



Il Mondo dei Partner
INNOVARE E CRESCERE. INSIEME **2006**
Milano 19 - 20 Ottobre ***Technical World***

ACG Service Bus
Converter, Connector e web services
Alessandro De Benedittis



Agenda

Service Oriented Architecture

- **Overview SOA**
- SOA Technologies
- Web services

ACG Service Bus

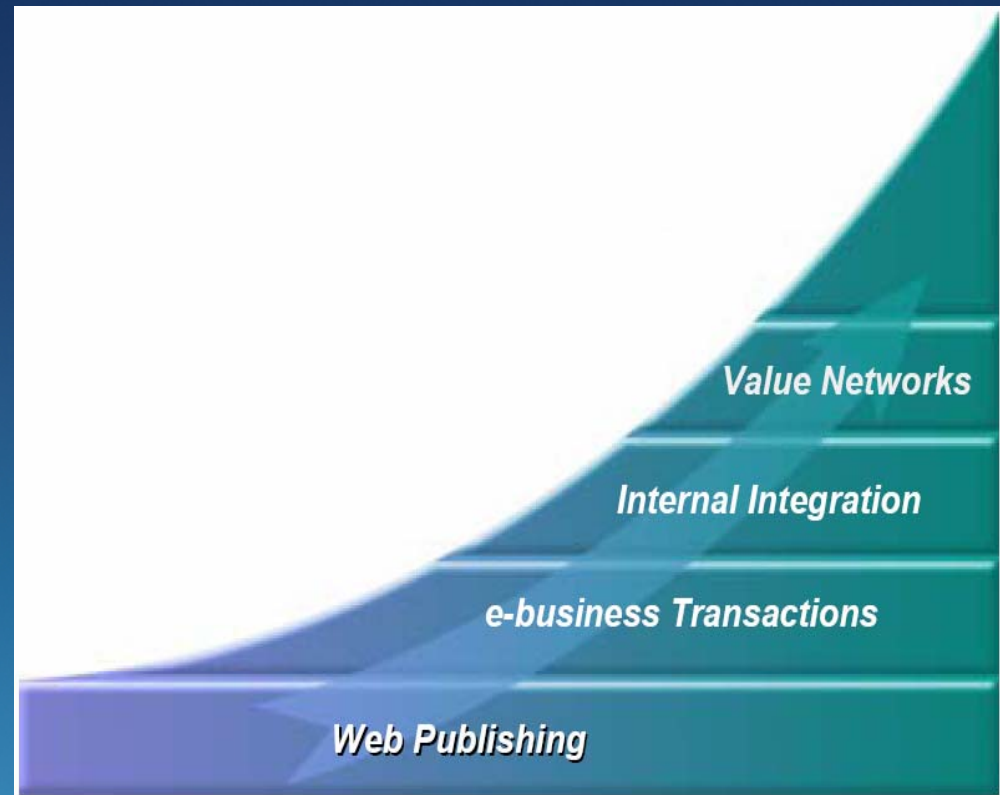
- Overview
- Programming model
- Development tools

L'evoluzione dell'e-business

Attualmente, si assiste ad una crescita esponenziale di:

- ✓ Terminali
- ✓ Dati
- ✓ Transazioni

in una rete che riconosce l'**integrazione** come vero valore dell'e-business



L'evoluzione dell'e-business

Le PMI devono evolvere i loro processi aziendali in modo che siano:

- ❖ integrati end-to-end, sia in ambito aziendale che con i loro partner, clienti, fornitori, ...
- ❖ capaci di mutare velocemente in base alle esigenze dei clienti, del mercato, di eventi esterni
- ❖ focalizzati sul core business
- ❖ in grado di incrementare la produttività aziendale

La Soluzione: SOA - Service Oriented Architecture

“La SOA è un’infrastruttura architeturale in cui le funzioni applicative sono realizzate come componenti (servizi) **indipendenti** e con compiti ben definiti per permettere l’**interoperabilità** e promuovere la **flessibilità** e il **riuso**”

- ❖ Le funzioni applicative sono esposte come servizi
- ❖ I servizi sono definiti secondo un’interfaccia standard
- ❖ I servizi rappresentano dei componenti atomici per sviluppare nuove applicazioni

SOA - I benefici

- ✓ **Facilità di costruzione di applicazioni di business end-to-end**
- ✓ **Riusabilità: le applicazioni legacy possono essere integrate nei processi come servizi**
- ✓ **Flessibilità: si richiama l'interfaccia del servizio e non la sua implementazione che può cambiare nel tempo**
- ✓ **Indipendenza dalla piattaforma**

Le radici della SOA

Applicazioni prima del Web

- ❖ Era previsto un processo di apprendimento del sistema SW
- ❖ Le tecnologie di riferimento erano (COBOL, C++, client/server, etc.)
- ❖ Hardware era dedicato
- ❖ L'integrazione e le evoluzioni erano difficili (EDI, CORBA)
- ❖ Il numero limitato degli accessi rendeva semplice gestire la sicurezza
- ❖ I tempi per lo sviluppo si attestavano sui 12-18 mesi

Le radici della SOA

I risultati dell'era “.com”

- ❖ **Chiunque deve essere in grado di utilizzare il nostro sistema (utenti interni, clienti, business partners, etc.)**
- ❖ **Le applicazioni sono sviluppate su architetture *n*-tiered**
- ❖ **Le applicazioni vengono distribuite su diversi hardware**
- ❖ **Il mercato impone degli STANDARD (HTTP, HTML, XML, Java, J2EE, etc.)**
- ❖ **Nasce l'esigenza del Supply chain management**
- ❖ **Diventa un problema gestire la sicurezza**
- ❖ **I tempi per lo sviluppo si accorciano a 3-6 mesi**
- ❖ **E' necessario garantire l'integrazione con applicazioni eterogenee**

SOA - Overview

SOA è costituita da principi, linee guida, best practices architettureali indipendenti da qualsiasi tecnologia e definisce una serie di proprietà che i servizi devono soddisfare per essere realmente riusabili e facilmente integrabili in ambiente eterogeneo.

SOA – Overview: I servizi

❖ I Servizi devono essere ricercabili e recuperabili dinamicamente.

Chi necessita di un servizio deve essere in grado di ricercarlo sulla base dell'interfaccia e di chiamarlo a tempo di esecuzione. Questo tipo di meccanismo permette un forte disaccoppiamento tra chi richiede la funzionalità e chi la fornisce, permettendo inoltre di cambiare l'entità che esegue il servizio a tempo di esecuzione in maniera trasparente rispetto al chiamante.

❖ I Servizi devono essere “esaustivi” e modulari.

Per essere realmente riusabili, è importante che i servizi non siano legati al contesto o allo stato di altri servizi. Ovviamente le applicazioni richiedenti necessitano di avere uno stato persistente tra le invocazioni, ma questo deve essere separato dal fornitore di servizio. In pratica non dovrebbe esserci uno stato conversazionale nelle chiamate tra chi richiede il servizio e chi fornisce il servizio o, per usare una terminologia più precisa, i servizi dovrebbero essere stateless.

SOA – Overview: I servizi

❖ I Servizi devono essere debolmente accoppiati (loosely coupled).

L'accoppiamento si riferisce al numero di dipendenze tra i moduli. Ogni tipo di architettura ben definita è orientata ad avere accoppiamento debole, cioè un numero di dipendenze tra le entità basso e ben controllato. Un sistema formato da componenti fortemente accoppiati è più rigido e difficilmente modificabile. In questo caso il disaccoppiamento si intende sia tecnologico che funzionale:

- Tecnologico: Linguaggio di programmazione, S.O., Protocollo
- Funzionale: Ogni sistema ha la propria rappresentazione dei dati; il messaggio sarà trasformato da un ente intermedio.

❖ I Servizi devono avere un'interfaccia distribuita e devono essere accessibili in maniera trasparente rispetto all'allocazione.

Un servizio con un'interfaccia distribuita può essere pubblicato sulla rete, diventando così disponibile ai componenti che lo vogliono utilizzare.

L'accesso tramite la rete permette inoltre di avere trasparenza rispetto alla reale allocazione del servizio.

SOA - Overview : I servizi

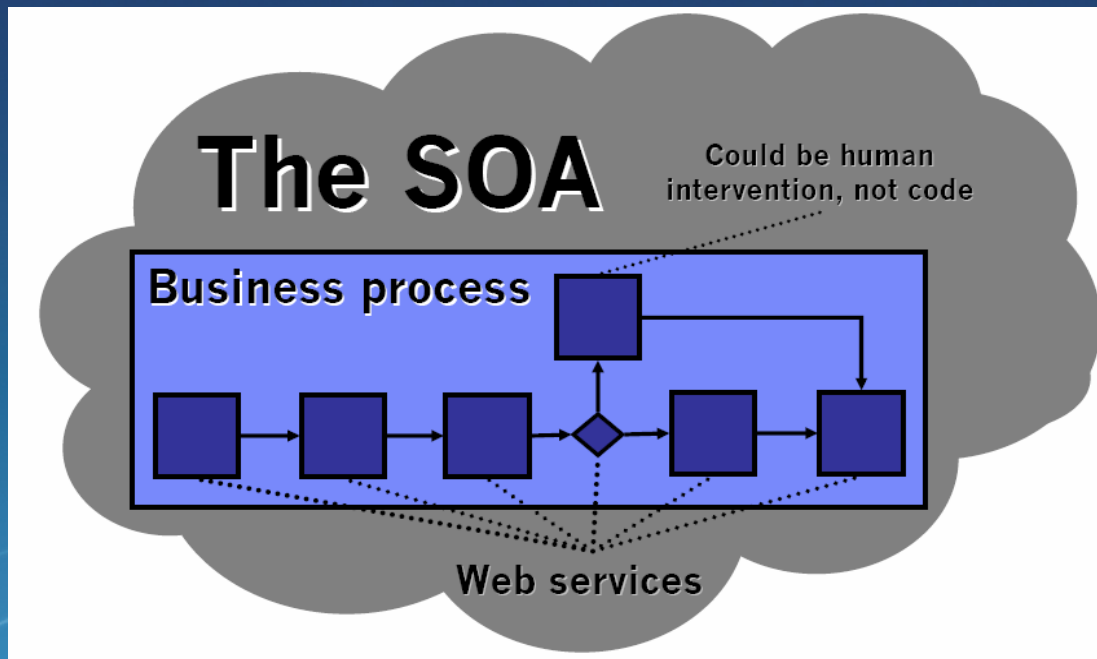
❖ I Servizi devono avere preferibilmente un'interfaccia a "grana grossa" (coarse-grained).

Un servizio che corrisponda ad un'unica chiamata ad un programma complesso ha in genere dei vantaggi rispetto a una serie di chiamate a tanti servizi più piccoli. In questo modo infatti vengono fatte meno chiamate remote (tipicamente poco efficienti), non c'è bisogno di trovare un sistema per mantenere lo stato tra più chiamate ed è più semplice gestire problematiche legate al fallimento della comunicazione remota. Comunque può essere sensato definire servizi "piccoli" soprattutto se questi servizi possono essere utilizzati per essere composti in altri servizi.



SOA - Overview : La mappatura dei Processi di Business

- ❖ **I Servizi devono essere componibili**, ovvero orchestrabili in processi di business ampi che rompano le tradizionali pile applicative verticali (silos) – La flessibilità dei processi di business è uno dei principali obiettivi di un progetto SOA in cui ogni step è rappresentato da un servizio applicativo



SOA - I ruoli

Le applicazioni SOA definiscono dei ruoli:

- ✓ **Service Requester:** l'entità che richiede il servizio; può essere un modulo di un'applicazione o un altro servizio.
- ✓ **Service Provider:** l'entità che fornisce il servizio e che ne espone l'interfaccia.
- ✓ **Service Description:** definisce il formato per la richiesta di un servizio e della relativa risposta.
- ✓ **Service Broker:** Direttorio in rete dei servizi consultabili.

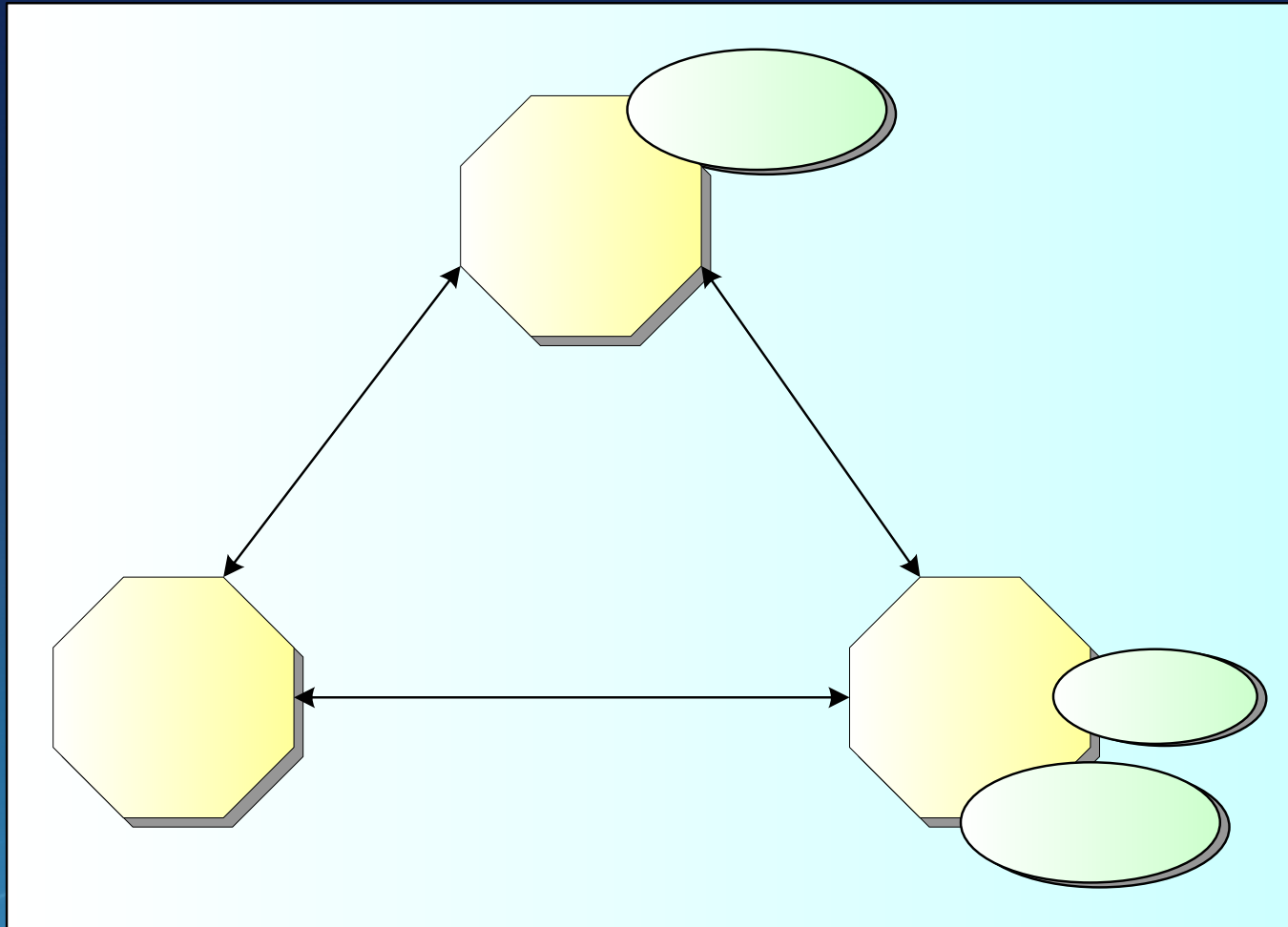
SOA – Le operazioni

Publish – L'operazione di publish rappresenta l'azione di registrazione del servizio o l'annuncio di inserzione di un nuovo servizio. Essa funge da contratto tra il service registry ed il service provider.

Find – Con l'operazione di find, il service requestor dichiara uno o più criteri di ricerca, come il tipo di servizio, il QoS, e così via. Il risultato di tale operazione è una lista di descrizioni dei servizi che rispettano tali criteri.

Bind – L'operazione di bind rappresenta la relazione client-server tra il service requestor e il service provider. Tale operazione può essere dinamica, come una connessione client-server 'al volo' basata sul fatto di poter invocare la descrizione di un servizio per utilizzarlo; oppure in modalità statica, consistente nel fatto di poter invocare un servizio manualmente, richiedendolo con del codice apposito sul lato client dell'applicazione.

SOA – Big Picture



Agenda

Service Oriented Architecture

- Overview SOA
- **SOA Technologies**

ACG Service Bus

- Overview
- Programming model
- Development tools

SOA Technologies: Overview

I Web Services sono la tecnologia abilitante per la SOA. Essi si basano su:

- **SOAP** è il protocollo, basato su XML, scelto dai provider per il trasporto dei messaggi.
- Il **Web Services Description Language (WSDL)** descrive come accedere ad un risorsa nella SOA.
- I **Web Service Discovery standards (UDDI & WSIL)**.

L'implementazione dei Business Processes avviene attraverso il **BPEL**

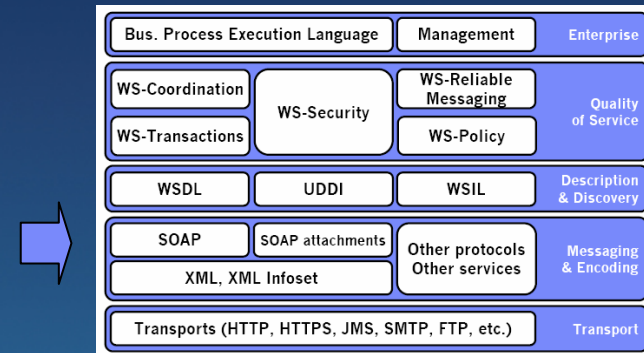
SOAP (Simple Object Access Protocol)

SOAP è un documento XML, costituito da un envelope, un header e un body, usato per spedire dati tra due o più end-point. E' indipendente dal:

- network
- protocol
- information
- endpoints

Un messaggio SOAP è caratterizzato da:

- la descrizione del contenuto di un messaggio e delle regole di elaborazione (SOAP envelope);
- una serie di regole di codifica per rappresentare i tipi di dati definiti per l'applicazione (SOAP encoding rules);
- una serie di convenzioni per rappresentare le invocazioni remote e le corrispondenti risposte (SOAP RPC).

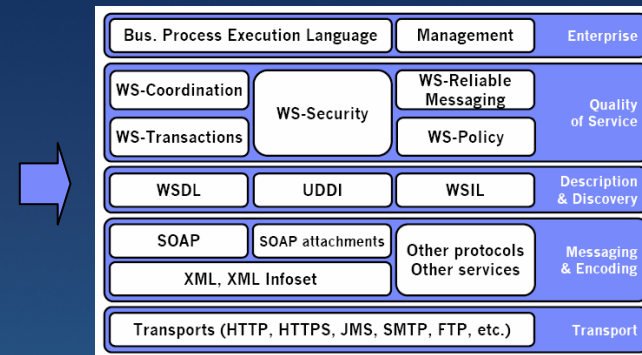


WSDL – Web Service Description Language

Un documento WSDL è un file in formato XML che descrive l'interfaccia di un Web Service in modo indipendente dal linguaggio e dal protocollo.

Le informazioni che contiene sono:

- l'URL dell'host
- il nome del servizio
- il tipo (RPC/Document)
- i metodi e i loro parametri



Ciò di cui ha bisogno il client per accedere ad un WS è rappresentato dal WSDL

Ulteriori info: <http://www.w3.org/TR/wsd>

UDDI (Universal Description, Discovery, and Integration)

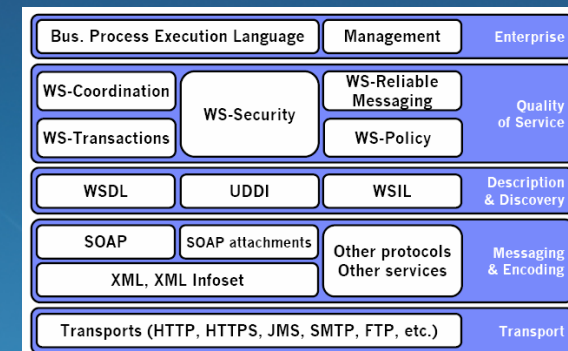
Il protocollo UDDI è utilizzato dai servizi di directory per consentire agli utenti di localizzare i Web Services

E' gestito dal consorzio UDDI, che annovera tra i suoi membri IBM, Microsoft, Ariba e SUN Microsystems.

