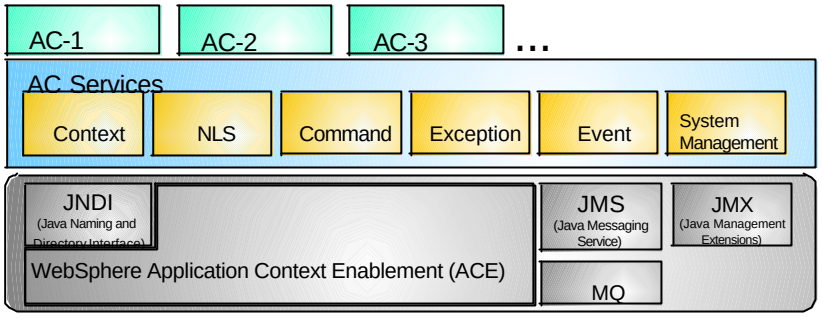
BC Services (BCS)

BCの開発に必要な基本機能を提供



AC Services (ACS)

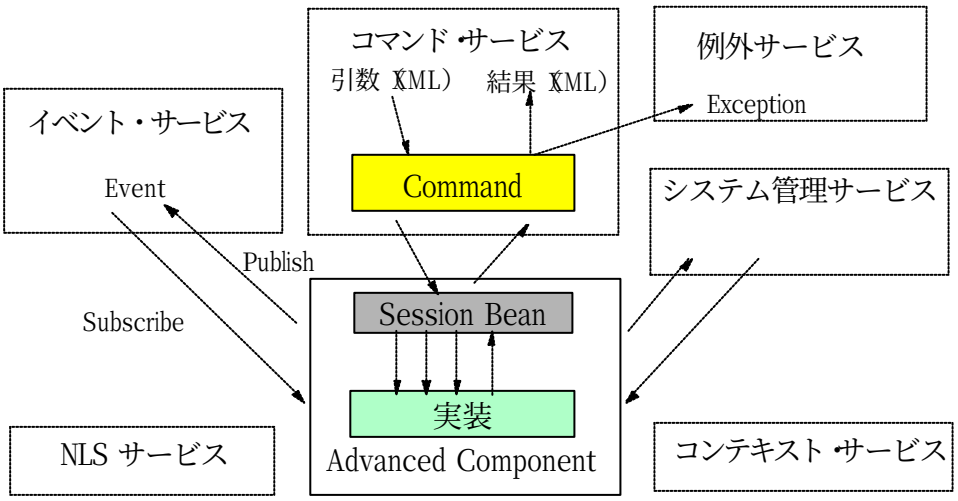
ACの開発・実行に必要な基本機能・サービスを提供



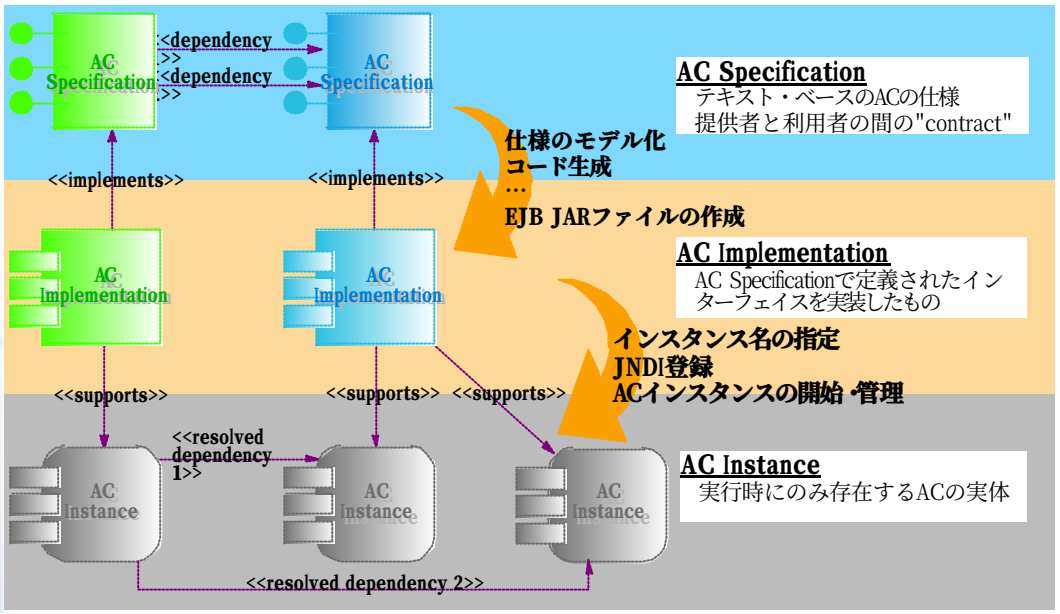
- **AC コマンド・サービス**
  - XMLインターフェースを利用してACを使用するためのサービス
- **AC イベント・サービス**
  - 分散環境での通知サービス
  - 疎結合： コンポーネント間の依存性の低いPublishとSubcribe
- **AC 例外サービス**
  - 分散環境での例外サービス
    - クライアント・サーバー間のスタック・トレース
    - クライアント・サーバー間の例外チェーン
- **AC NLSサービス**
  - 分散環境でのNLS(各国言語対応)サービス
- **AC コンテキスト・サービス**
  - ACのインスタンスに情報を付加
  - JNDIコンテキスト・インターフェースを使用
- **AC システム管理サービス**
  - ACを外からコントロール
    - 開始・終了、バッチ・スケジューリング、トレースのOn/Off、など
  - ACの実装方法を隠蔽

