WebSphere software

IBM

e buziness software

WSAD IE 5.0 and WebSphere 5.0 Beta

Introduction

IBM Software Group

Welcome
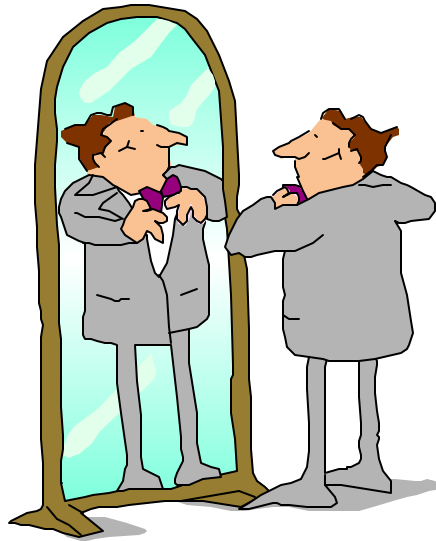
Course Overview

- Introduction to WSAD IE and WAS Enterprise
- Web Services
- Workflow
    - ► Concepts
    - ► Runtime Perspective
    - ► Tool Perspective
    - ► labs
- Extended Messaging
    - ► Concepts
    - ► Runtime Perspective
    - ► Tool Perspective
    - ► labs
- Programming Model Extensions
    - ► Concepts
    - ► labs
- Getting Started

Introduction to WSAD IE and WAS Enterprise

- A New World of e-Business
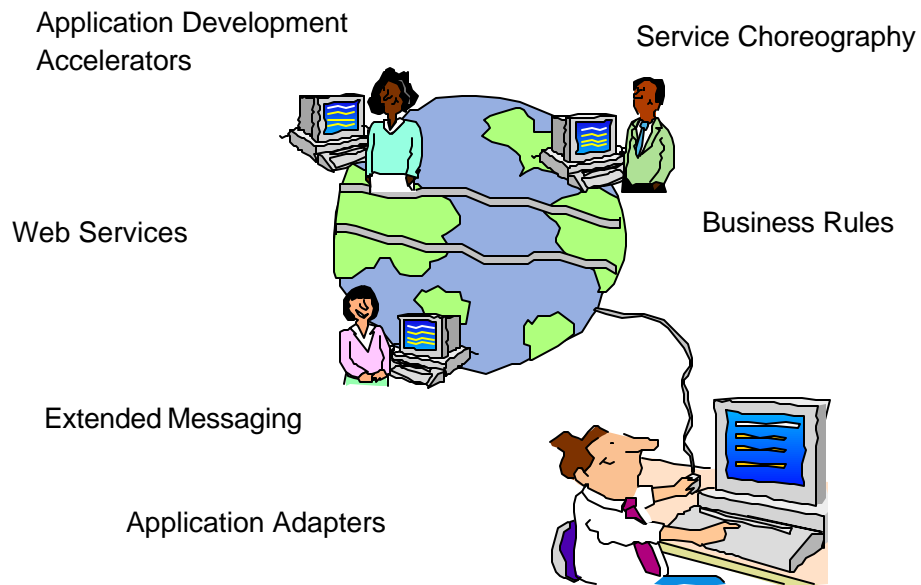- WebSphere Enterprise 5.0 Features
- WebSphere Service Choreography
- Programming Model Extensions
- Workflow Programming Model
- High level architectural components
- WSAD IE and WAS EE
  - ► Tools vs Runtime
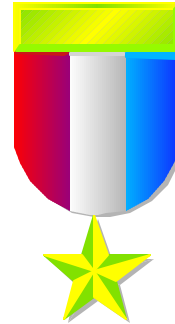
A New World of e-Business with WebSphere Enterprise

Application Development
Accelerators

Service Choreography

Web Services

Business Rules

Extended Messaging

Application Adapters

WebSphere Enterprise 5.0 features

- Service Choreographer
  - ► Flow Choreographer
  - ► Compensation Services
  - ► Human Interaction facilities
- Advanced Transactional Connectivity
  - ► Web Services model
- Extended J2EE Development Accelerators
  - ► Asynch Beans
  - ► Extended Messaging (CMM)
  - ► Internationalization (i18n)
  - ► Work Area
  - ► Application Profiling (Dynamic Access Intent)