L' UDDI si propone come un *“framework aperto ed indipendente dalla piattaforma per descrivere servizi, individuare società e integrare servizi di business utilizzando Internet”*

Ha le funzionalità di un registro contenitore, all'interno del quale i documenti WSDL possono essere rintracciati come voci di registro.

Ulteriori info: <http://www.uddi.org>



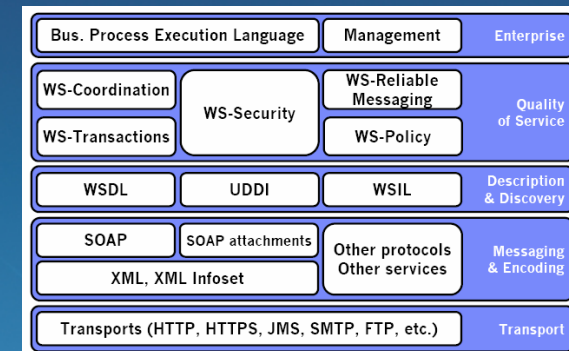
WSIL (Web Services Inspection Language)

Il WS-Inspection Language definisce la location di un sito Web dove si trova il Web service description document (WSDL).

Il WSIL fornisce informazioni relativamente a :

- Service: le differenti service description per un Web Service
- Link: gli URL di UDDI o delle location dei vari WSDL

Il WSIL si propone come metodo complementare all'UDDI per facilitare la scoperta dei servizi disponibili da parte di un'organizzazione



SOA Technologies: Overview

Il mapping tra le entità SOA definite e i WebServices è molto semplice:

WS	Service Provider
Client di WS	Service Requester
SOAP/HTTP	Communication Protocol
WSDL	Service Contract
UDDI	Service Broker

Web Services

“I Web Services sono applicazioni modulari e atomiche che possono essere descritte, pubblicate, localizzate, ed invocate in una rete mediante standard aperti”

Qualsiasi codice legacy (RPG, Java, ..) può essere reso fruibile come Web Service.

Ogni Web Service fornisce funzionalità utilizzabili remotamente come se fossero locali.

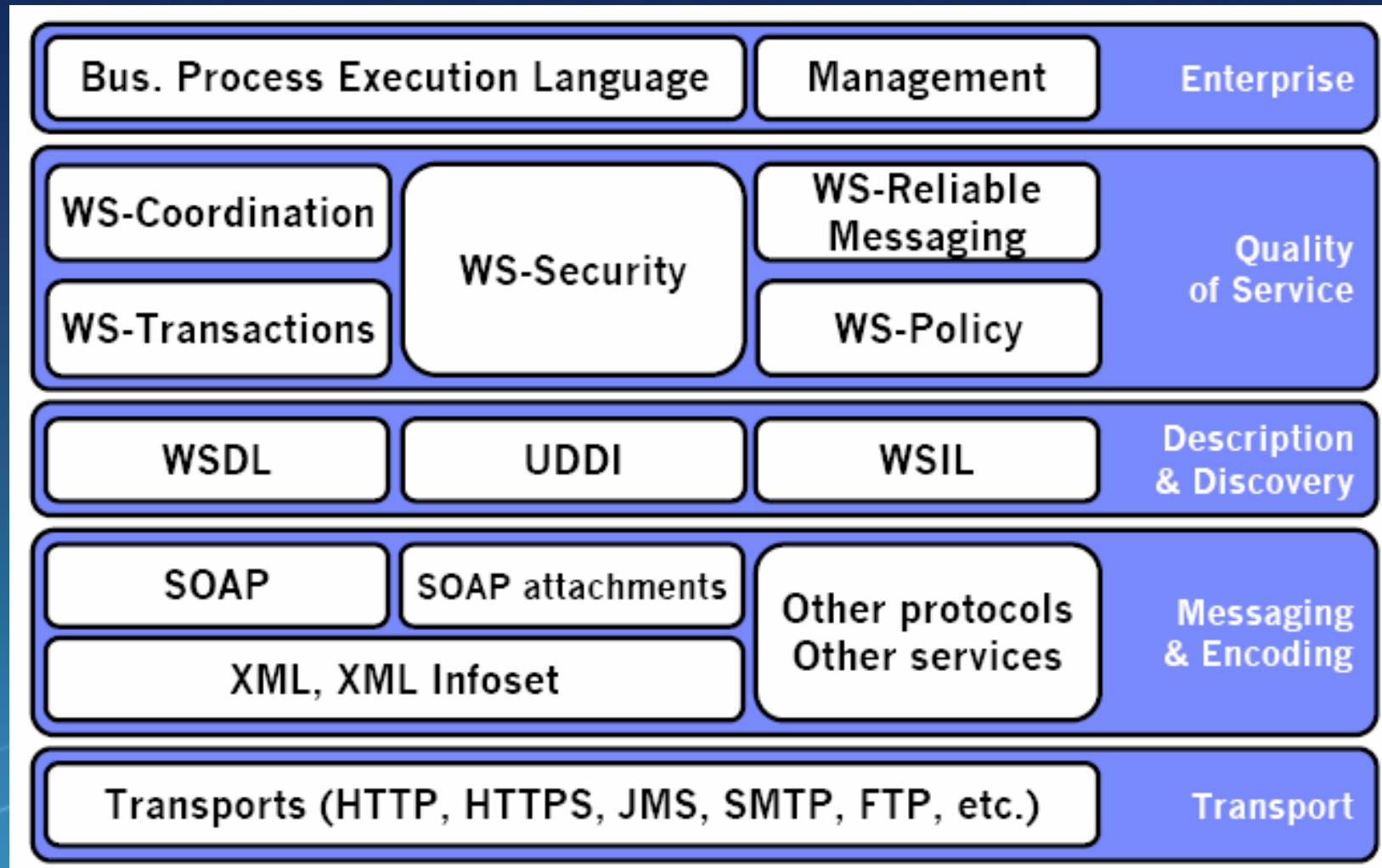
I Web Services scambiano messaggi XML tra le applicazioni

Web Services

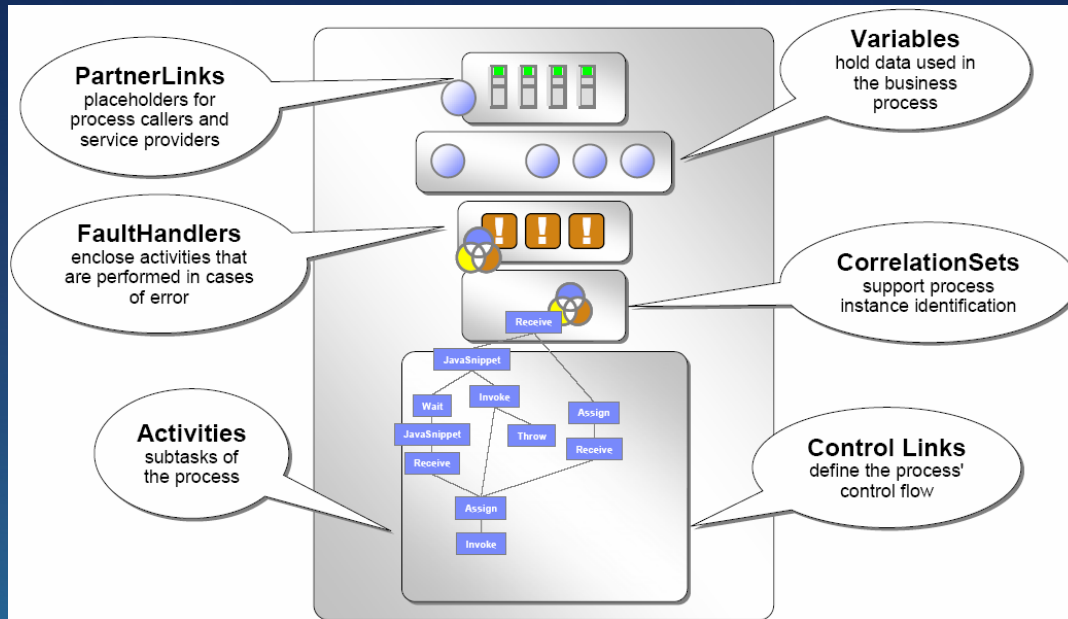
Un Web Service:

- ✓ ha un'interfaccia pubblica descritta da un file WSDL
- ✓ comunica tramite standard aperti (XML, HTTP, JAVA,..)
- ✓ può richiamare altri Web Services in modo da realizzare un workflow di operazioni atomiche

Web Services - Standards



BPEL (Business Process Execution Language)



Gli elementi più importanti di un BPEL Process sono:

Activities – Rappresentano uno step del processo di Business svolto da un'applicazione o da una persona

Control Links – Insieme concettuale di attività all'interno del flusso

Fault Handlers – Rappresentano le azioni da intraprendere in caso di errori

Variable – rappresentano informazioni significative che vanno tracciate durante l'esecuzione del processo

Un documento BPEL è un file XML che definisce un Processo di Business come flusso di chiamate tra Web Services.

Bus. Process Execution Language	Management	Enterprise
WS-Coordination	WS-Security	WS-Reliable Messaging
WS-Transactions		WS-Policy
WSDL	UDDI	WSIL
SOAP	SOAP attachments	Other protocols
XML, XML Infoset		Other services
Transports (HTTP, HTTPS, JMS, SMTP, FTP, etc.)		Transport

Agenda

Service Oriented Architecture

- Overview SOA
- SOA Technologies
- Web services

ACG Service Bus

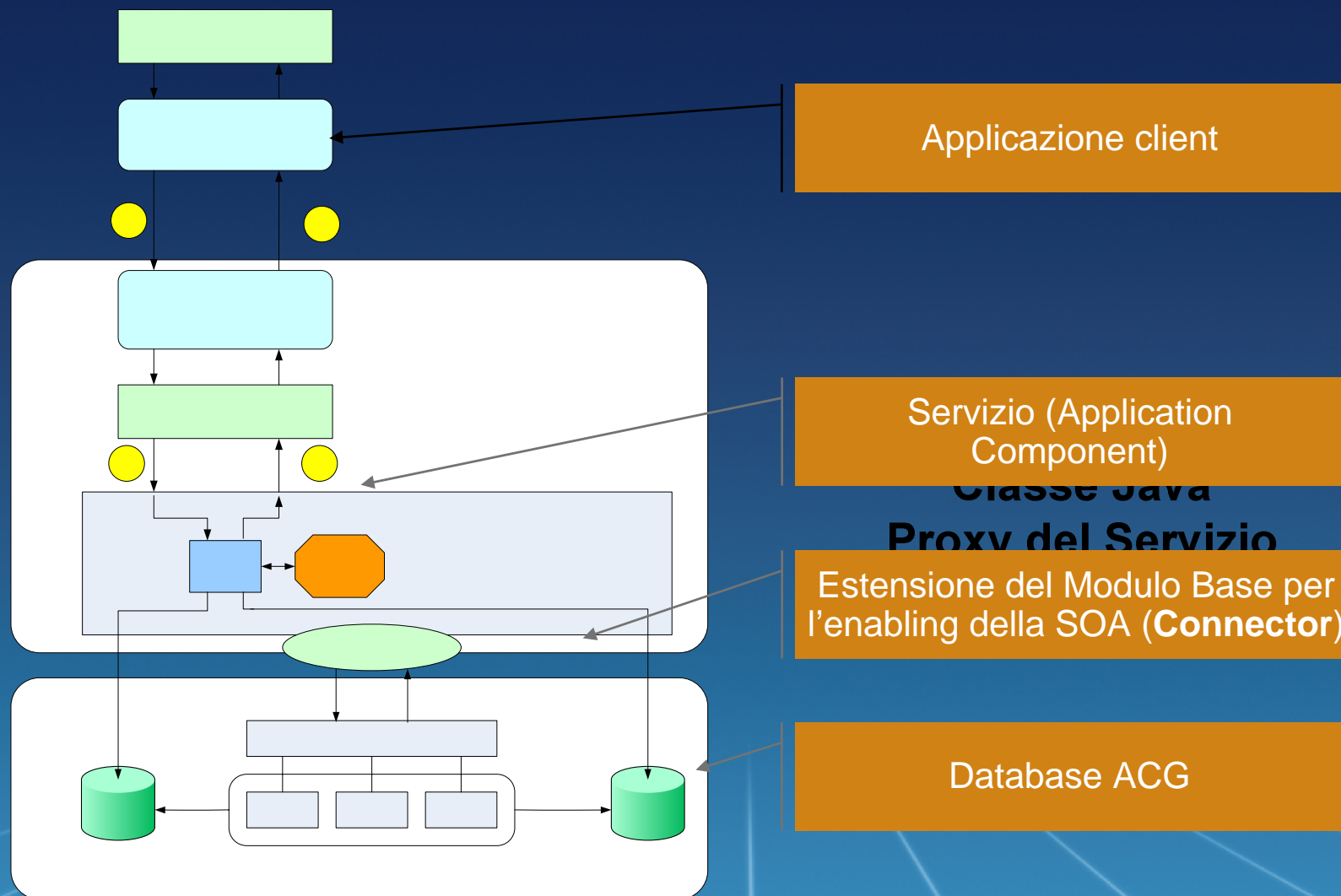
- **Overview**
- Programming model
- Development tools

ACG Service Bus - Overview

Implementare una tecnologia che traghetti le funzionalità ACG verso il mondo SOA

Rendere fruibili le azioni ACG da parte di un Service Requester tramite le tecnologie proprie dei Web Services

ACG Service Bus - Architettura



Applicazione client

Servizio (Application Component)

Classe Java
Proxy del Servizio

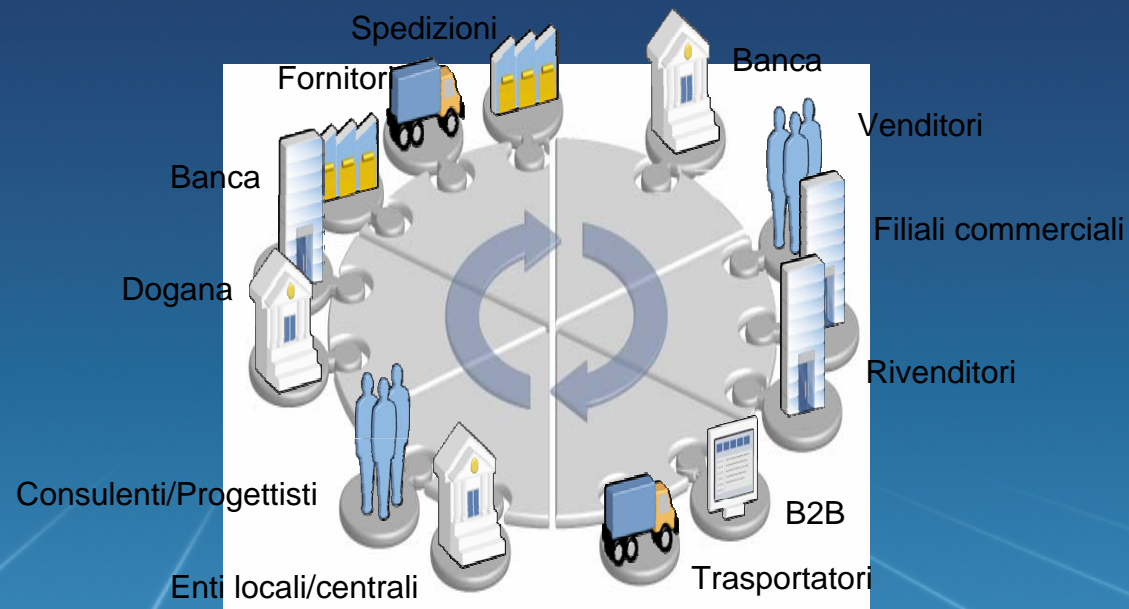
Estensione del Modulo Base per l'enabling della SOA (Connector)

Database ACG

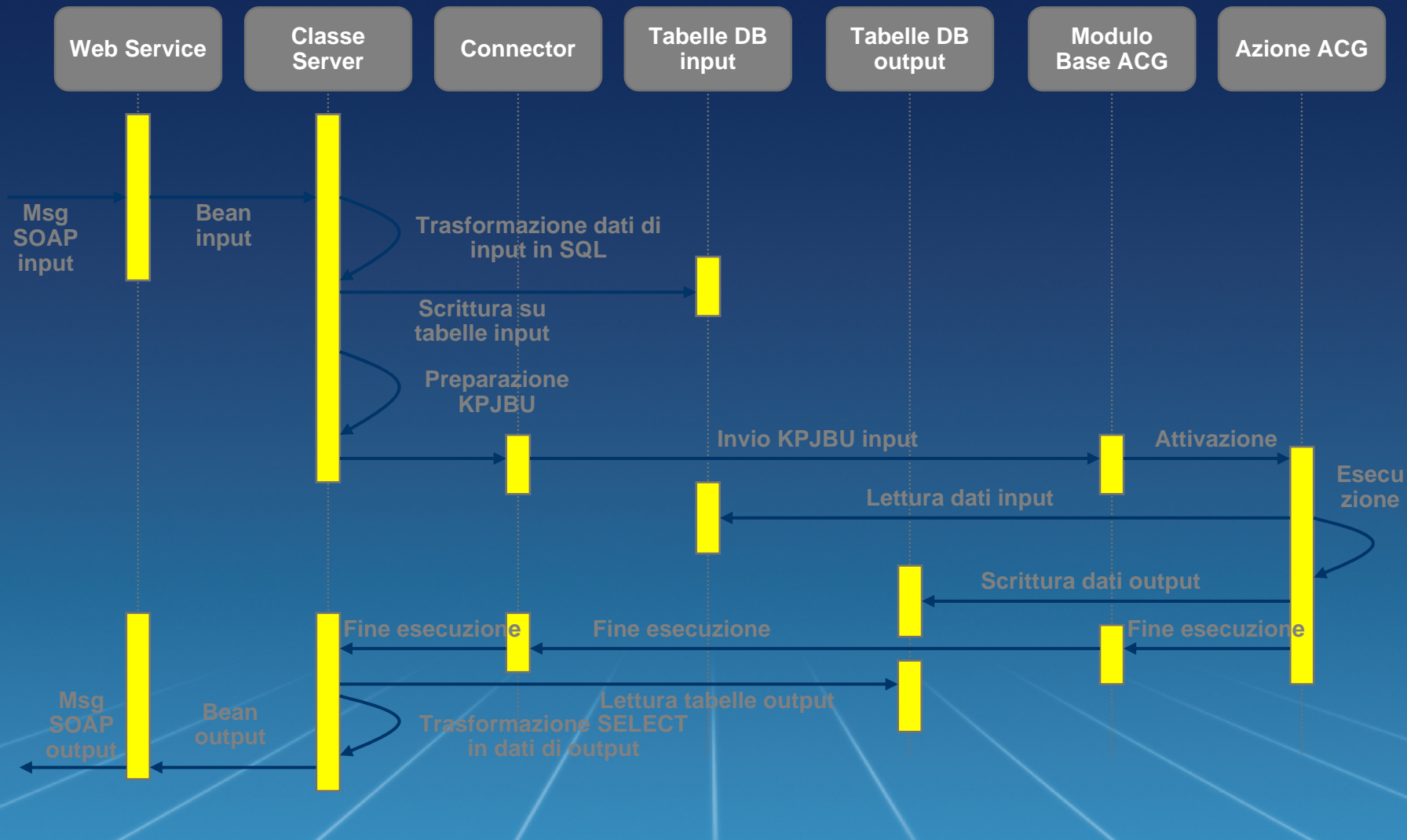


ACG Service Bus - Architettura

Client: qualsiasi applicazione in grado di richiamare un Web Service (J2EE, Websphere Portal Server, .Net, SAP, ...)