Advanced Component Services (ACS)

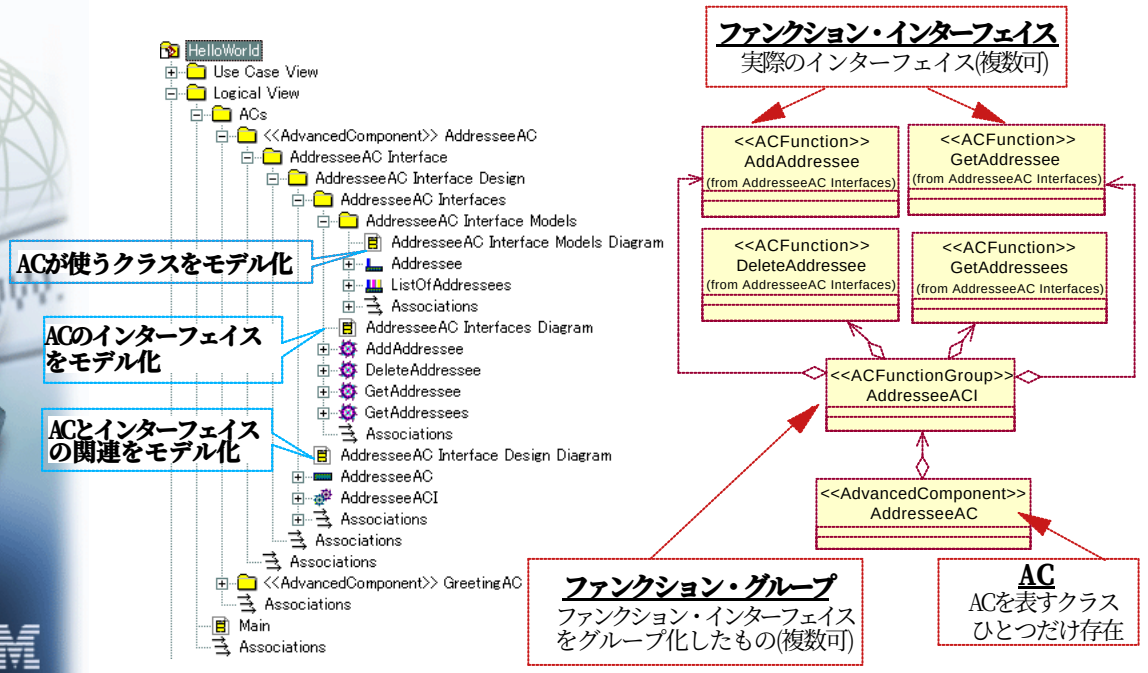
続き



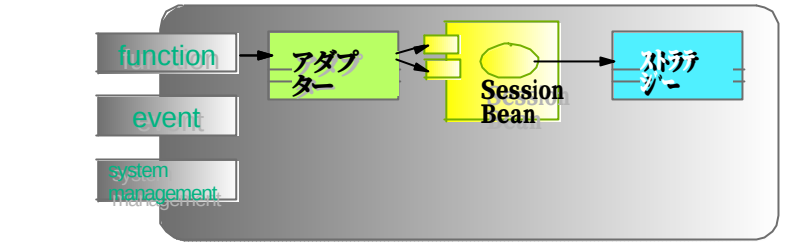
参考）ACの開発・ディプロイ概略

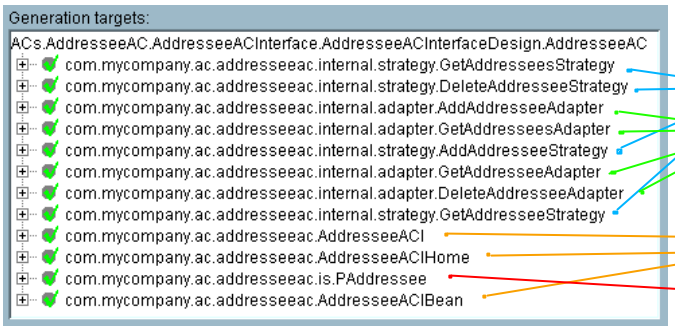


参考）AC 開発ステップ 1
モデリング (Rational Rose)



参考）AC 開発ステップ２
コード生成 (WSBC コード・ジェネレーター)





参考）AC 開発ステップ３
XML生成(WSBC コード・ジェネレーター)

ACDefinition.xml

ACそのものの定義 依存関係、インターフェイスの種類、etc.

```
<AdvancedComponent .... >
<name>AddresseeAC</name>
<specVersion>1.1.0</specVersion>
<interface>
 <name>GetAddressees</name>
 <requestSchema>.../AddresseeAC/Functions.xsd</requestSchema>
 <responseSchema>.../AddresseeAC/Functions.xsd</responseSchema>
</interface>
<interface>
 <name>GetAddressee</name>
 <requestSchema>.../AddresseeAC/Functions.xsd</requestSchema>
 <responseSchema>.../AddresseeAC/Functions.xsd</responseSchema>
</interface>
.....
</AdvancedComponent>
```

ACImplementation.xml

ACの実装・構成情報の定義

```
<AdvancedComponentImplementation....>
 <specVersion>1.1.0</specVersion>
 <specSchema></specSchema>
 <name>AddresseeAC</name>
 <implVersion>1.0.0</implVersion>
 <loadBalancable>true</loadBalancable>
 <interfaceImpl>
  <name>GetAddressees</name>
  <adapter>com.mycompany.ac.addresseeac.internal.adapter.GetAddresseesAdapter</adapter>