Features (continued)

- Dynamic Application Adapter Support
  - ► Activity Session
  - ► Last Participant Support
- Dynamic Application Support
  - ► Dynamic Query Support
  - ► Business Rule Beans
- Single Unified Environment (tools and runtime)

WebSphere Service Choreography

- What is it?
  - ▶ the next step in the Web Services Revolution
- Web Services bring us a web-oriented model that lets us provide key pieces of business functionality to web clients using a standard interface and protocols.
  - ▶ a black box with well defined inputs, outputs and behavior.
- A business process, is the way we use this functionality,
  - ▶ the order in which we call our business functions
  - ▶ the rules that govern our decisions about how and when to use these functions.
  - ▶ inherently involves interactions with people and software systems.
- **Service Choreography**:
  - ▶ the composition and arrangement of business services

a Business Process (flow)

Service
binding

Service
binding

Service
binding

Service
binding

a Business Service
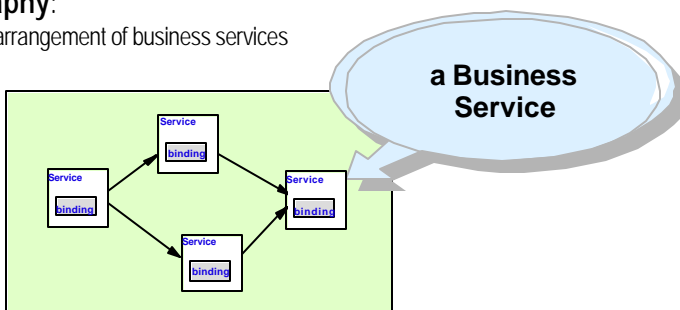
▶ This is the very high end of the Application Development Accelerators.

Business Process Modeling (definitions)

Business Process Modeling (BPM) is the automation and facilitation of _business interactions_ both within and outside a company.

Business Operations: the steps that are necessary to complete the tasks related to running the business.

Business Services: choreograph the interaction of these business operations with the help of a Service Broker that _manages the flow_ of messages between them.

This process can be further automated with the use of;

Flow models: dictate the flow of information and control _between_ and _within_ services.
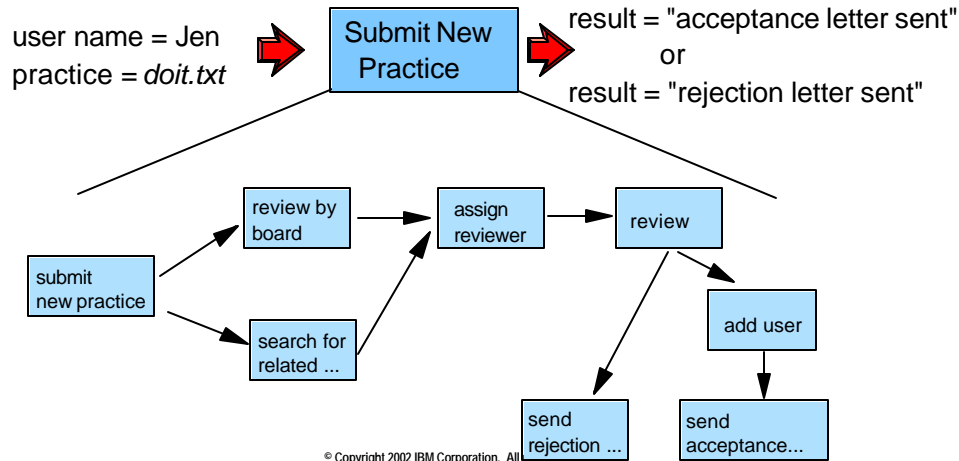
- ► Emphasis on the Process in "BPM" e.g. interactions.

- ► This really starts with the corporate information model for the corporation

- ► The Service Broker is the underlying workflow engine which is part of the WAS Enterprise 5.0 runtime.