Architettura ACG SOA: Sequence Diagram



ACG Service Bus - Architettura

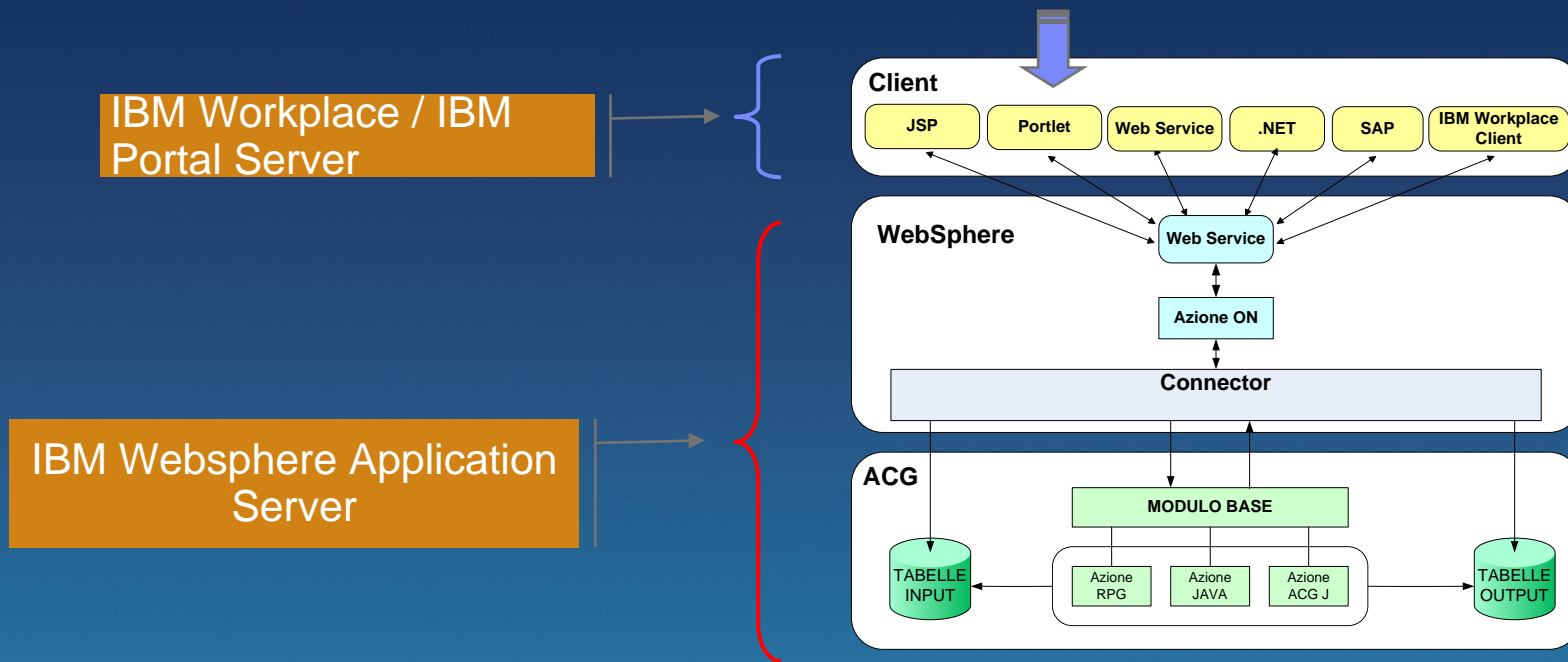
Application Component (Servizio): una classe JAVA (denominata “action”) che espone le funzionalità:

- **runService():** è un metodo che nello standard ACG SOA provvede a richiamare un’azione di Modulo Base e a restituire il risultato dell’elaborazione
- **runHelp():** è un metodo che nello standard ACG SOA fornisce informazioni di supporto all’uso

Connector: è un insieme di classi Java che eseguono:

- Logon ad ACGforJava/ACG WE
- Lanciano un’azione di ACG Service Bus

ACG Service Bus - Architettura

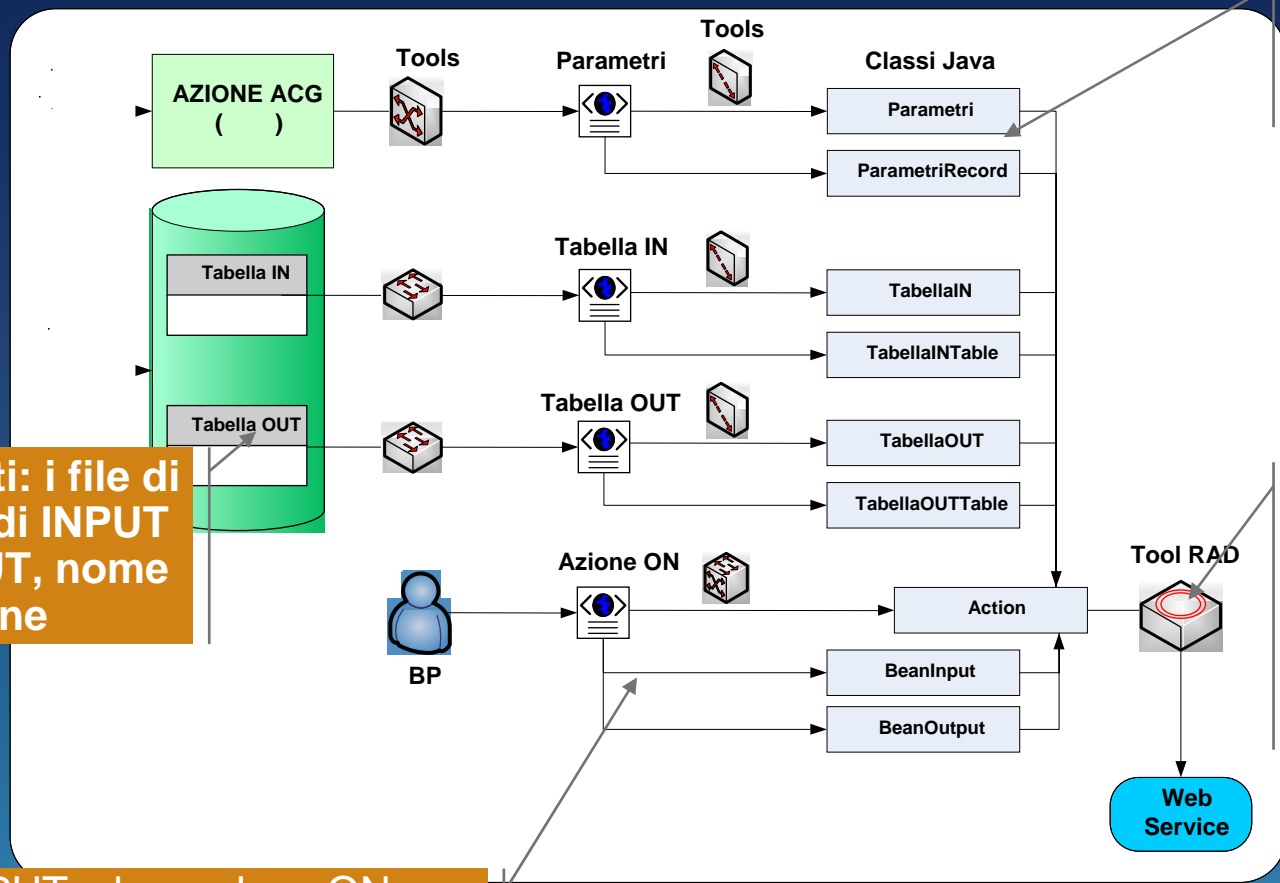


Output: Java Beans e classe Action

Utilizzo di Rational Application Developer per la generazione del Web Service

Prerequisiti: i file di database di INPUT ed OUTPUT, nome azione

INPUT: classe Java ONxx, File XML



La DS ONXXDS

```
tecp.WS
File Modifica Visualizza Comunicazioni Azioni Finestra ?
Colonne . . . : 1 71 Esame ACGON_SRC/QDDSDS
SEU==> ON01DS
FMT PF .....A.....T.Nome+++++RLun++TPdB.....Funzioni+++++
***** Inizio dati *****
0001.00 00010 REF(*LIBL/DIZACGV3)
0002.00 00020A R ON01R
0003.00 00030A* ID transazione
0004.00 00040A IDENT 35A
0005.00 00050A* Codice cliente/fornitore
0006.00 00060A CLIF00 R REFFLD(CDCLI)
0007.00 00070A* Flag cliente/fornitore (val. ammissibili C/F)
0008.00 00080A FCLF00 R REFFLD(FCLF0)
***** Fine dati *****

F3=Fine F5=Rivisual. F9=Duplicaz. F10=Cursore F11=Alternare F12=Annullam.
F16=Ripetiz. ricerca F24=Altri tasti
(C) COPYRIGHT IBM CORP. 1981, 2002.
MA b MW 02/009
Collegato con server/host remoto 9.87.104.10 mediante l'utilizzo della porta 23
```

E' OBBLIGATORIO il campo IDENT di 35 caratteri

La Tabella ONXXO00F

```
tecp.WS
File Modifica Visualizza Comunicazioni Azioni Finestra ?
Colonne . . . : 1 71      Esame      ACGON_SRC/QDDSPF
SEU=>          ON020
FMT PF .....A.....T.Nome+++++RLun++TPdB.....Funzioni+++++.....
***** Inizio dati *****
0001.00 00010A          REF (DIZACGV3)
0002.00 00020A          R ON020
0003.00 00030 *-----*
0004.00 00040 * Informazioni generali *
0005.00 00050 *-----*
0006.00 00060A          CDCLI      R
0007.00 00070A          RASCL      R
0008.00 00080A          RASC2      R
0009.00 00090A          CDRIC      R
0010.00 00100A          INDCL      R
0011.00 00110A          LOCCL      R
0012.00 00120A          PROCL      R
0013.00 00130A          CAPCL      R
0014.00 00140A          NTECL      R
0015.00 00150A          CNOTE      R
0016.00 00160A          CDNAZ      R

F3=Fine   F5=Rivisual. F9=Duplicaz. F10=Cursore F11=Alternare F12=Annullam.
F16=Ripetiz. ricerca F24=Altri tasti

MA  b          MW          02/009
Collegato con server/host remoto 9.87.104.10 mediante l'utilizzo della porta 23
```

Definizione del file fisico di output del WS

E' OBBLIGATORIO il campo IDENT di 35 caratteri

Il programma RPG

```
tecp.WS
File Modifica Visualizza Comunicazioni Azioni Finestra ?
Colonne . . . : 1 71      Esame      ACGON_SRC/QRPGSRC
SEU==> _____      ONO2PGM
FMT H .....H.....1..CDYI...S.....1.F.....
***** Inizio dati *****
0001.00 00010H      YJ
0002.00 00020 *
0003.00 00030FANCL201LIF E      K      DISK
0004.00 00040FANF0201LIF E      K      DISK
0005.00 00050FON02000F0 E      DISK      A
0006.00 00060FONERR00F0 E      DISK      A
0007.00 00070E*
0008.00 00080E      MSG      1      3 80
0009.00 00090I*
0010.00 00100IKPJBA E DS
0011.00 00110ION01DS E DS
0012.00 00120I* Struttura per parametri di architettura
0013.00 00130C*****
0014.00 00140C* DICHIARAZIONE DEI PARAMETRI RICEVUTI ALLA PARTENZA
0015.00 00150C*-----*
0016.00 00160C*

F3=Fine F5=Rivisual. F9=Duplicaz. F10=Cursore F11=Alternare F12=Annullam.
F16=Ripetiz. ricerca F24=Altri tasti

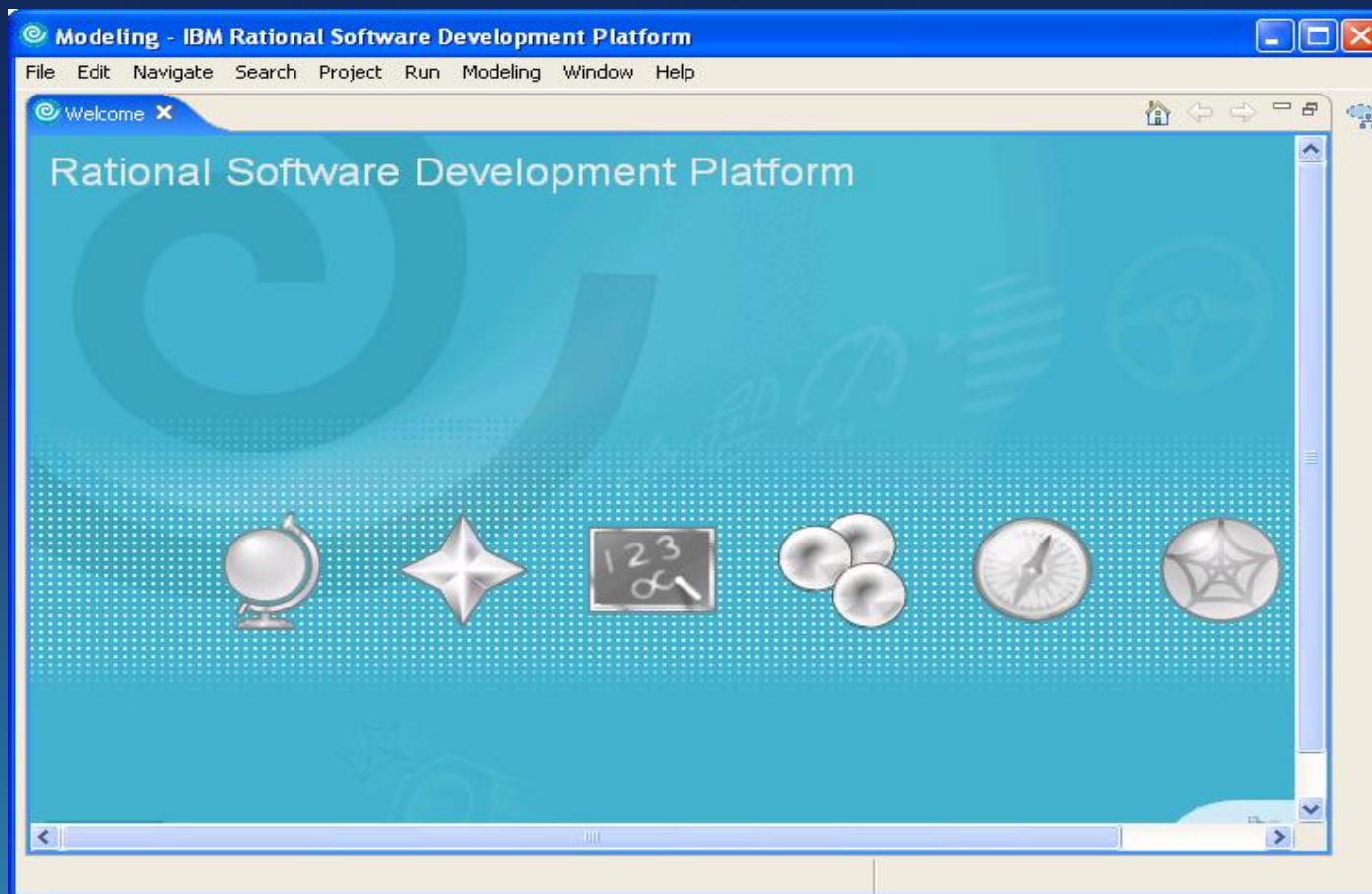
(C) COPYRIGHT IBM CORP. 1981-2000
MA b MW 02
Collegato con server/host remoto 9.87.104.10 mediante l'utilizzo della porta 23
```