 </interfaceImpl>
 <interfaceImpl>
  <name>GetAddressee</name>
  <adapter>com.mycompany.ac.addresseeac.internal.adapter.GetAddresseeAdapter</adapter>
 </interfaceImpl>
 ....
</AdvancedComponentImplementation>
```

ACFunctions.xsd

ファンクション・インターフェイスの文法を定義

```
<xsd:element name="AddAddresseeRequest">
 <xsd:complexType>
  <xsd:element name="addressee" type="ac:PAddressee"/>
 </xsd:complexType>
</xsd:element>
<xsd:element name="AddAddresseeResponse">
 <xsd:complexType>
  <xsd:element name="addressee" type="ac:PAddressee"/>
 </xsd:complexType>
</xsd:element>

<xsd:element name="GetAddress
 <xsd:complexType>
  <xsd:element name="identifier"
 </xsd:complexType>
</xsd:element>
<xsd:element name="GetAddresseeResponse">
 <xsd:complexType>
  <xsd:element name="addressee" type="ac:PAddressee"/>
 </xsd:complexType>
</xsd:element>
```

ACInterfaceModels.xsd

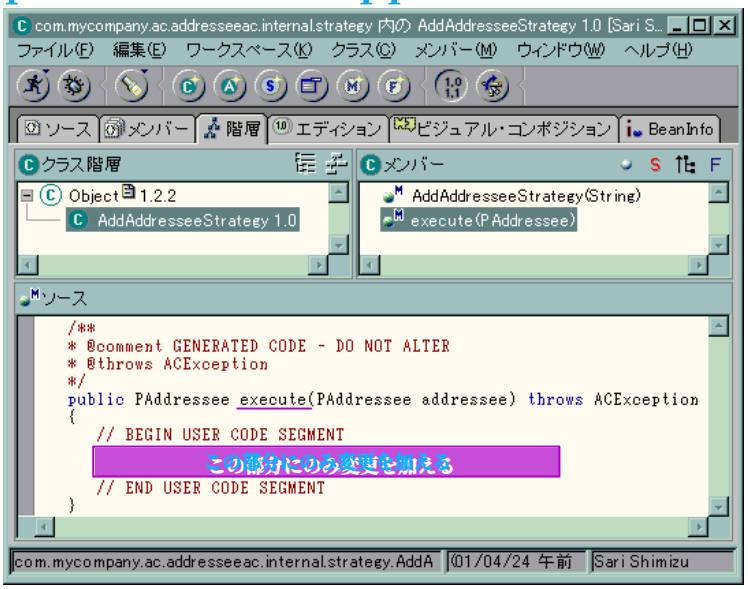
AC Interface Model の定義

```
<xsd:complexType name="PAddressee">
 <xsd:element name="name" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 <xsd:element name="address" type="xsd:string" minOccurs="0" maxOccurs="1"/>
 <xsd:element name="identifier" type="xsd:string" minOccurs="0" maxOccurs="1"/>
</xsd:complexType>
```