Example of a Business Process Flow

Business Operation: Review a "Best Practice" for publication on the "Best Practices" website (portal)

Description: A person submits a **WebSphere Best Practice** idea which is reviewed by a member of the "Best Practices" review board and accepted or rejected.

user name = Jen
practice = *doit.txt*

➡ **Submit New Practice** ➡

result = "acceptance letter sent"
or
result = "rejection letter sent"

submit new practice → review by board → assign reviewer → review

submit new practice → search for related ...

search for related ... → assign reviewer

review → send rejection ...

review → add user

add user → send acceptance...

▸ This is a long running flow that chains several business processes together, also known as macroflow.

▸ You will see shortly that the 'add user' activity is what we call a microflow.

WebSphere Service Choreography (continued)

- Brings Workflow capabilities to WebSphere
- Invocation of services
  - ▶ J2EE components: EJBs, Connectors, Java beans, JMS, MDB
  - ▶ Web Services
- Tight integration into WebSphere w.r.t.
  - ▶ Configuration
  - ▶ Administration
  - ▶ Deployment and installation of applications
- Deliver universal, OEM-able flow engine
  - ▶ Focus on flow (key: performance)
  - ▶ Pluggable and extensible via plugins
- Support for human interactions with the workflow
- Support for interruptible flows that last over time
  - ▶ fully forward recoverable
  - ▶ facilities for error handling and undoing committed operations.

- ▶ EAB - Enterprise Access Builder
- ▶ MQAO - MQ Adapter Offering
- ▶ MQWF - MQ Work Flow
- ▶ LWF - Lotus Work Flow
- ▶ JMS - Java Messaging Service
- ▶ MDB - Message Driven Bean

Creating Services

- This is where the other WebSphere Enterprise 5.0 features come into play.
- You have a reusable business service that you will want to use in several different scenarios.
  - ► How do you implement this service?
    - EJBs, Java Class, CICS, IMS, a combination these?
  - ► How do you optimize a given implementation for a particular application access pattern?
  - ► How you build it quickly and consistently?
  - ► How you provide alternate implementation (or migrate from a old to a newer imp)?

Programming Model Extensions

- Application Profiling: Access Intent
  - ► EJB performance optimizations
  - ► Application Development Accelerator
- Extended Messaging
  - ► Container managed
    - hide complexity of underlying messaging system
    - Generate and use common application programming patterns for messaging
    - Application Development Accelerator
- Dynamic Query
  - ► EJB optimizations
  - ► Application Development Accelerator
- Last Participant Support
  - ► Advanced Transactional support
  - ► adding a 1 phase resource to a 2-phase commit transaction

► There will be more on these topics later in the class.

► The important thing to note here is that when we actually development an implementation that is going to fit into an Enterprise Application we start to discover that there are many impediments due to existing systems and architectures.

► The programming model extensions are features that IBM has developed to overcome these impediments.

► They are ways to
  ► improve performance
  ► extend J2EE applications at deployment time rather than development time
  ► meet the diverse transactional needs of existing systems
  ► build and evolve applications smoother and quicker.

Example: EJB Dilemma

- A mailing list application needs to use a customer EJB to get the names and addresses of all the customers that bought something within the last 3 months.
  - ► The application does not update the customer information but only reads it.
- How do we tell "the system", the EJB Container, that we only need to use this in a **read-only** fashion?
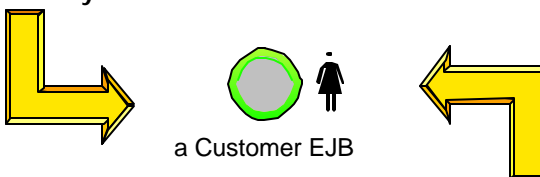
a Customer EJB

- Another application wants to use the same customer EJB, except this application needs to **update** the customer information?
- What are our options at this point?

► What happens when you're using your EJB as the implementation of a service in a Business Process Flow?