File Applicativi

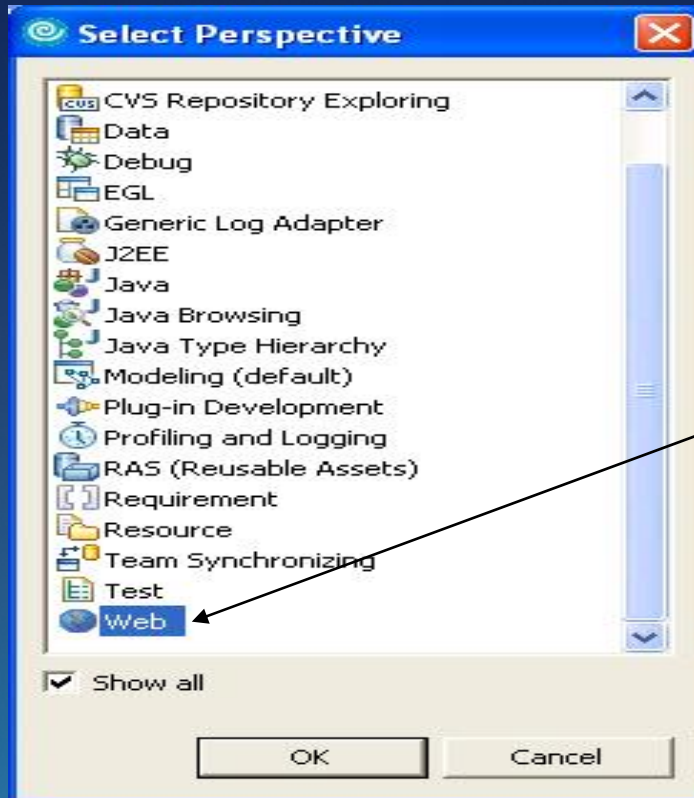
File di lavoro del WS

DS di decodifica della KPJBU impostata dal WS

Lo sviluppo Java Web Service

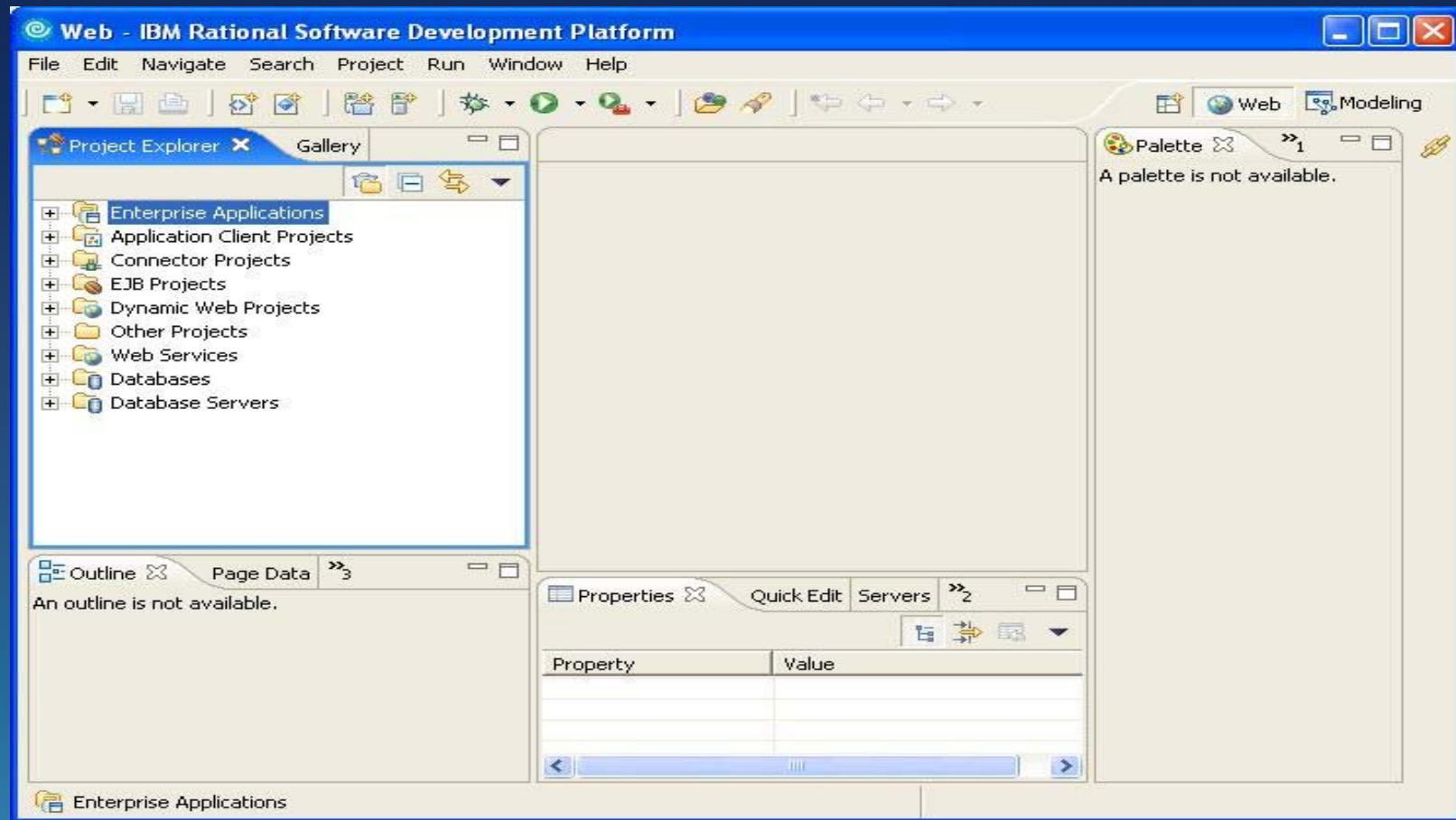


Lo sviluppo Java Web Service

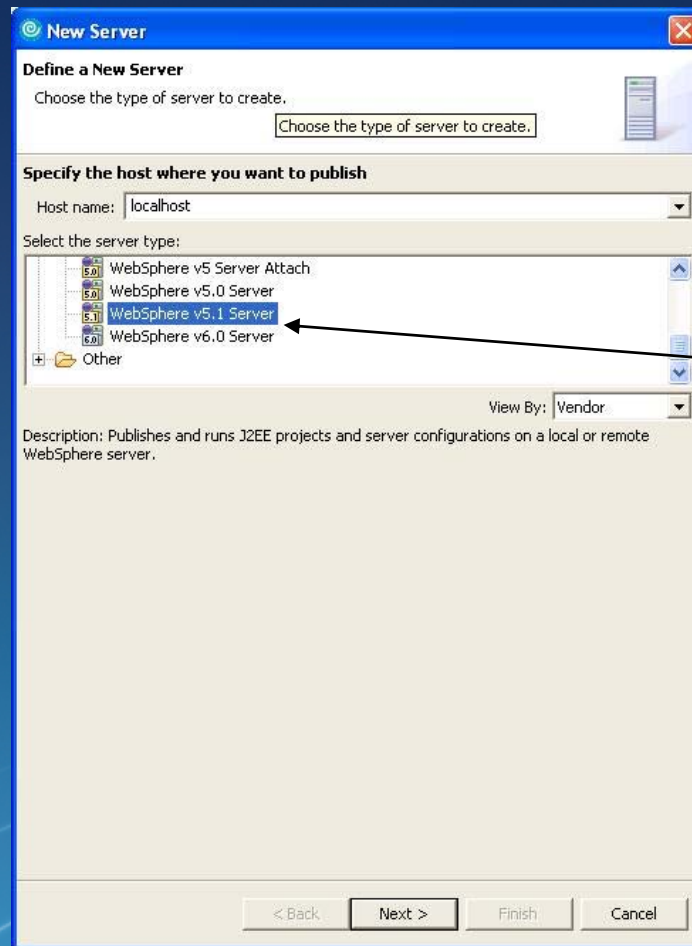


Selezionare la
prospettiva web

Lo sviluppo Java Web Service

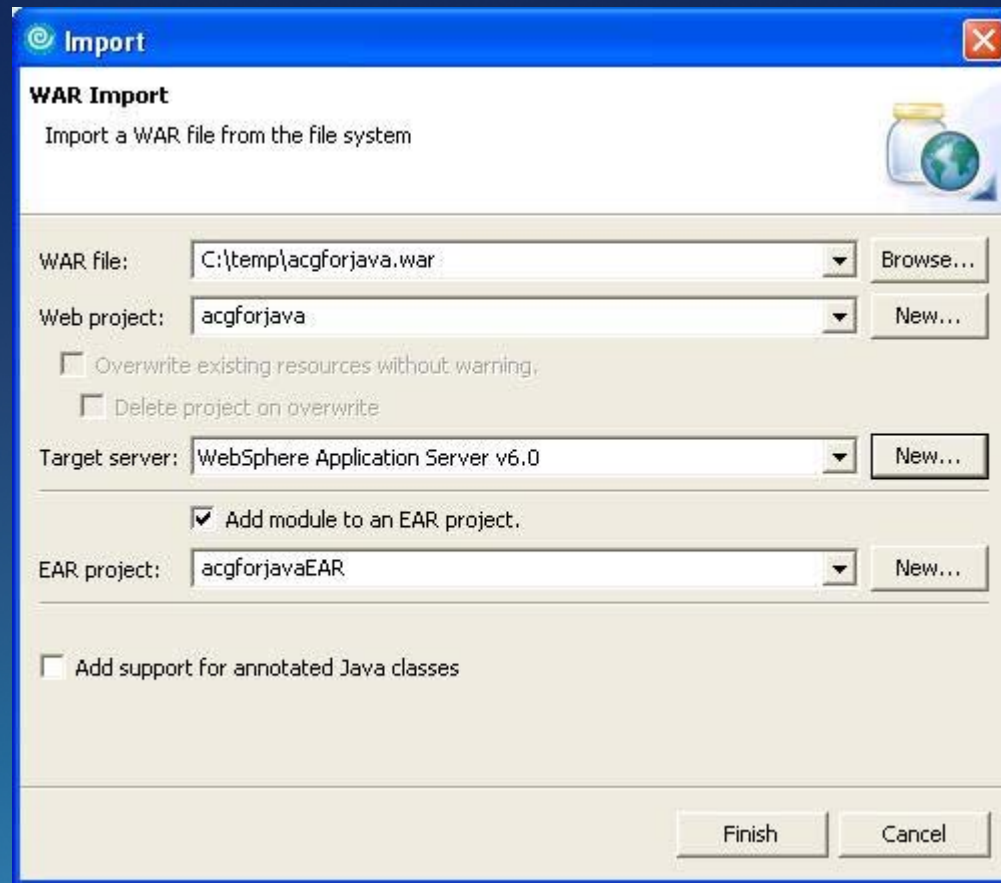


Lo sviluppo Java Web Service



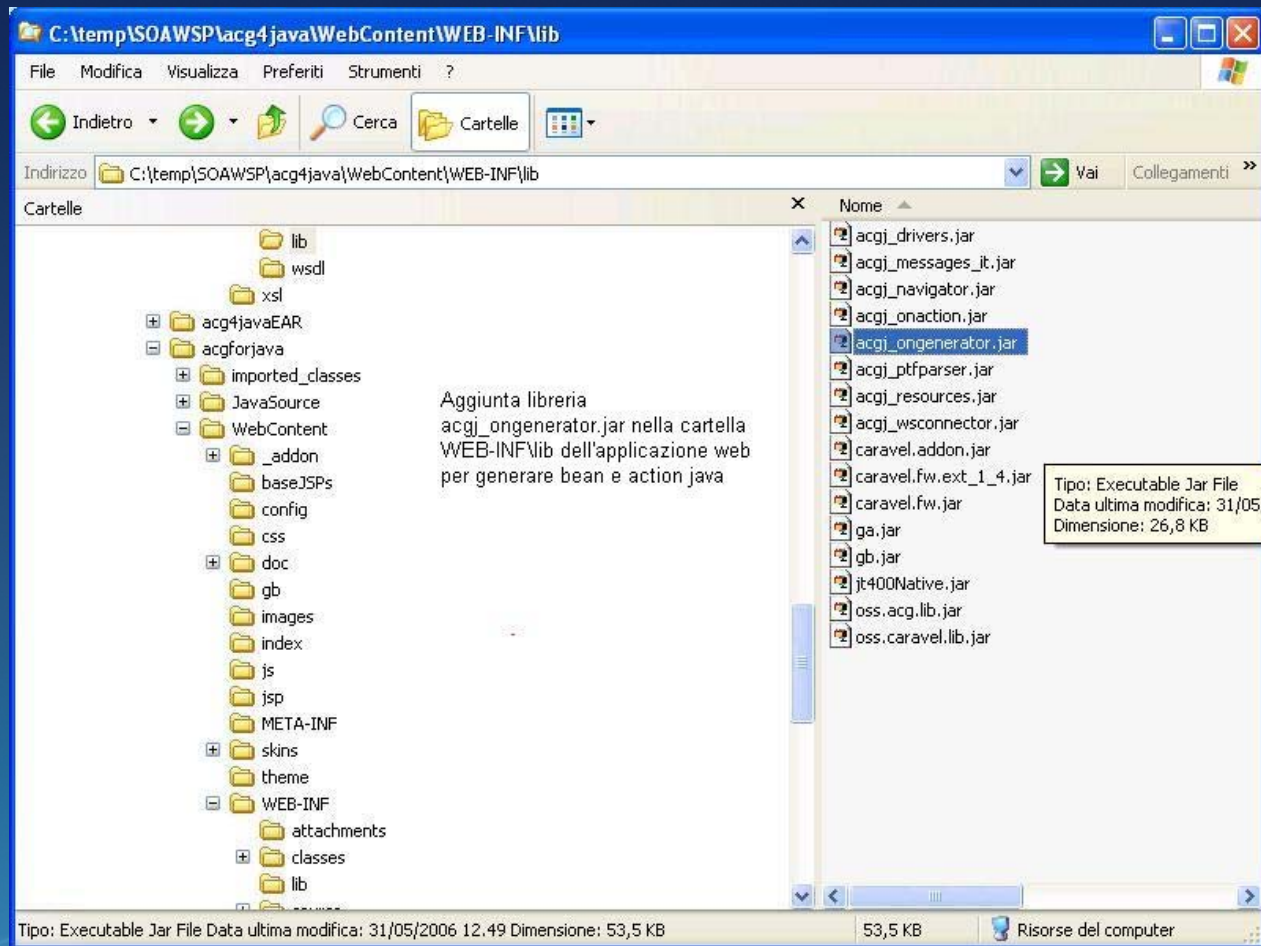
Creare un'istanza di WAS in RAD per lo sviluppo ed il test dei web services

Lo sviluppo Java Web Service



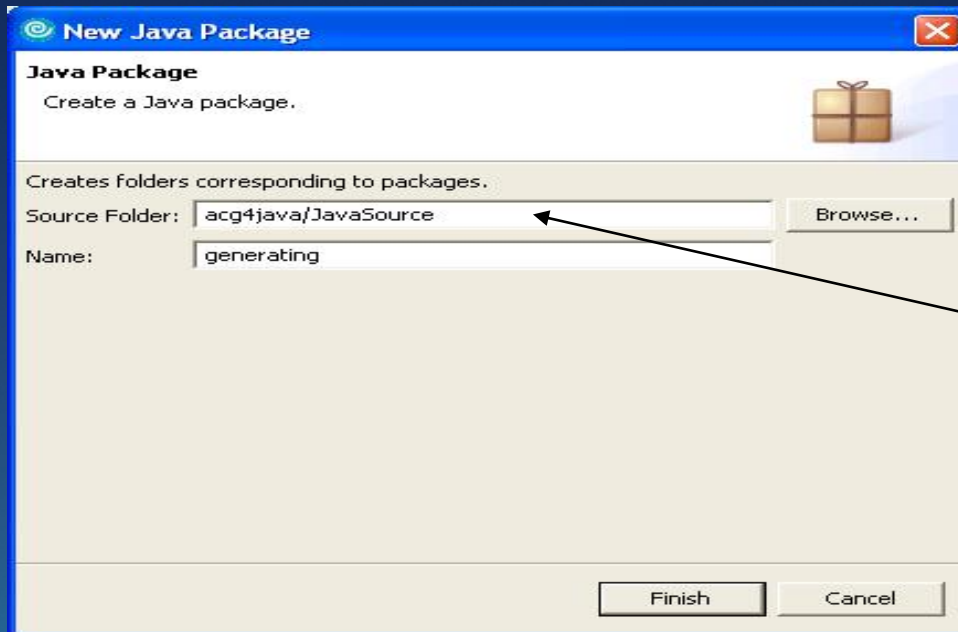
Importare la web application in RAD come WAR file

Lo sviluppo Java Web Service



Importare il generatore di codice **acgj_generator.jar** dalla cartella “Web services” nel CD nr. 4 di ACG Service Bus

Lo sviluppo Java Web Service



Creare un package per contenere le classi java funzionali alla generazione del codice java per i WS

Lo sviluppo Java Web Service

New Java Class

Create a new Java class.

Source Folder: Browse...

Package: Browse...

Enclosing type: Browse...

Name:

Modifiers: public default private protected
 abstract final static

Superclass: Browse...

Interfaces: Add...
Remove

Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

Finish Cancel

**Creare una classe
Java per la
generazione dei
bean java relativi ad
un WS che invoca
un'azione di ACG
Service Bus**

Lo sviluppo Java Web Service

```
package generating;

import com.ibm.acg.on.generator.ACGGenerator;

public class ON02 {

    public static void main(String[] args) {

        String sistema = "olyp";
        String user = "ACGFORJAVA";
        String password = "ACGFORJAVA";
        String lib_obj= "ACGONJ";
        String lib_dat= "XDAT4";

        ACGGenerator gen = new ACGGenerator();
        gen.createDS(sistema, user, password, lib_obj, "ON01DS");
        gen.createTable(sistema, user, password, lib_dat, "ON02O00F");
        gen.createTable(sistema, user, password, lib_dat, "ONERR00F");
        gen.createAction("xml/ON02Action.xml");

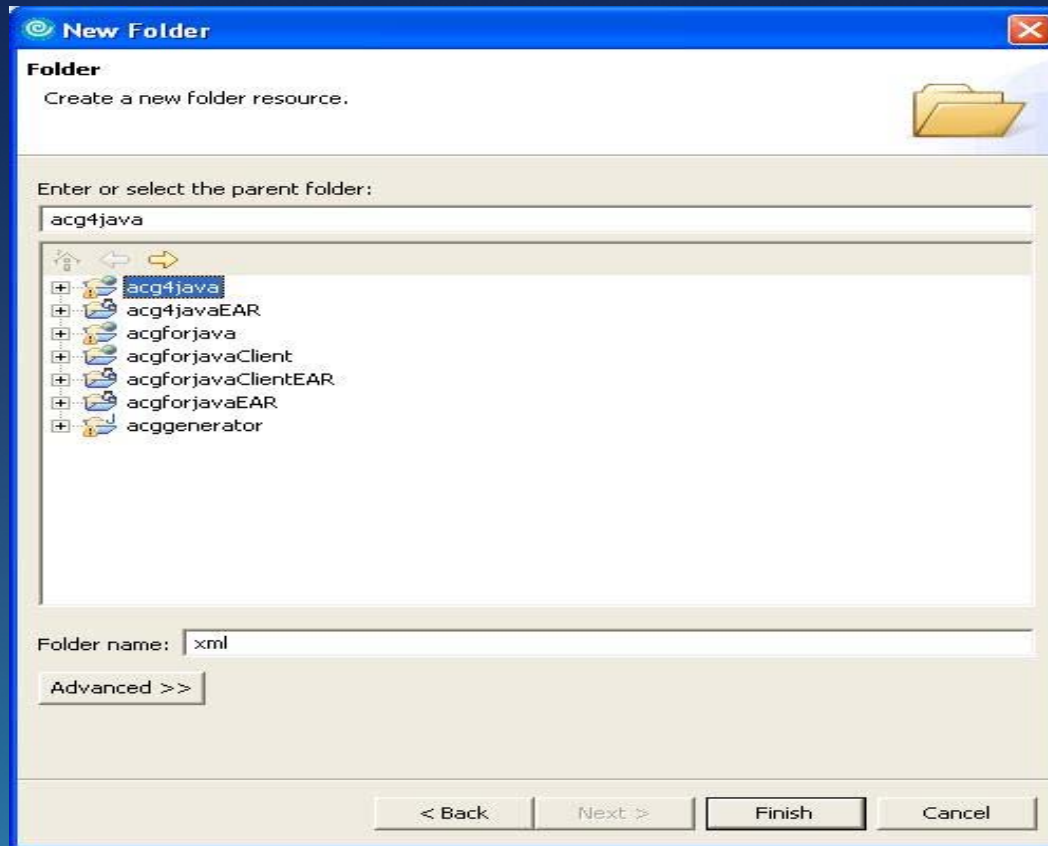
        System.out.println("----- GENERATING END -----");

    }

}
```

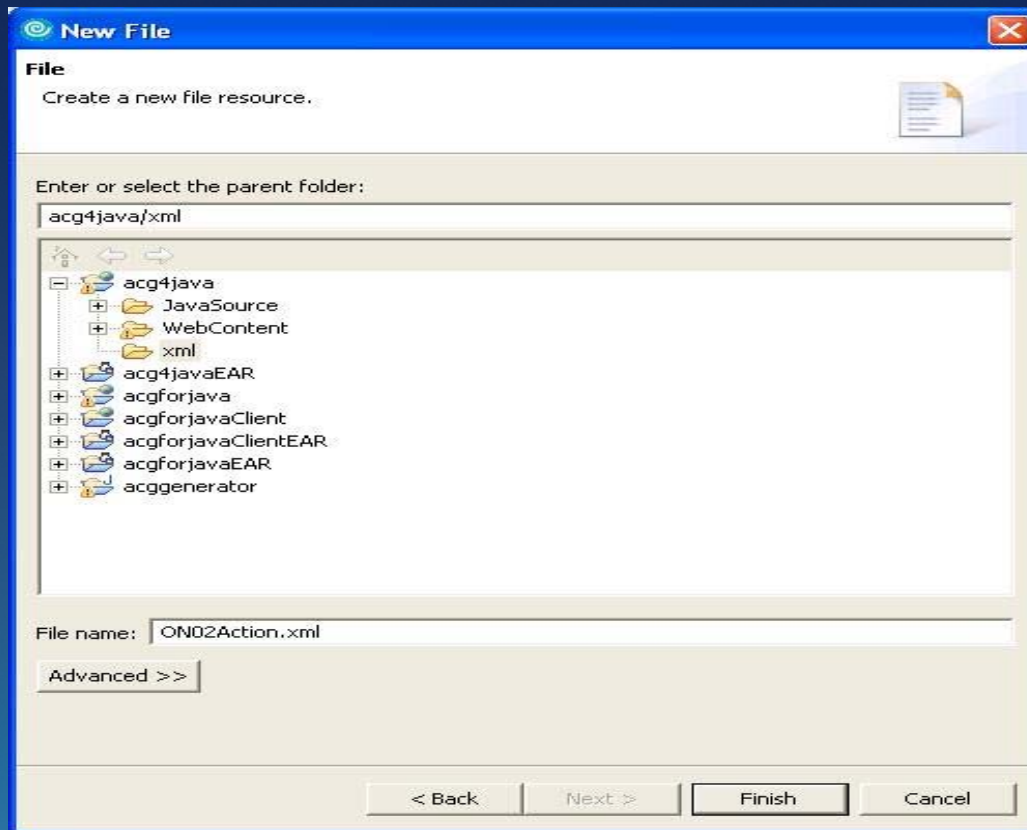
**Un esempio di
codice per la
generazione**

Lo sviluppo Java Web Service



Creare un folder (cartella) nel progetto per inserirvi un file XML che sarà utilizzato dalla classe java per la generazione dei bean e dell'azione associati al WS