ACEvents.xsd

イベント・インターフェイスの文法を定義

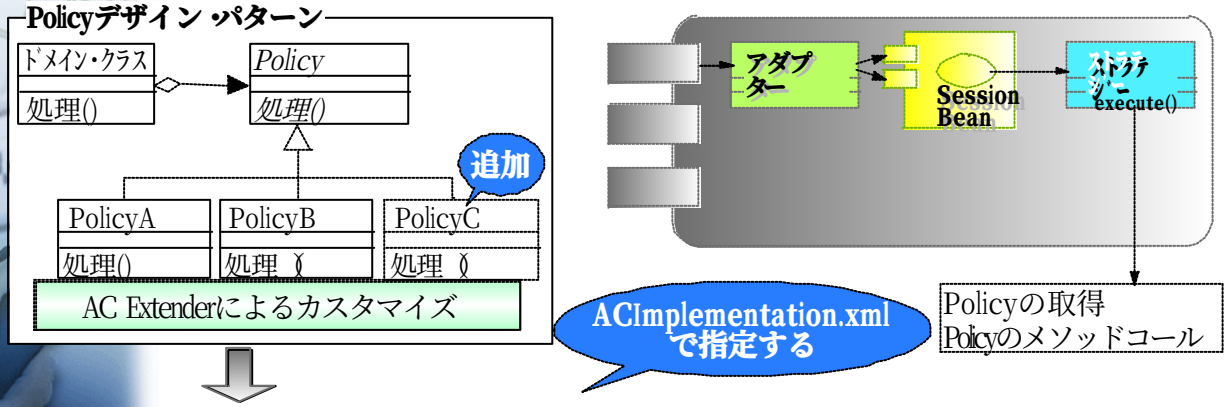
参考）AC 開発ステップ4
コード完成 (VisualAge for Java, または WebSphere Studio Application Developer)



- ビジネス・ロジックはXxxStrategyのexecute()に実装
- Code Generatorが生成したコードで編集の対象になるもの
  - XxxStrategyのexecute()メソッド --> ビジネス・ロジックの実装
  - ACImplementation.xml --> システム管理対応の構成用タグを追加

参考）AC 開発ステップ 参考）
Policy デザイン・パターンによるロジックのカスタマイズ

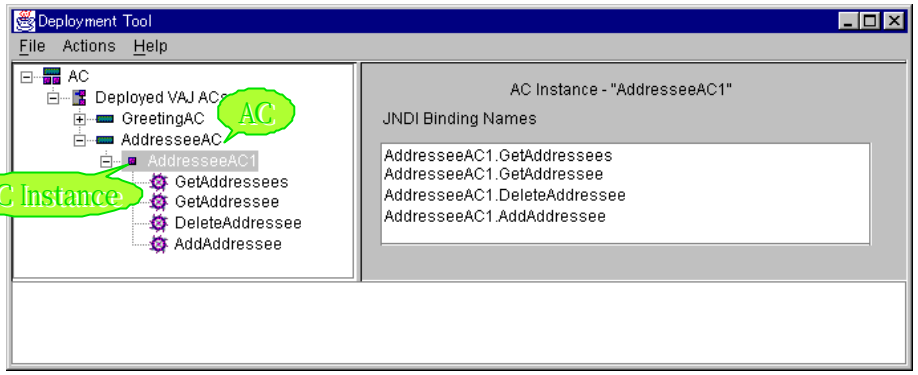
- Policy デザイン・パターン
  - If文やSwitch文を使用せずにロジックを切り替えられる仕組み