► Which access pattern do you optimize for?
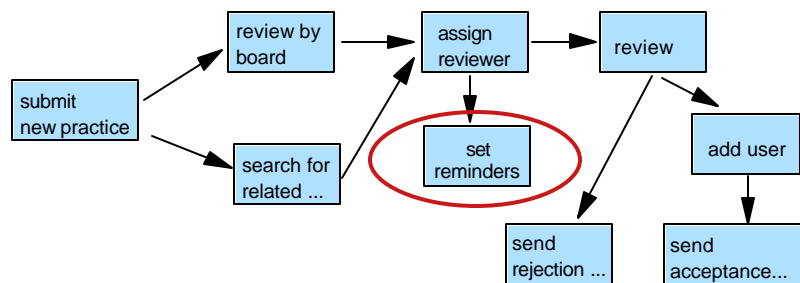
Programming Model Extensions (continued)

- Business Rule Beans
- Activity Session
- Asynchronous Beans
  - ▶ multi-threaded programming and thread pool support
- Work Area
  - ▶ a "*user managed*" application context for sharing application data in a distributed application
  - ▶ Application Development Accelerator
- Internationalization
- Application Adapter Support
  - ▶ Generate Web Service interfaces for JCA adapters
    - CICS, IMS, ...

▶ The programming model extensions (PME) can be used without using workflow but together they are a powerful combination.
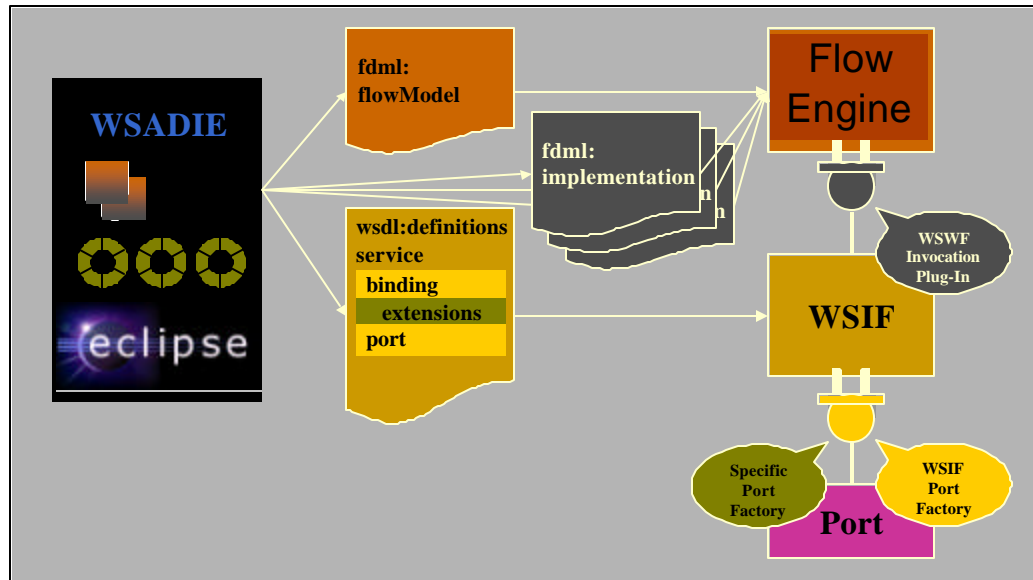
Example: Scheduler and Business Process Flow