Lo sviluppo Java Web Service



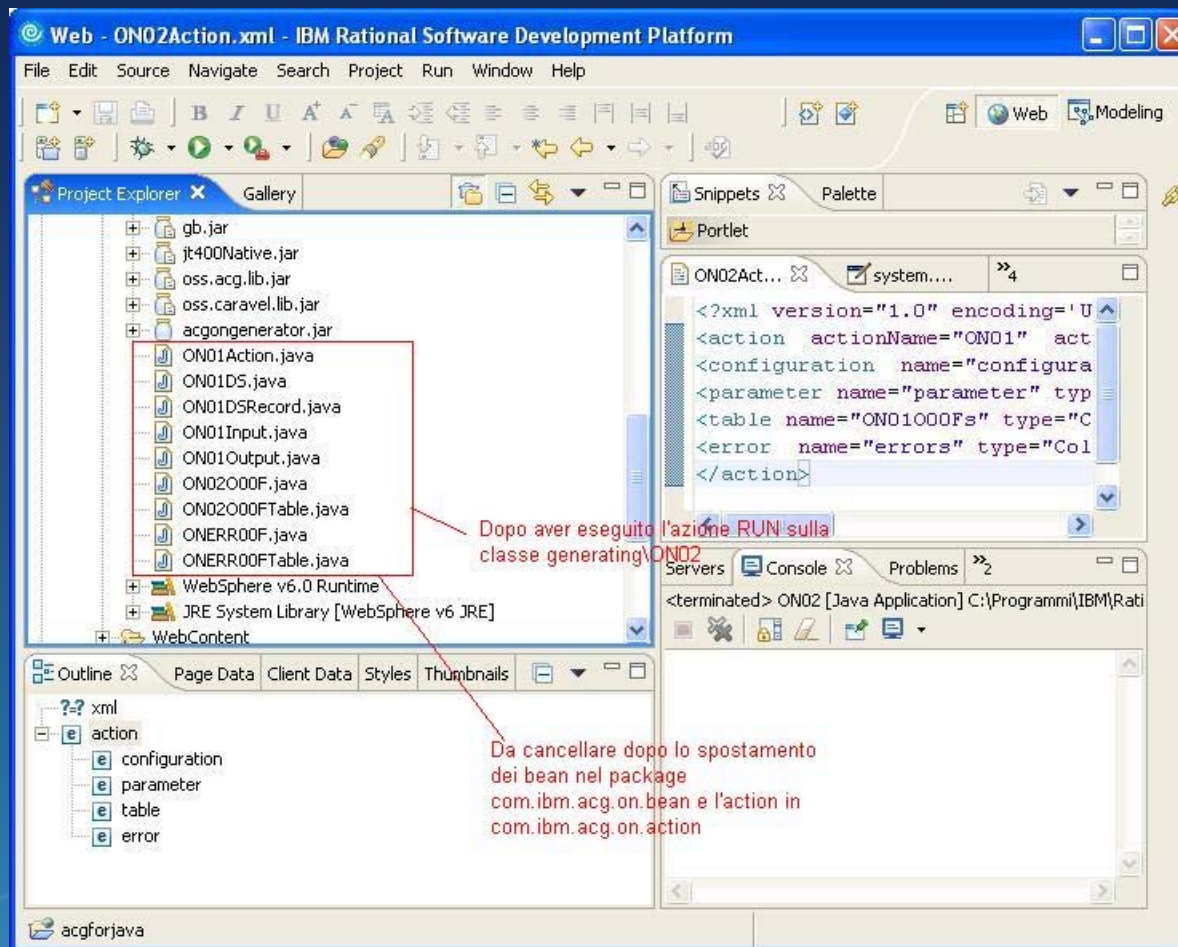
**Creazione del file
XML per la
generazione del
codice java**

Lo sviluppo Java Web Service

```
<?xml version="1.0" encoding='UTF-8' ?>
<action actionName="ON01" actionType='AS400RPG'>
<configuration name="configuration" type="ConfBean"/>
<parameter name="parameter" type="ON01DS"/>
<table name="ON01000Fs" type="Collection" baseType="ON01000F" usage="o
<error name="errors" type="Collection" baseType="ErrorBean" />
</action>
```

Esempio di contenuto del file XML per la generazione del codice java: contiene il nome e le informazioni relative all'input e all'output dell'azione

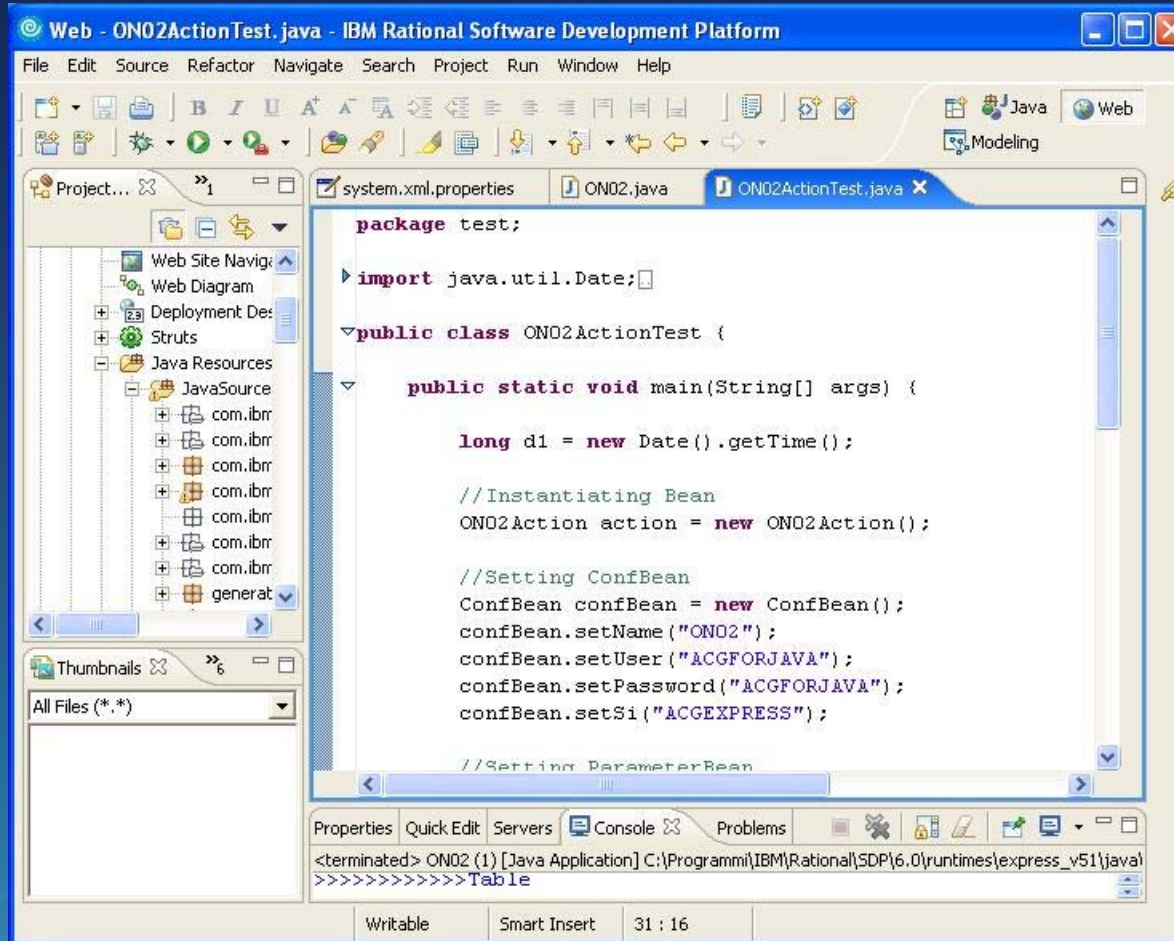
Lo sviluppo Java Web Service



Il risultato della generazione del codice Java, ottenuta dall'esecuzione della classe Java di generazione come semplice applicazione Java

Eseguire il refresh del progetto per visualizzare il risultato della generazione

Lo sviluppo Java Web Service



```
package test;

import java.util.Date;

public class ONO2ActionTest {

    public static void main(String[] args) {

        long d1 = new Date().getTime();

        //Instantiating Bean
        ONO2Action action = new ONO2Action();

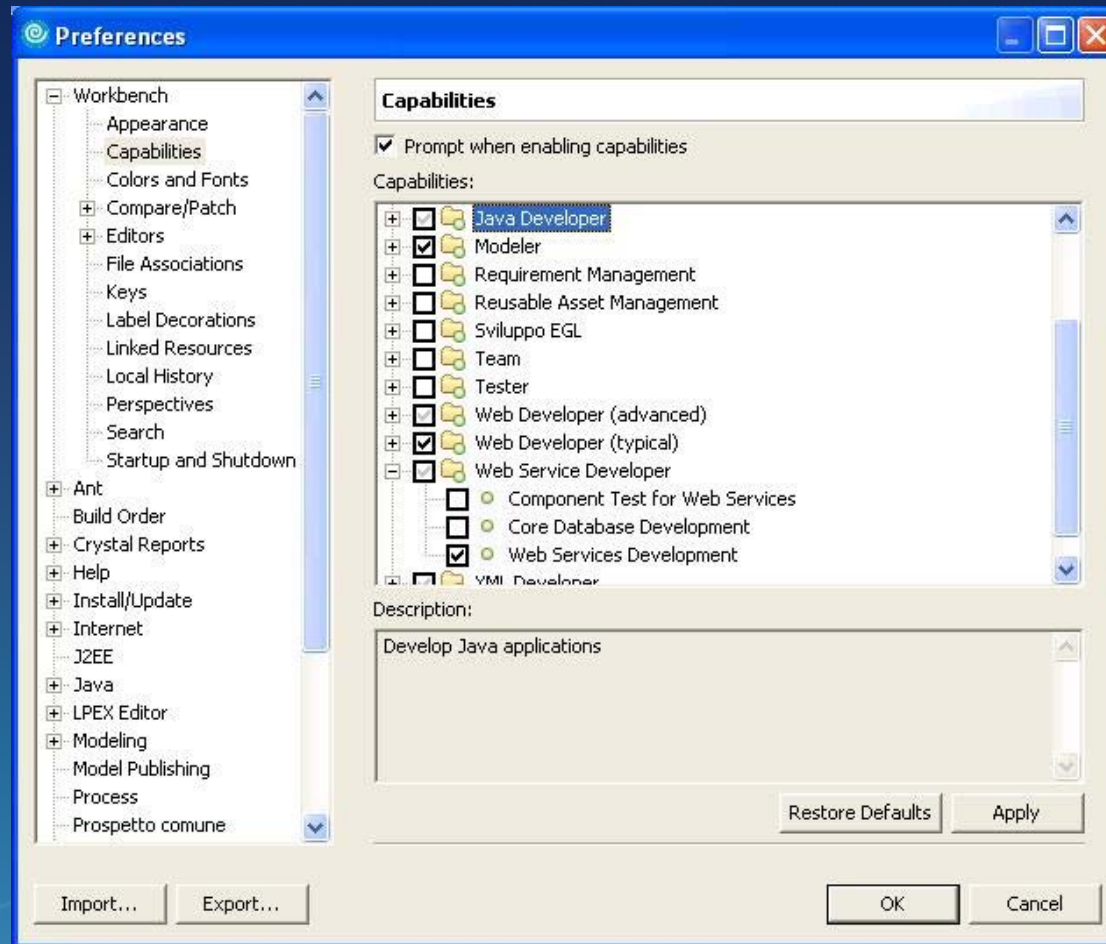
        //Setting ConfBean
        ConfBean confBean = new ConfBean();
        confBean.setName("ONO2");
        confBean.setUser("ACGFORJAVA");
        confBean.setPassword("ACGFORJAVA");
        confBean.setSi("ACGEXPRESS");

        //Setting ParameterBean
```

The screenshot shows the IBM Rational Software Development Platform interface. The main editor window displays the source code for the class `ONO2ActionTest` in the `test` package. The code includes an import for `java.util.Date` and a `main` method that demonstrates the instantiation and configuration of a `ConfBean` object. The IDE's Project Explorer on the left shows a project structure with various resources. The Console window at the bottom shows the execution output, including a terminated message and the output of a `Table` object.

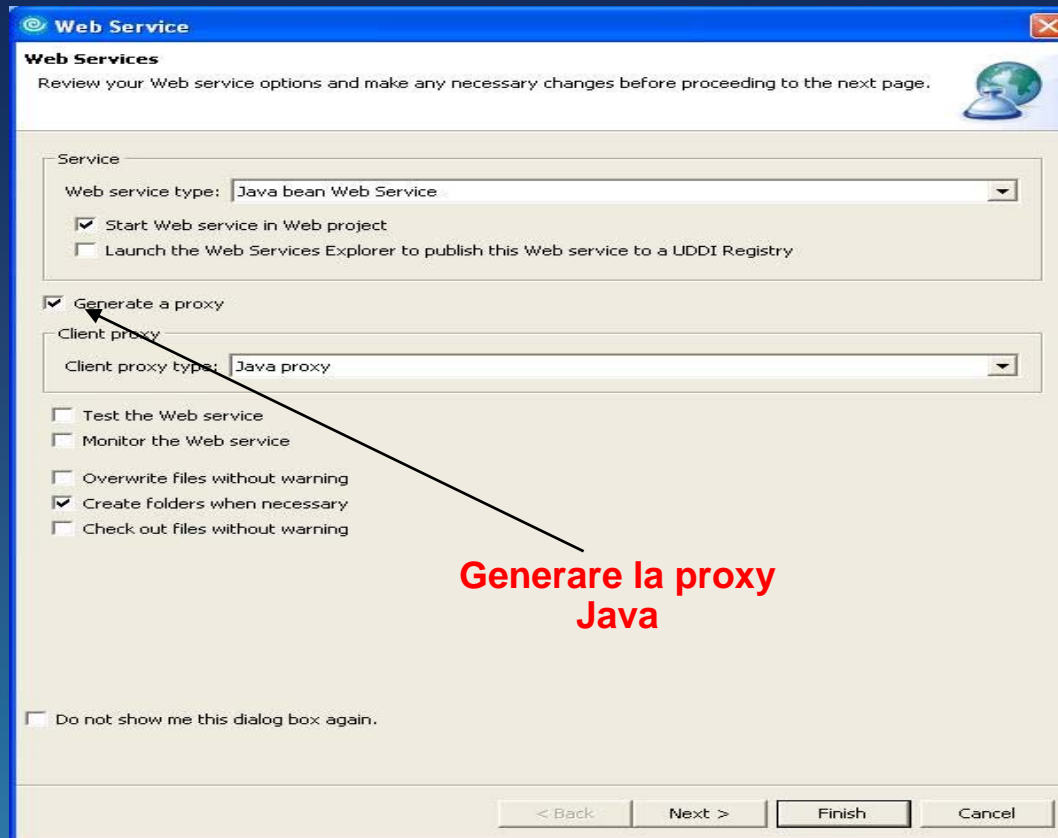
Il risultato della generazione del codice Java può essere verificato scrivendo una classe di test

Lo sviluppo Java Web Service



**Abilitare la
generazione dei WS
dalla voce
“Workbench >
Capabilities” della
finestra delle
Preferenze**

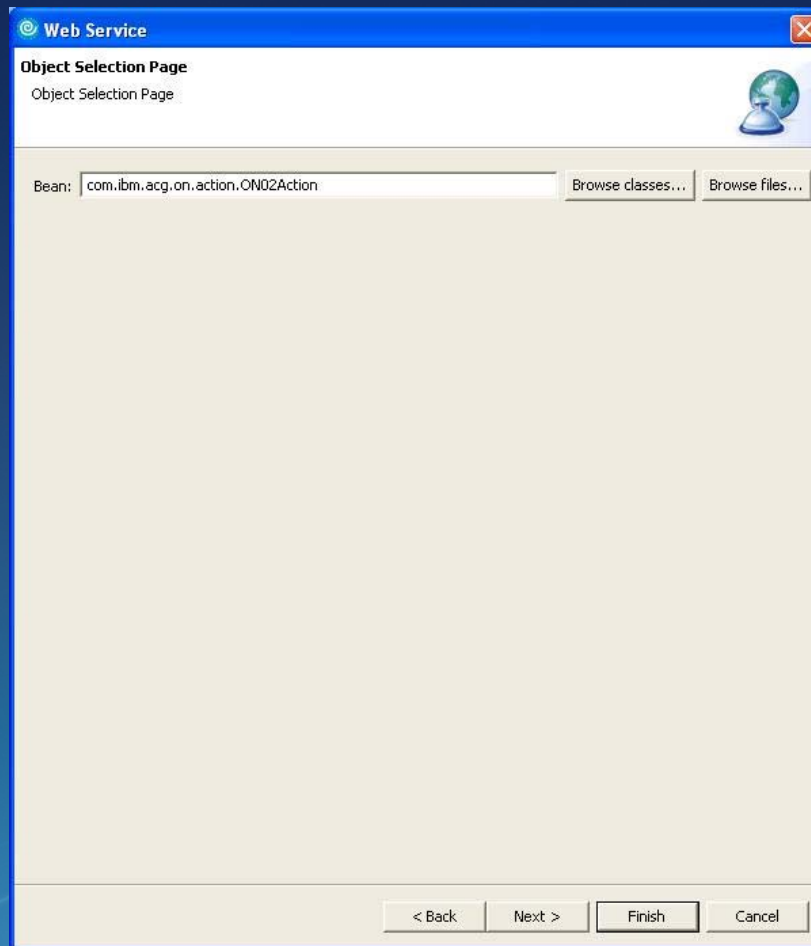
Lo sviluppo Java Web Service



**Creare il Web
service a partire
dalla classe Java
che definisce
l'azione WS**

**Avviare
preliminarmente il
server WAS nel
test environment**

Lo sviluppo Java Web Service



Specificare il nome dell'azione cui si riferisce il WS

Lo sviluppo Java Web Service

Web Service

Service Deployment Configuration

Choose from the list of runtime protocols and deployment servers, or use the default settings.

Select the service project and the EAR project with which you want it to be associated. If an EAR or project does not exist or is currently unassociated, it will be created and associated as required when you click Next.

Server-Side Deployment Selection:

Web service runtime: IBM WebSphere
Server: WebSphere Application Server v6.0
J2EE version: 1.4
Edit...

Service project: acgforjava
EAR project: acgforjavaEAR

Client-Side Environment Selection:

Web service runtime: IBM WebSphere
Server: WebSphere Application Server v6.0
J2EE version: 1.4
Edit...

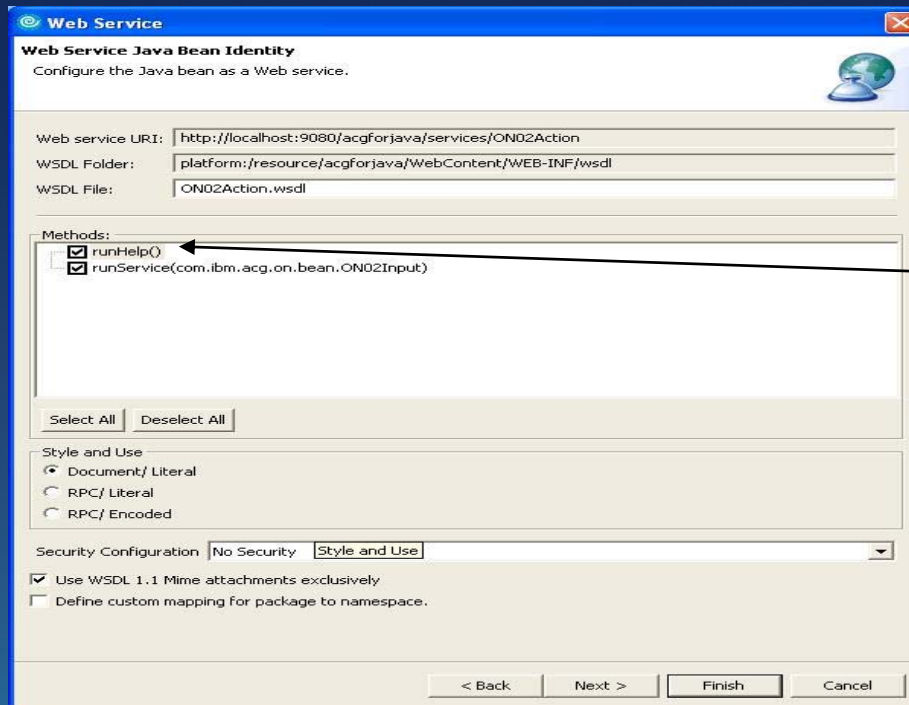
Client type: Web
Client project: acgforjavaClient
EAR project: acgforjavaClientEAR

The client project and EAR project will be created and associated with one another.