アプリケーション・アッセンブラーには、パラメータとして提示される
ACの内部コードを修正する必要はまったく無い）

参考）AC ディプロイメント
ACの配置 (AC ディプロイメント・ツール)

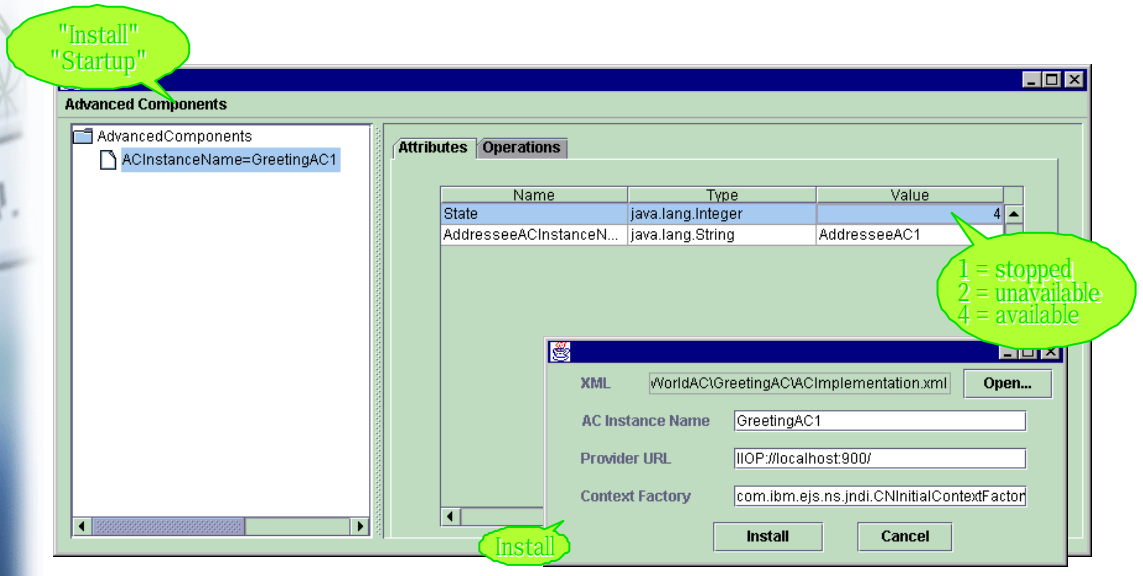
- ディプロイメント・ツールによるACインスタンスの作成



★ ACのEJB JARファイルからACの情報をロード
- ロードした時点では"Undeployed ACs"
  - ➡ AC Instance名、ACCommandTargetのJNDI名、アダプター名をセットして "Deploy"

参考）AC ディプロイメント
AC の開始 (システム管理コンソール)

ACの実装方法を隠蔽した管理が可能



目次

- 1. WSBC 概要
- 2. WSBC を使用したコンポーネント開発
  - UML ベースの開発
  - デザインパターンの適用
- 3. WSBCアーキテクチャー
- 4. まとめ

まとめ

- WSBCは単に出来合いのコンポーネントのみを提供するだけではありません。
- WSBCはUMLモデルから新規コンポーネントを作成するための強力なツールを提供します。
  - UML をベースにしたコード生成機能 (WSBC Code Generator)
  - SanFranciscoを利用しているお客様は、既にこれと同等な機能により、大きなメリットを得ていらっしゃいます。
- WSBCはEJB以外のテクノロジーでコンポーネントを実装することを可能にします。
  - AC のラッピング機能
  - ACSのシステム管理サービスによる一元管理
  - ACSのコマンド・サービスによるXMLインターフェース
- WSBCはコンポーネント・ベース開発を行うための強力な開発基盤と実行基盤を提供します。
  - BC Services
  - AC Services

関連サイト

- 英語 :WebSphere Business Components
  http://www.ibm.com/software/components
- 英語 :WebSphere Developer Domain
  WebSphere Business Componentsゾーン
  http://www7b.boulder.ibm.com/wsdd/zones/wsbc/
- 評価版BC(Beta版)のダウンロード
  http://www.ibm.com/software/webservers/components/download.html

問い合わせ先

- 日本アイ・ビー・エム株式会社 ソフトウェア事業部
  e-ビジネス ソフトウェア営業推進
  星 野　寧 夫
  - e-mail　:hoshino y@jp.ibm.com
  - 電話　:03-3808-5541