- In the "WebSphere Best Practices" business process flow, the reviewers must respond within 30 days.
- The process now has to **periodically** check to see what submissions are outstanding for each reviewer and send them notifications as the deadline approaches.
- This is a perfect opportunity to add in a new service that uses the scheduler to implement the periodic checks and send the notifications.

```
submit          review by        assign          review
new practice    board            reviewer
                                                  add user
                search for       set
                related ...      reminders
                                 send             send
                                 rejection ...    acceptance...
```
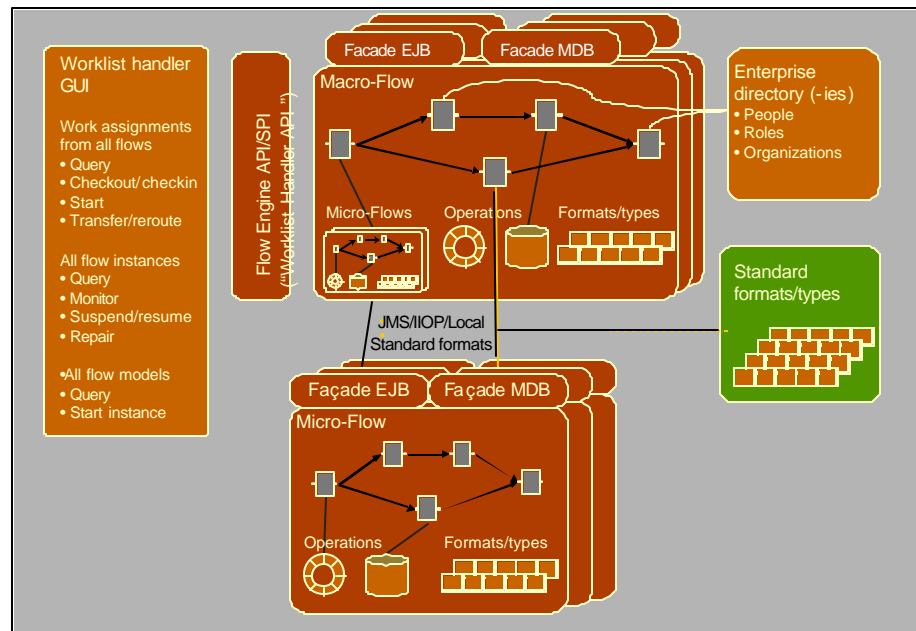
▸ The key here is that there is a task that needs to be done periodically.

▸ Note that the Business Process can be easily extended by adding in a new service (Activity Node)

WSADIE, Workflow and WSIF

Workflow Programming Model

Worklist handler GUI

Work assignments from all flows
• Query
• Checkout/checkin
• Start
• Transfer/reroute

All flow instances
• Query
• Monitor
• Suspend/resume
• Repair

•All flow models
• Query
• Start instance

Flow Engine API/SPI ("Worklist Handler API")

Facade EJB   Facade MDB

Macro-Flow

Micro-Flows   Operations   Formats/types

Enterprise directory (-ies)
• People
• Roles
• Organizations

Standard formats/types

JMS/IIOP/Local
Standard formats

Façade EJB   Façade MDB

Micro-Flow

Operations   Formats/types

- ► Asynchronous programming model is supported with the Facade MDB
- ► Synchronous programming model is supported with the EJB facdae
- ► Interruptable flows are supported with Events and Staff (human interactions)

Plug-in Domains

- Data handling – dealing with data involved in flows
  - ▶ Access, conditions, mapping
  - ▶ Examples: Java object, XML document
  - ▶ Examples: WebService, Java, EJB, J2EE connecto
- Staff – all operations involving people
  - ▶ Authentication, authorization, resolution
  - ▶ Examples: WebSphere user registry, LDAP
- State Observer – notifications about state transition in flow or activity
  - ▶ Examples: Audit trail