< Back Next > Finish Cancel

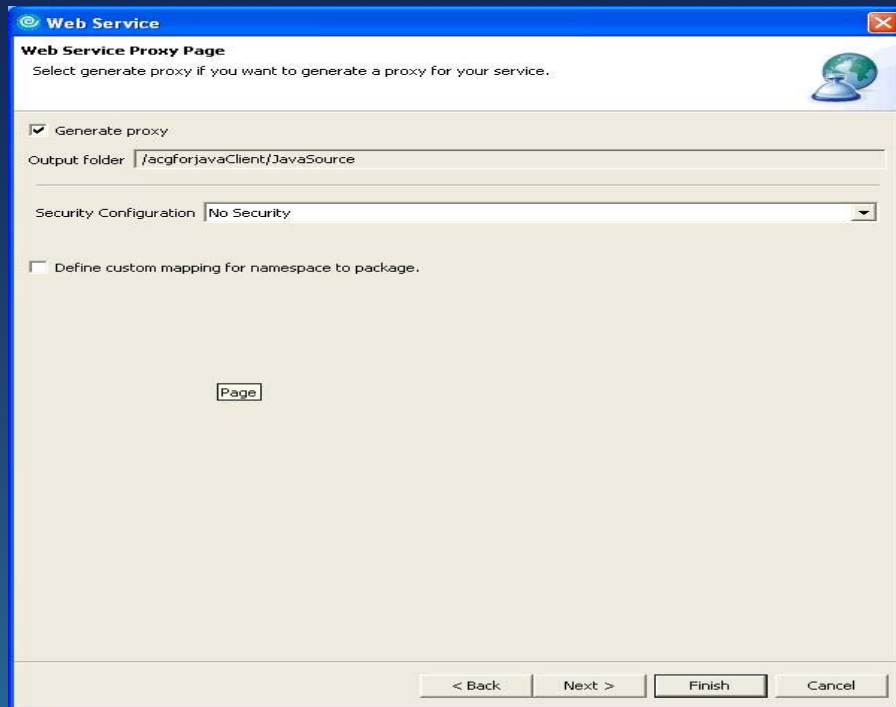
Specificare i server target e client

Lo sviluppo Java Web Service



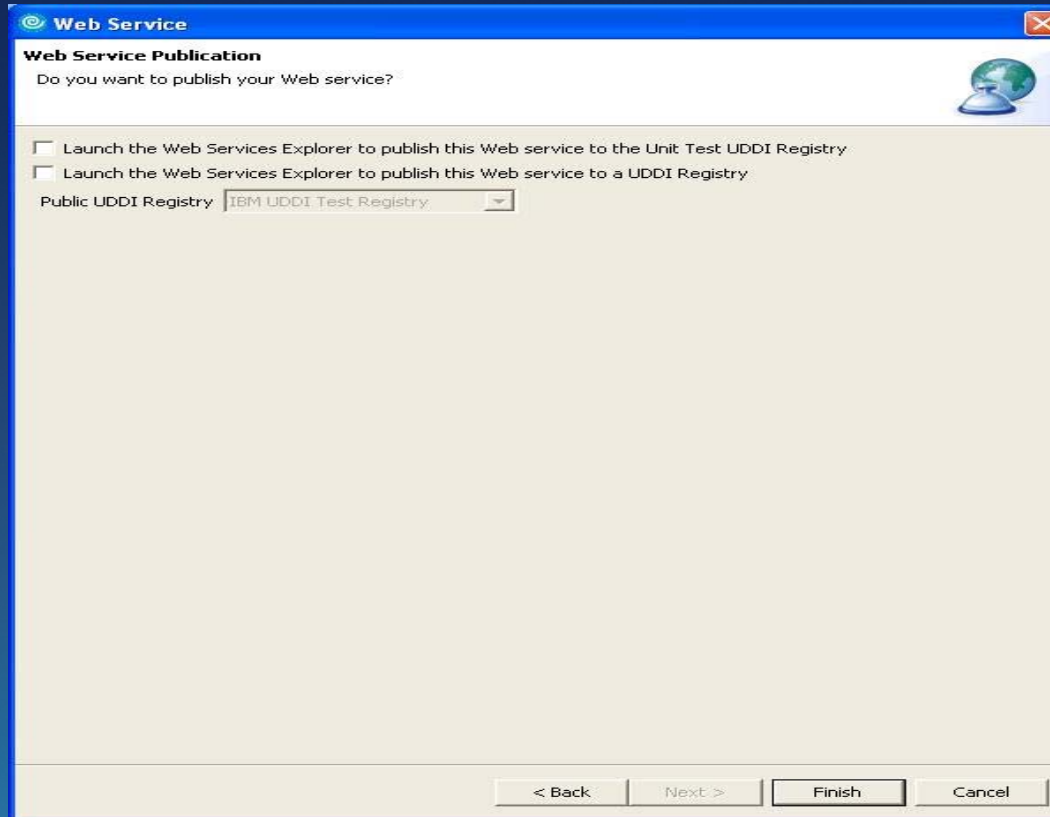
Selezionare i metodi da esporre nel Web Service

Lo sviluppo Java Web Service



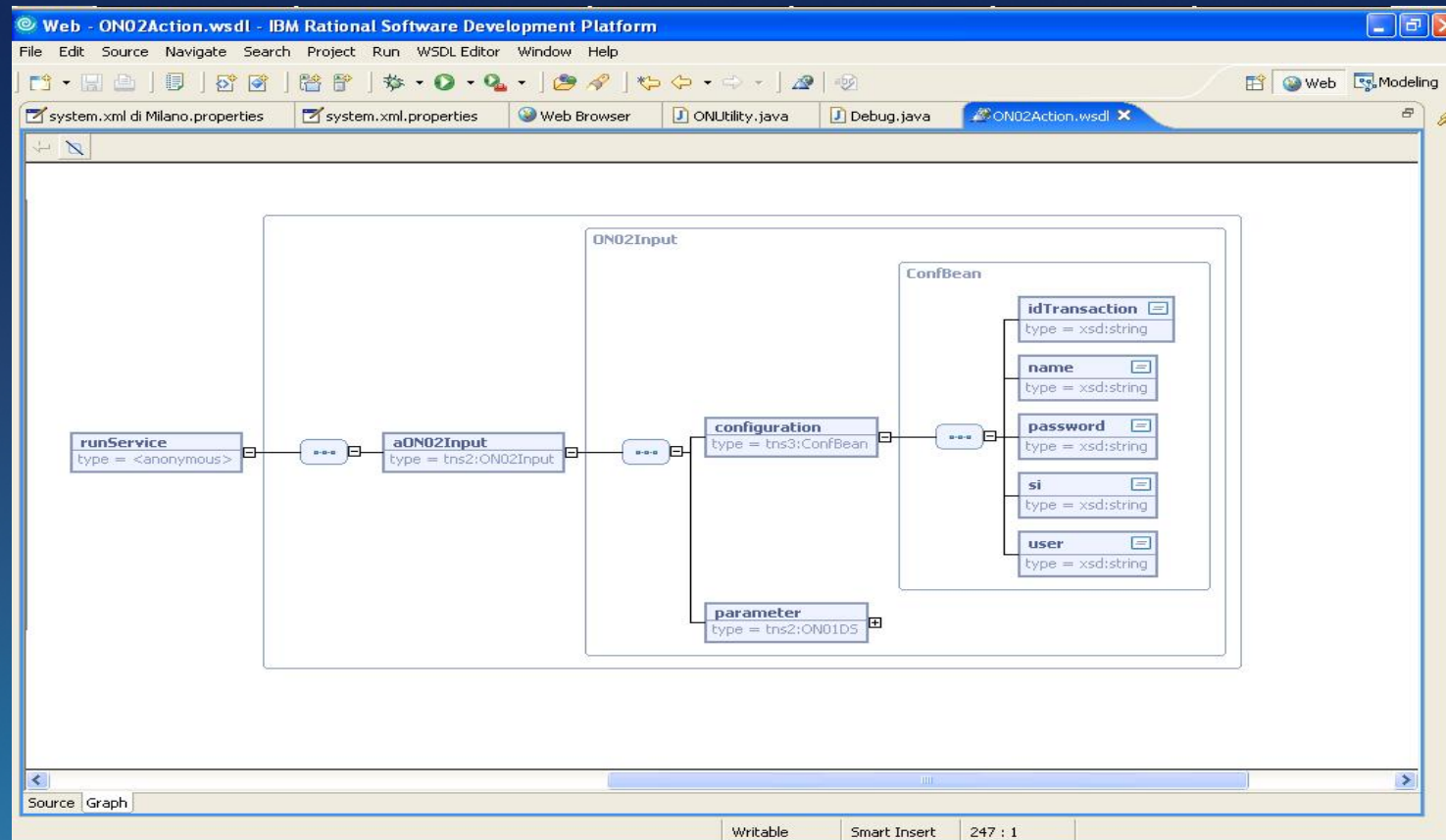
Non modificare i default

Lo sviluppo Java Web Service

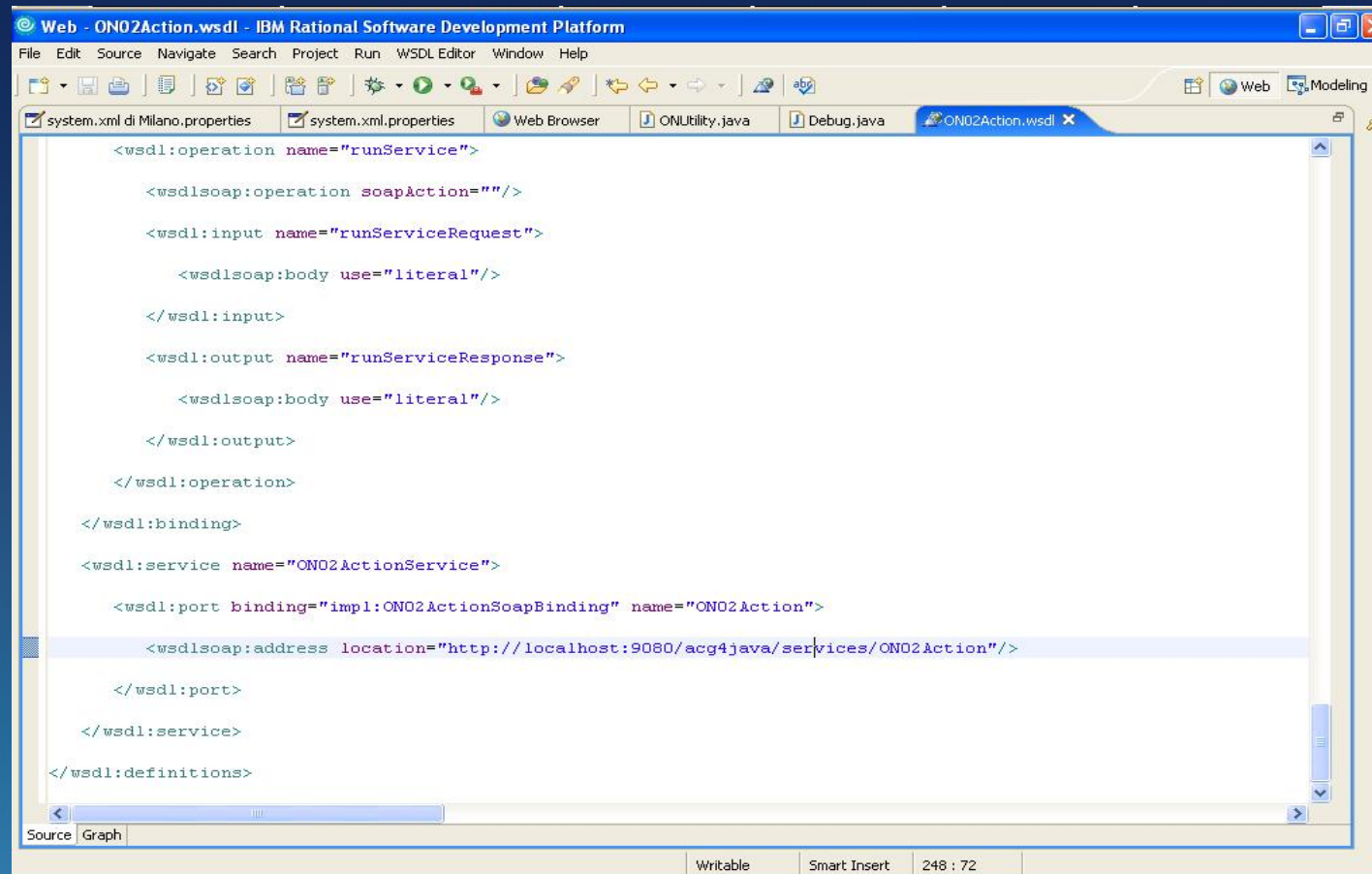


Al termine dell'operazione (pressione del tasto Finish) vengono create le classi di serializzazione/deserializzazione ed helper, nonché un progetto client del WS

Programming Model & development tools



Programming Model & development tools



The screenshot shows the IBM Rational Software Development Platform (RSDP) interface. The main window displays the WSDL editor for 'ONO2Action.wsdl'. The code is as follows:

```
<wsdl:operation name="runService">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="runServiceRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="runServiceResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ONO2ActionService">
  <wsdl:port binding="impl:ONO2ActionSoapBinding" name="ONO2Action">
    <wsdlsoap:address location="http://localhost:9080/acg4java/services/ONO2Action"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

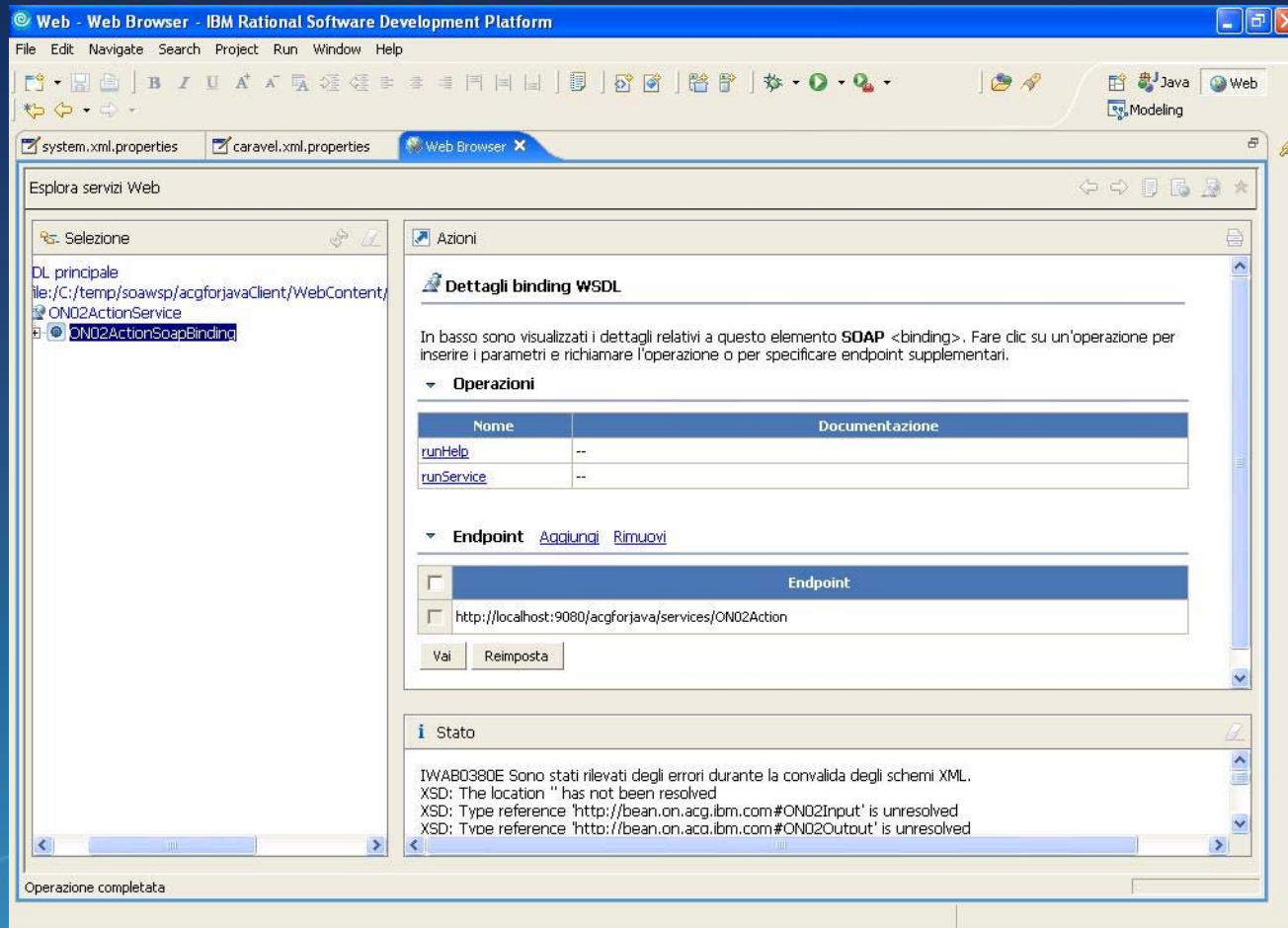
The status bar at the bottom of the editor shows 'Source' selected, 'Graph' as an alternative view, and 'Writable' and 'Smart Insert' modes. The cursor is positioned at line 248, column 72.

Lo sviluppo Java Web Service



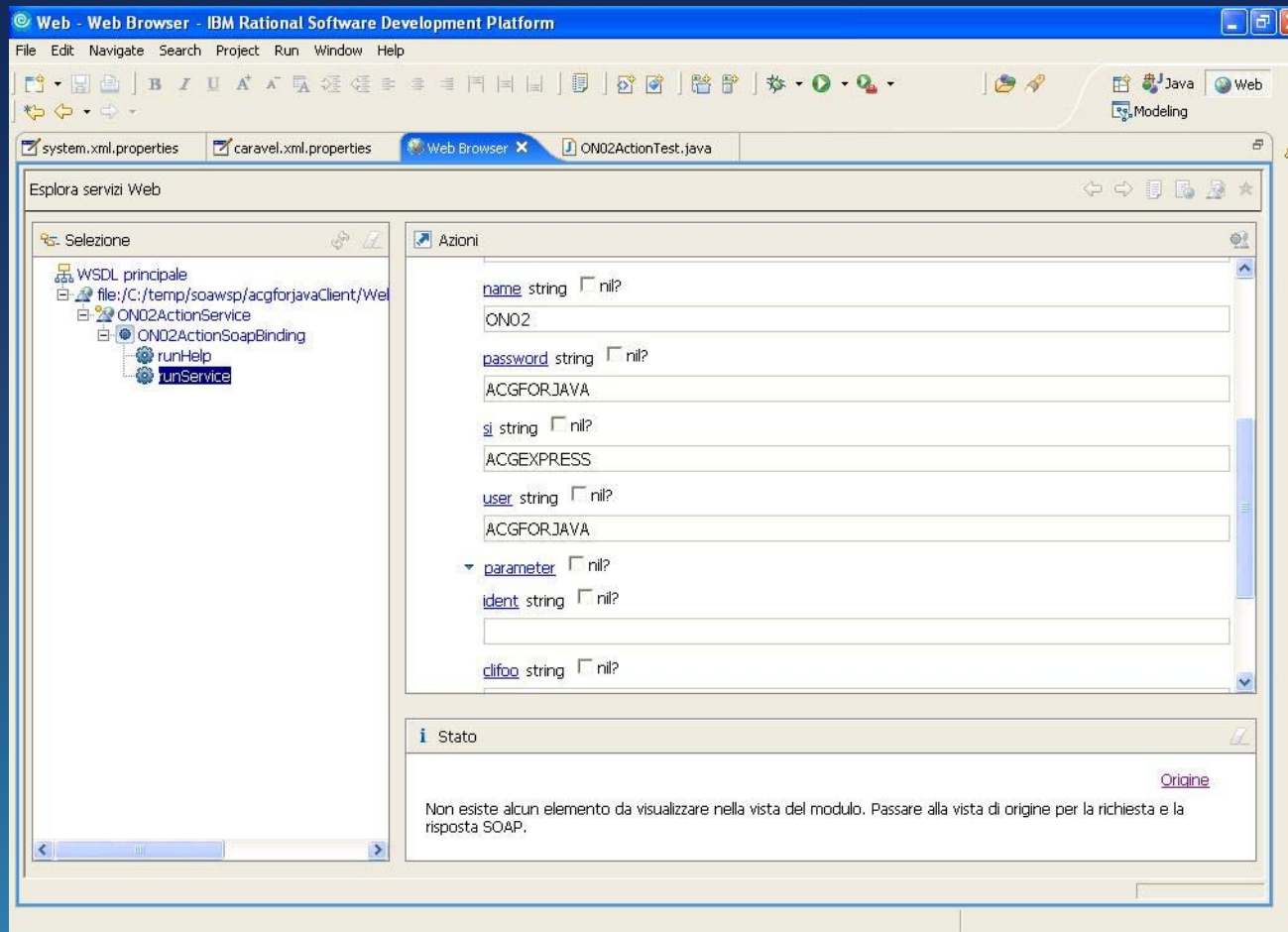
Si verifichi che il
WS è attivo
invocandolo da
browser

Lo sviluppo Java Web Service



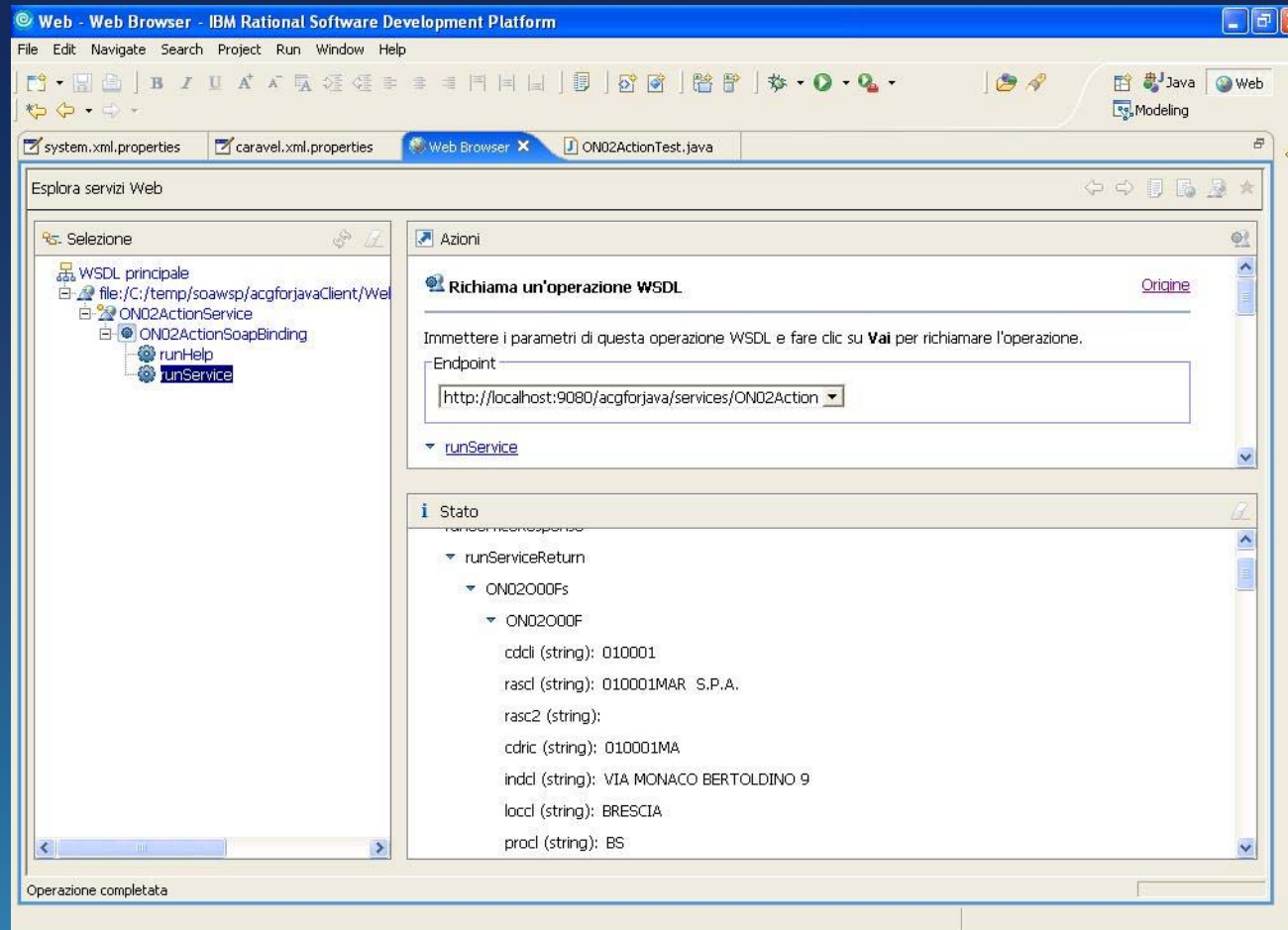
E' possibile utilizzare il wizard di test del WS (attivabile sul file WSDL) per verificare il WS

Lo sviluppo Java Web Service



Si inseriscano opportunamente i parametri per l'invocazione del servizio

Lo sviluppo Java Web Service



L'invocazione del servizio produce un file XML di risposta che viene visualizzato in una pagina di risultato

Lo sviluppo Java Web Service

The screenshot displays the IBM Rational Software Development Platform's web browser window. The address bar shows the URL: `http://localhost:9080/acgforjavaClient/sampleON02ActionProxy/TestClient.jsp`. The interface is divided into three main sections:

- Methods:** A list of available methods including `useJNDI(boolean)`, `getEndpoint()`, `setEndpoint(java.lang.String)`, `getON02Action()`, `runHelp()`, and `runService(com.ibm.acg.on.bean.ON02ActionInput)`.
- Inputs:** A form for the `aON02Input` parameter. Fields include:
 - `fcfoo:` (text input with value 'C')
 - `clfoo:` (text input with value '020010')
 - `ident:` (text input)
 - configuration:**
 - `user:` (text input with value 'ACGFORJAVA')
 - `si:` (text input with value 'ACGEXPRESS')
 - `password:` (text input with value 'ACGFORJAVA')
 - `name:` (text input with value 'ON02')
 - `idTransaction:` (text input)Buttons for `Invoke` and `Clear` are located below the input fields.
- Result:** A section for the response, showing `returnp:` and `oN02O00Fs: []`.

At the bottom left of the browser window, the status bar indicates "Operazione completata".

E' possibile creare uno scheletro di client web dell'applicazione utilizzando il wizard di RAD "Create test client"

Lo sviluppo Java Web Service: Step 1 Generazione Prg JAVA

```

ACGON_SAC/000SDS
SEU=>
***** Inizio dati *****
REF(=LIBL/DIZACGV3)
0001.00 00010
0002.00 00020A      R ON01R
0003.00 00030A* ID transazione
0004.00 00040A      IDENT          35A
0005.00 00050A* Codice cliente/nitore
0006.00 00060A      CLIF          R      REFFLD(CDCLI)
0007.00 00070A* Flag cliente/fo       tore (val. ammissibili C/F)
0008.00 00080A      FCLFO          R      REFFLD(FCLFO)
***** Fine dati *****
F3=Fine F5=Rivisual F9=Duplica F10=Copio F11=Alt+esce F12=Revisal
F16=Ripetiz. ricerca F24=Altri ta
  
```

Programma ACG in RPG convertito in Java RPG

```

package generator;

import com.ibm.ecg.on.generator.ACGGenerator;

public class ONXX {

    public static void main(String[] args) {

        String BI = "11_01esac";
        String user = "TTTTT_ACCP";
        String password = "PASSWORD_ACCP";
        String lib_obj = "LIBRER13_PPG_ACG00A";
        String lib_date = "LIBRER13_PPG_ACG00A";
        String ds = "DS_AZIONE_SGA";
        String input = "FF_01_INPUT_AZIONE_SGA";
        String error = "FF_01_ERROR_AZIONE_SGA";
        String url = "XML_DESCRITTURE_AZIONE_SGA";

        ACGGenerator g = new ACGGenerator();
        g.createDS(BI, user, password, lib_obj, ds);
        g.createTable(BI, user, password, lib_date, input);
        g.createTable(BI, user, password, lib_date, error);
        g.createAction(url);
    }
}
  
```

2

Classe java e file XML per la generazione del codice Java

```

package com.ibm.ecg.on.action;

import java.sql.ResultSet;
import java.util.Vector;
import com.ibm.ecg.on.util.DataAccess;
import com.ibm.ecg.on.util.BaseException;
import com.ibm.ecg.on.connector.*;
import com.ibm.ecg.on.bean.*;

public class ON02Action {

    private String host = JavaUtil.ResourceBundle.getBundle("com.ibm.ecg.on.resource.ecg.on").getString("host");

    protected void runWrite(DataAccess da, String idTransaction, ON02Input aON02Input) throws Exception {}

    protected ON02Output runRead(DataAccess da, String idTransaction, ON02Output aON02Output) throws Exception {}

    public ON02Output runService(ON02Input aON02Input) throws Exception {}

    public void runHelp() {}

    protected void runDelete(DataAccess da, String idTransaction) throws Exception {}

    protected String createEjeba(DataAccess da, String idTransaction, ON02Input aON02Input) {}

    private String newIdTransaction() {}
}
  
```

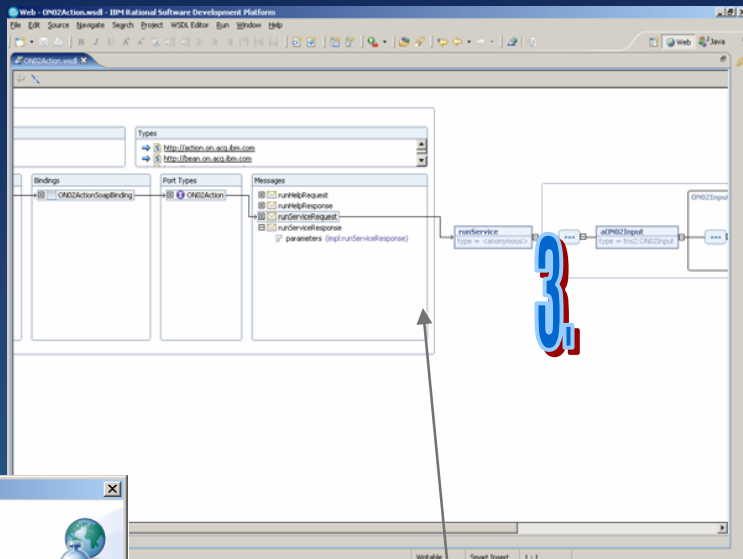
3

Action Java

Lo sviluppo Java Web Service: Step2 Generazione del Web Service

```
package com.ibm.ecg.on.action;  
  
import java.sql.ResultSet;  
import java.util.Vector;  
import com.ibm.ecg.on.util.DataAccess;  
import com.ibm.ecg.on.util.BaseException;  
import com.ibm.ecg.on.connector.*;  
import com.ibm.ecg.on.bean.*;  
  
public class OR02Action {  
    private String host = java.util.ResourceBundle.getBundle("com.ibm.ecg.on.resources.ecg0n").getString("host");  
  
    protected void runWrite(DataAccess da,String idTransaction,OR02Input aOR02Input) throws Exception {}  
    protected OR02Output runRead(DataAccess da,String idTransaction,OR02Output aOR02Output) throws Exception {}  
    public OR02Output runService(OR02Input aOR02Input) throws Exception {}  
    public void runHelp() {}  
  
    protected void runDelete(DataAccess da,String idTransaction) throws Exception {}  
    protected String createSpjba(DataAccess da,String idTransaction,OR02Input aOR02Input){}  
  
    private String newIdTransaction() {}  
}
```

Action Java



Web Service

Web Service
Review your Web service options and make any necessary changes before proceeding to the next page.

Service
Web service type: Java bean Web Service
 Start Web service in Web project
 Launch the Web Services Explorer to publish this Web service to a UDDI Registry

Generate a proxy

Client proxy
Client proxy type: Java proxy

Test the Web service
 Monitor the Web service
 Overwrite files without warning
 Create folders when necessary
 Check out files without warning

< Back Next > Finish Cancel

Wizard RATIONAL di Generazione del WS

Programming Model & development tools

Programming Model & development tools

Lo sviluppo di un Web Service richiede:

1. la definizione dei file database di input e output
2. la scrittura di un programma da richiamare che legge i dati da uno o più file di input (opzionale) e scrive su uno o più file di output
3. la scrittura di un file XML di descrizione del programma, dell'input e dei file di I/O
4. la creazione di un'azione di Modulo Base associata al programma
5. la creazione dei bean java e di una classe "action" che invoca l'azione ACG
6. la creazione del Web service tramite RAD

Lo sviluppo di WVS: Step3 Integrazione dei WVS nella web application

L'integrazione dei WS creati nella web application installata su WAS richiede:

- ❑ la copia dalla cartella "WebContent" di RAD alle cartelle di installazione della web application in WAS su iSeries (sia sotto installedApps che sotto config) di:
 - la cartella WEB-INF\wsdl con il suo contenuto
 - il file ibm-webservices-bnd.xmi
 - il file ibm-webservices-ext.xmi
 - il file webservices.xmi
 - i file xxxxActionMapping
- ❑ La copia delle classi java action e bean con i relativi bean ser/deser ed helper (contenute rispettivamente nei package com.ibm.acg.on.action e com.ibm.acg.on.bean) nella cartella WEB-INF\classes della web application sotto installedApps (in alternativa possono essere impacchttate in un jar file da copiare in WEB-INF\lib)

SOA resources

SOA resources:

developerWorks è il repository IBM's dei documenti tecnici per gli sviluppatori:

- documenti e codice scaricabile gratuitamente
- tutto è costruito su standard aperti e condivisi

Software Evaluation Kit :

- è possibile scaricare l'ultima versione di SDK da ibm.com/developerworks/
- è possibile scaricare prodotti da ibm.com/developerworks/downloads.

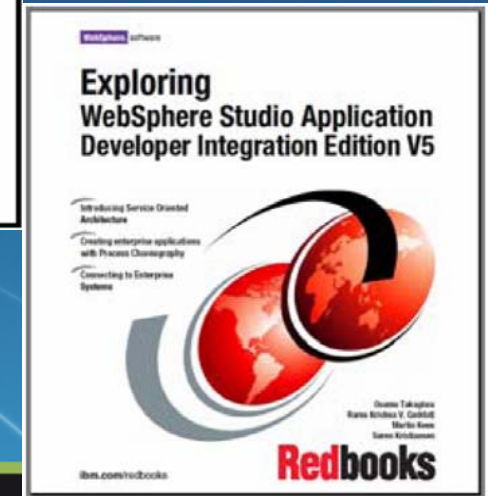
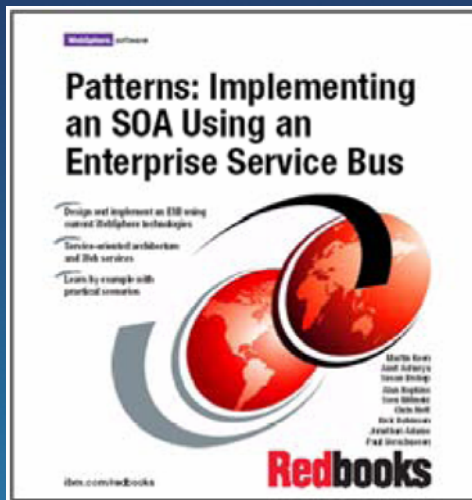
WebSphere Web services zone
(www.ibm.com/developerWorks/websphere/zones/webservices/)

articoli su Web services articles
(www.ibm.com/developerworks/webservices)

SOA resources: Web services books

The dW online bookstore online bookstore
(devworks.krcinfo.com)

Redbooks (ibm.com/redbooks)

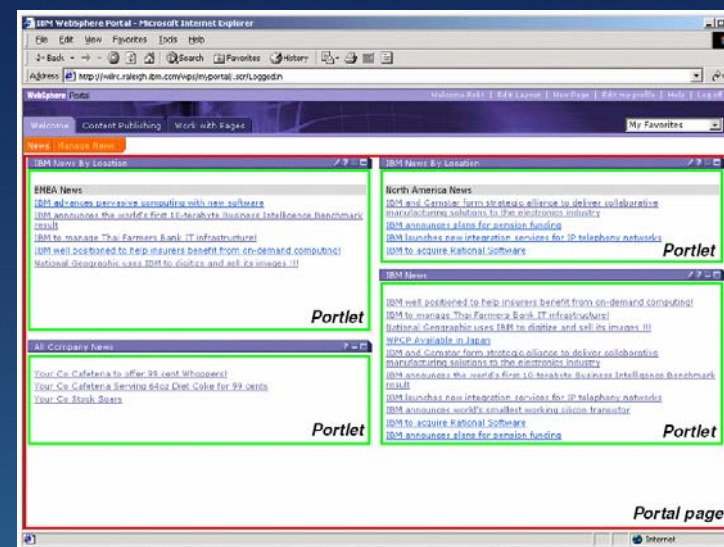


Lo sviluppo Portlet

Un portlet è un singola applicazione residente all'interno di una pagina del portale.

E' rappresentata da una finestra che presenta delle informazioni o permette delle funzionalità.

Per lo sviluppatore sono delle applicazioni Java-based.



Le modalità di funzionamento di una IBM Portlet sono:

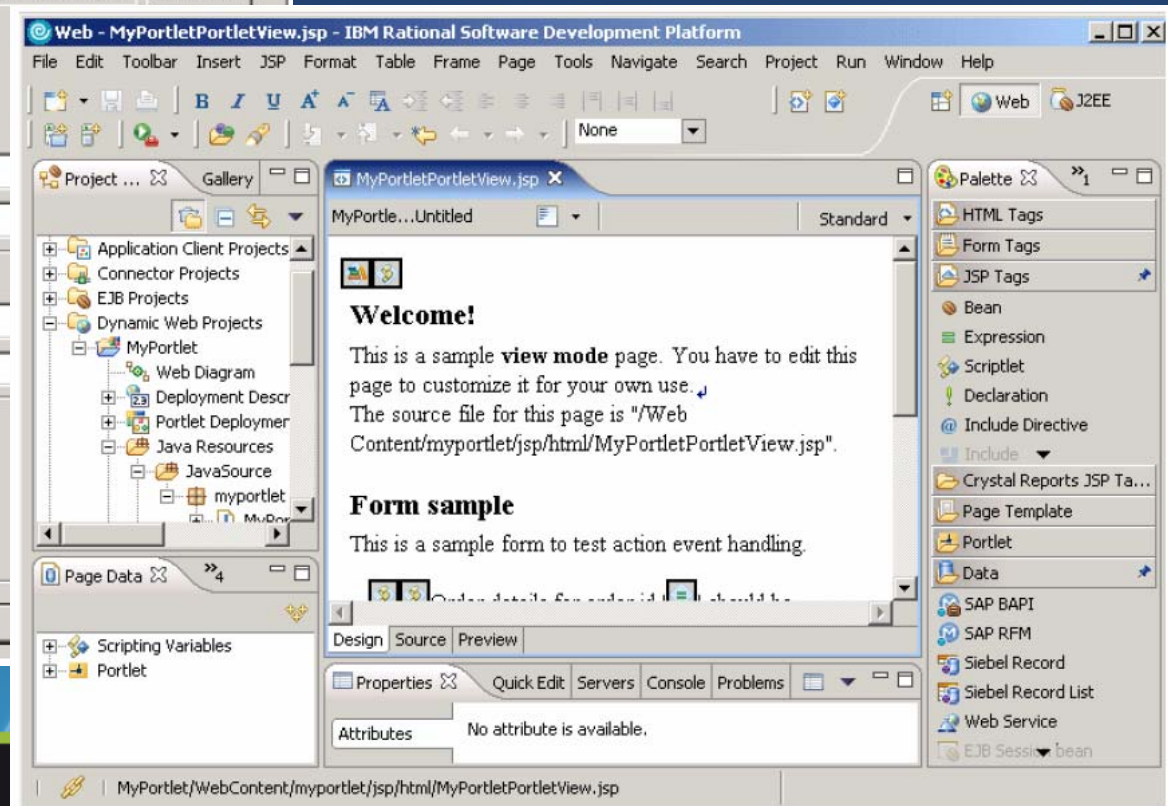
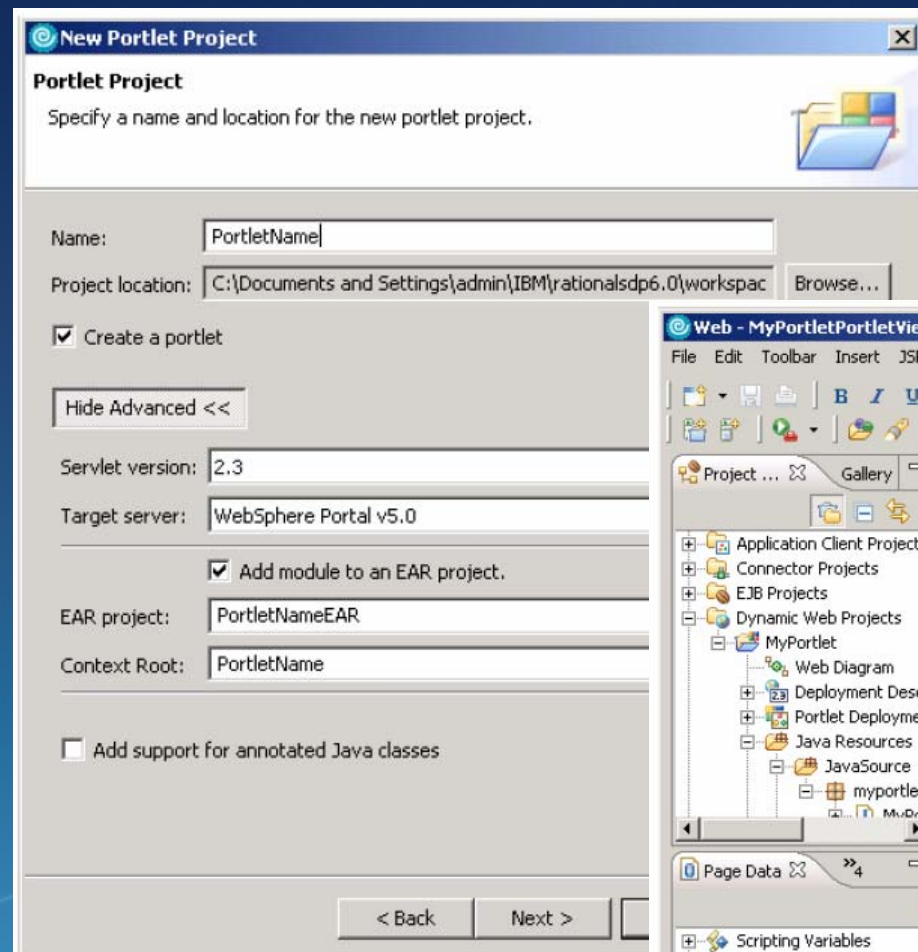
View - si fornisce il contenuto informativo all'utente

Edit - Consente all'utente di definire alcuni parametri di funzionamento

Help - Restituisce informazioni di supporto

Configure - Consente all'amministratore di definire alcuni parametri di funzionamento

Lo sviluppo Portlet : Step 1 Portlet Project wizard



Lo sviluppo Portlet : Step 2 Aggiunte al codice generato