Flow Container: Internal Structure
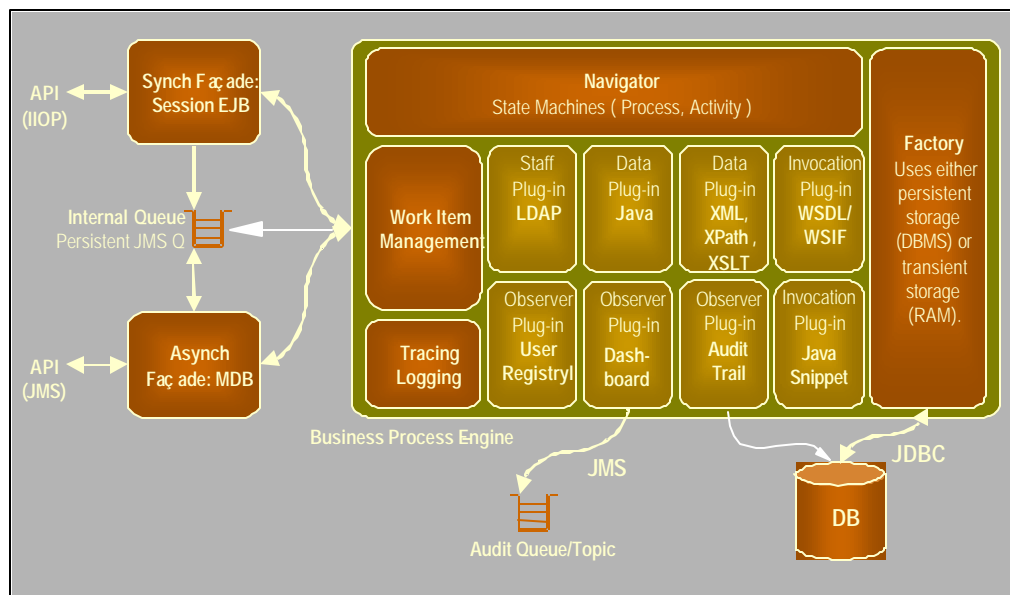
- ► Remember we are still in the Beta. Everything is subject to change.
- ► The name Flow Container may change to something like Business Process Container
- ► The Dashboard plugin will not be available for a while.
- ► The intent of this picture is to show you that we have an architecture that is designed with extensibility in mind.

External Interfaces

- Work item handler interfaces:
  - Work item management for people-based flows (query, transfer)
  - Flow instance management (call, spawn, query, sendEvent, suspend, resume, terminate, monitor, repair)
  - Activity management (claim, complete, repair)
  - Default Web Client for simple testing and verification of flows.
- API renderings:
  - Synchronous via IIOP: Stateless session EJB
  - Asynchronous via JMS: Request / reply messages using XML format sent to Business Process Engine's JMS queue
- Choreography Web Client Framework
  - model-view-controller framework that allows you to quickly define simple users interfaces using jsps.

- The Choreography Web Client Framework will not be available during the beta timeframe.

External Interfaces

- aFlow.Call – synchronous invocation of a flow
  - ► Creates and starts a flow instance
  - ► Returns when flow instance is complete
- aFlow.Initiate – asynchronous invocation of a flow
  - ► Creates and starts a flow instance
  - ► Returns immediately after successful instance creation
- Query – learn about assigned work
  - ► Query assignments
  - ► Query flows
- Activity.Claim – person begins working with an activity
  - ► Get access to activity to work with it, access to input data
- Activity.Complete – person is done with an activity
  - ► Complete work on activity, pass result data

► More on this when we talk about the runtime.

Tools vs. Runtime

- **WSAD IE**
  - ► Create Business Process Flows
  - ► Create Container Managed
    - – Sender/Receiver Beans
 - ► Web Service interfaces and wrappers
  - Generate code
  - Package Applications
  - Unit Test Environment
  - WSAD IE help

- **WAS Enterprise**
  - ► Admin Console
    - – deploy and manage flows
      - • flow container
    - – deploy and manage Container Managed Messaging Beans
    - – deploy and manage J2EE applications
  - ► Default Thin Client
  - ► Choreography Web Client Framework
  - ► Programming Model Extensions
  - ► Infocenter
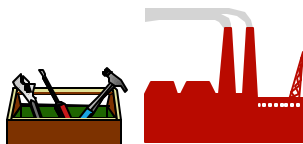
► It is important to keep in mind where you are in the development process and how the 2 products are delivered.

► There are some tools for working with flows that are delivered with the runtime.

► There are also samples delivered with the runtime as well as samples that come as part of WSAD IE.

► There may be some information in the Infocenter regarding runtime components that don't require tooling, which you won't find in the WSAD IE help and vicea versa. I look through both.

Summary

- With WebSphere Enterprise 5.0 and WebSphere Studio Application Developer Integration Edition 5.0, you now have the tools and runtime to take you into the next level of e-Business and Enterprise computing . . .

  - ► Application Development Accelerators
  - ► Rule Based Programming
  - ► Entended Messaging
  - ► Application Adapters

- all tied together using <u>Web Services</u> with a <u>work flow programming model</u>

► A loosely coupled but integrated system.