```
package com.ibm.esg.on.portal;  
  
import java.io.IOException;  
  
public class ON02Portlet extends PortletAdapter implements ActionListener {  
    public void init(PortletConfig portletConfig) throws UnavailableException {  
        super.init(portletConfig);  
    }  
    public void doView(PortletRequest request, PortletResponse response) throws PortletException, IOException {}  
    public void actionPerformed(ActionEvent event) throws PortletException {}  
}
```

1

ONXXPortlet

Scheda Cliente	
Scheda del Cliente: 000180	
Ragione Sociale: AZIENDA CARTAFIA ITALIANA	
... Dettagli Cliente ...	
Indirizzo:	VIA CAPPPELLI 10
Località:	MILANO
Provincia:	MI
CAP:	20100
Partita IVA:	00470100150
Codice Fiscale:	
Numero Telefono:	
Numero Fax:	
Codice Modaltà di Pagamento:	A 10
Codice Aperta:	A02

Costruzione della Tabella di OUTPUT

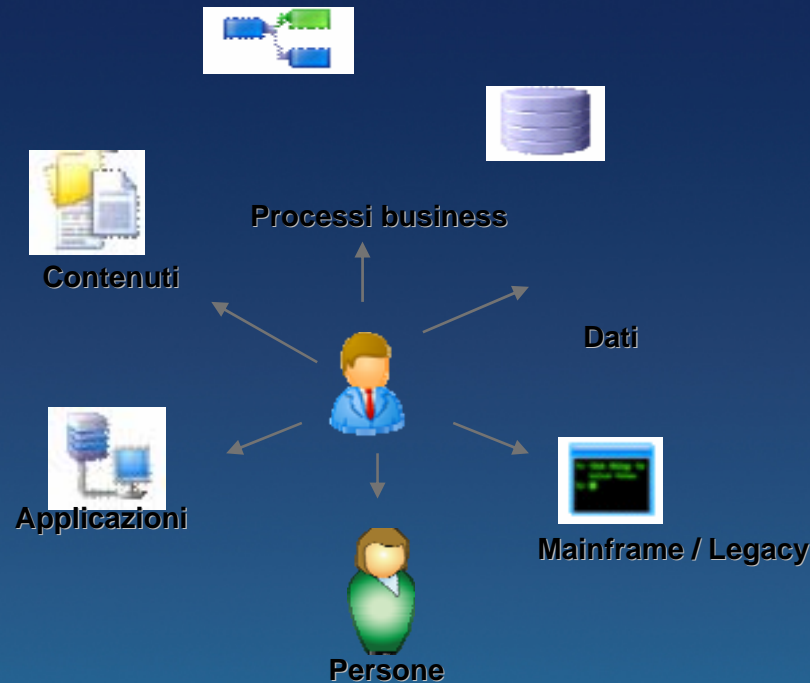
ACG SOA - Scheda Cliente

Codice cliente:

Avvia Annulla

Costruzione del Form di INPUT

Il punto di partenza è ...



L'utente interagisce con:

- Moltitudine di informazioni,
- Interfacce spesso molto diverse tra di loro

Problematiche:

- Gestione di diverse tipologie di client (manutenzione, aggiornamento software, diversi livelli di supporto)
- Usabilità
- Gestione password
- L'utente **DEVE** provvedere all'aggregazione di diverse informazioni per ottenere ciò di cui necessita

Il Portale IBM: 'the big picture'

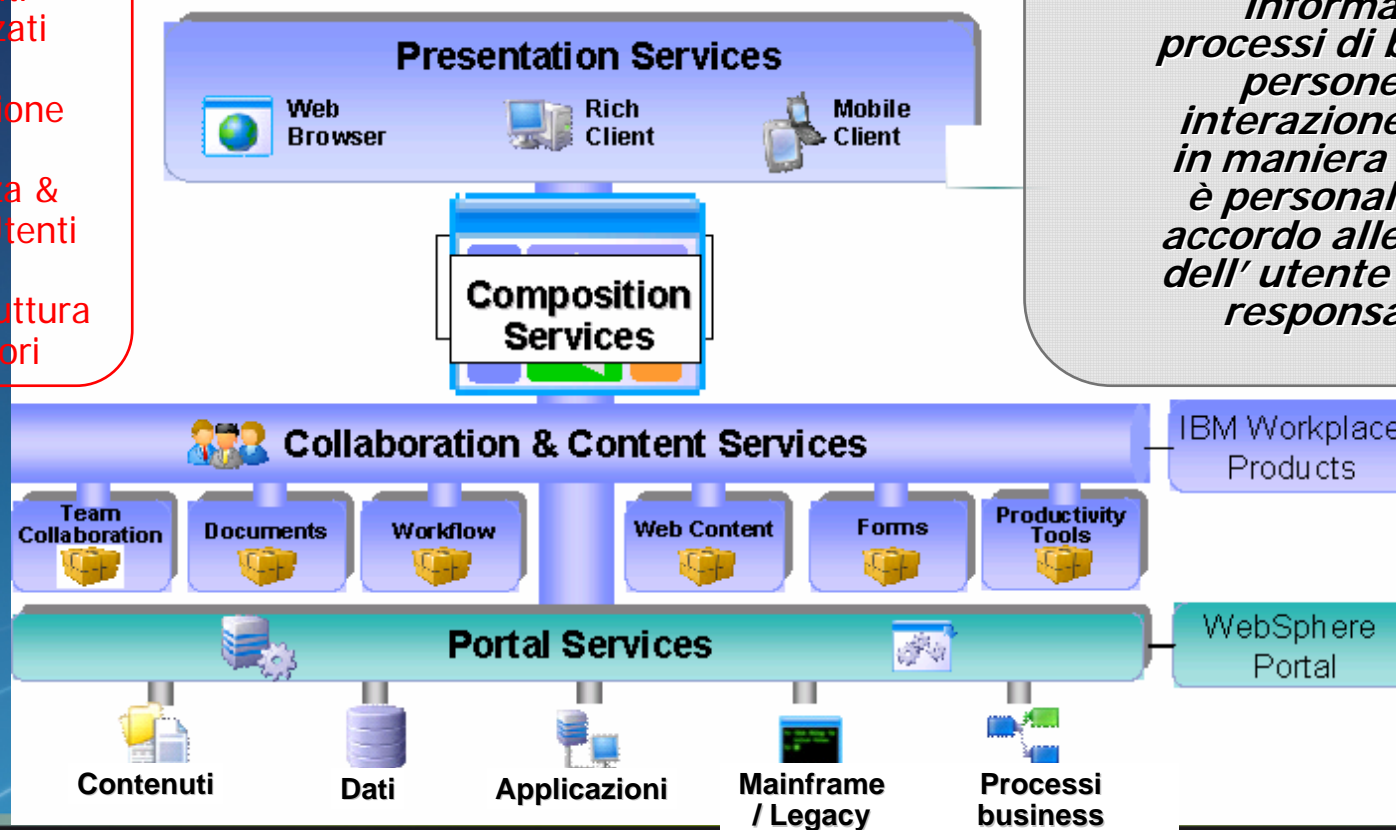
Servizi

- ❖ Aggregazione
- ❖ Navigazione
- ❖ Contenuti personalizzati
- ❖ Collaborazione
- ❖ SSO
- ❖ Sicurezza & Gestione Utenti (LDAP)
- ❖ Infrastruttura
- ❖ Connettori



Definizione

" Il Portale fornisce un unico punto di interazione con varie informazioni, processi di business e persone. Tale interazione avviene in maniera sicura ed è personalizzata in accordo alle esigenze dell' utente e alle sue responsabilità"

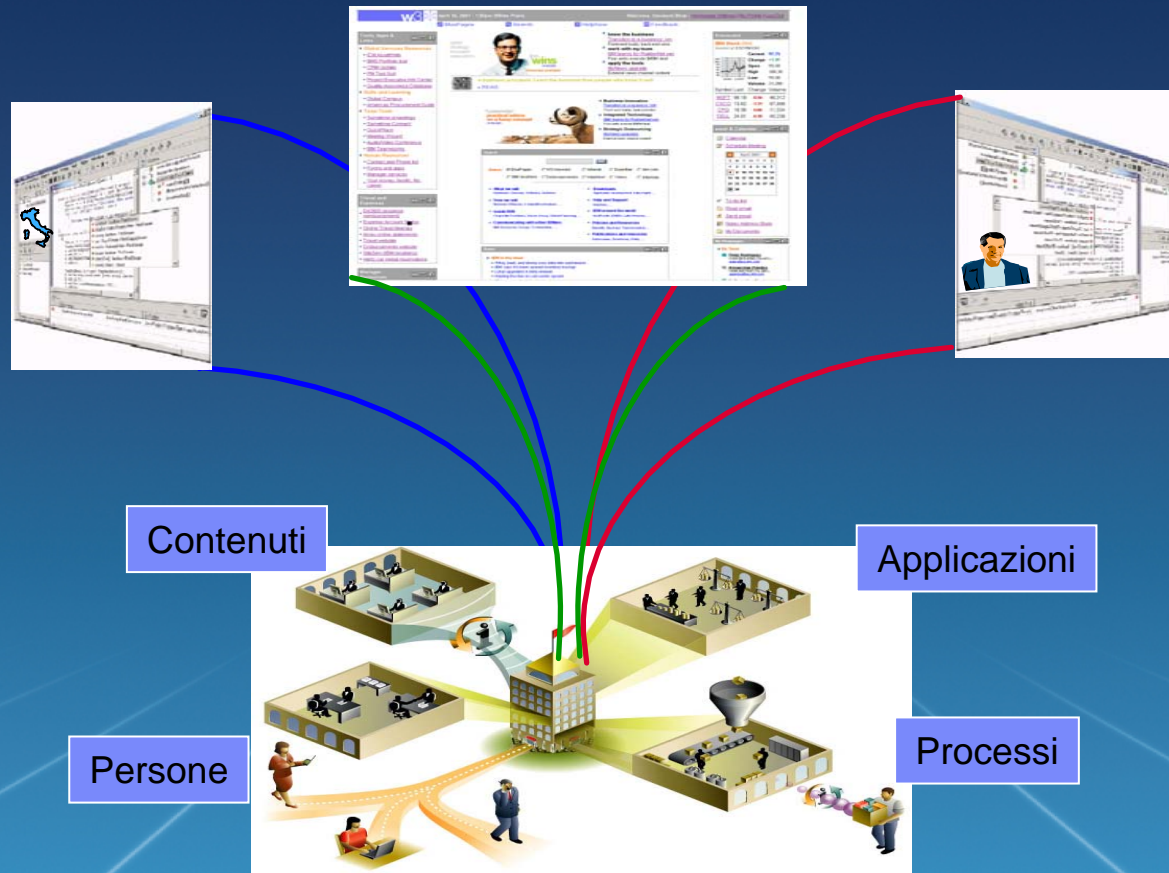


Il Portale IBM fornisce *Accesso Integrato* a persone, applicazioni, processi e informazioni

I portali **B2E** migliorano la produttività degli impiegati e consentono decisioni rapide

I portali **B2B** costruiscono relazioni tra i partner attraverso l'integrazione con i vostri processi di business

I portali **B2C** aumentano la loyalty dei clienti e il fatturato cross-sell



Una interfaccia di portale comune riduce i costi e si adatta alle necessità di cambiamento

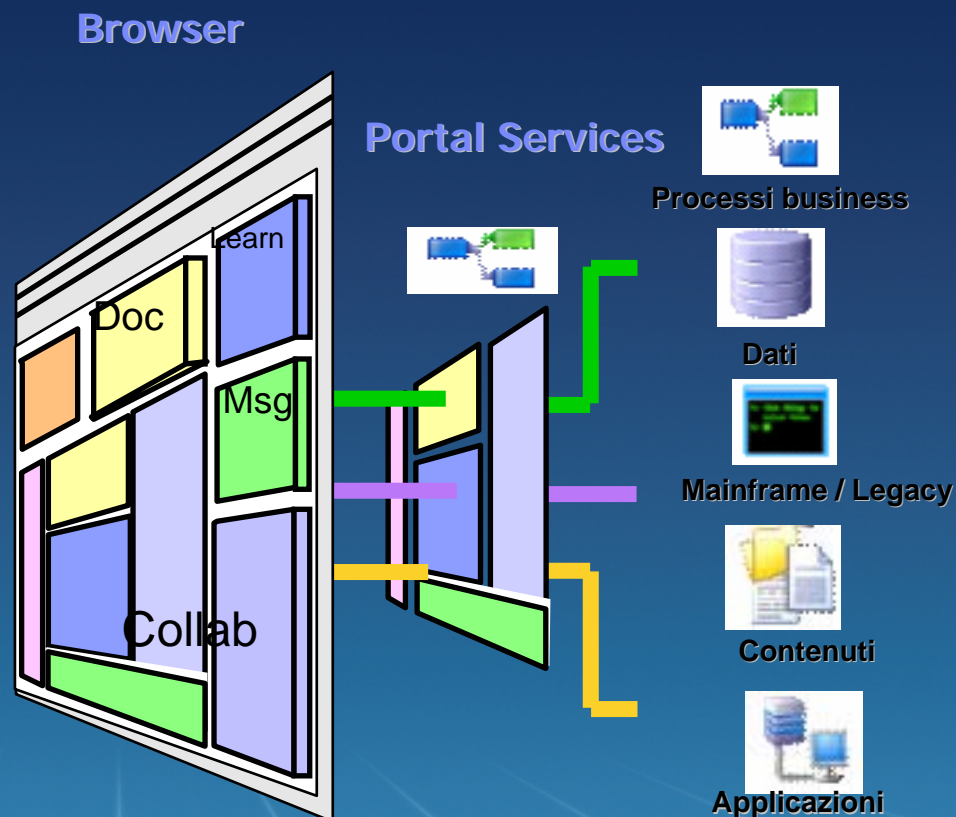
Il Portale IBM: aggregazione contenuti

Le pagine web servite all'utente sono il risultato di un servizio di aggregazione di diverse componenti

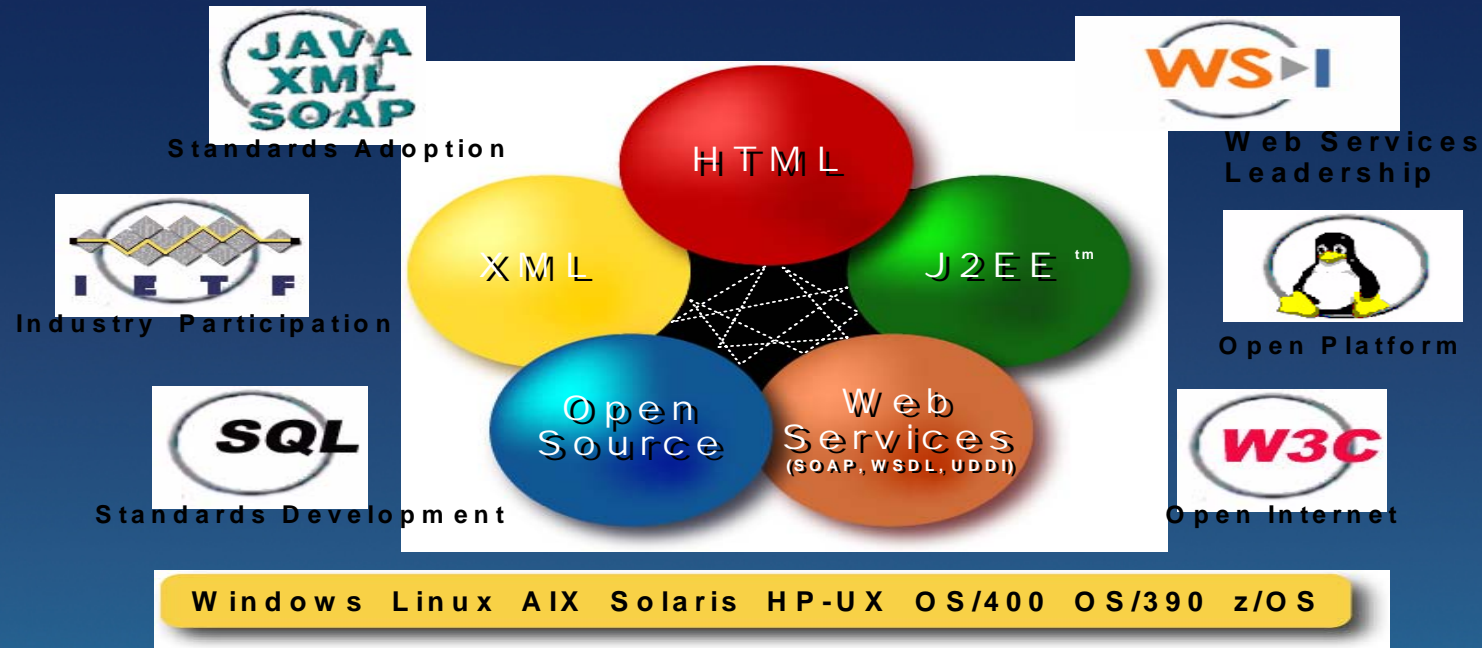
L'elemento base è la PORTLET

Principi di base della PORTLET

- Applicazione web
- Sviluppati indipendentemente
- Configurabili singolarmente
- Soggetti alle politiche di gestione della sicurezza



Prodotto basato su Standard di mercato



La Gestione degli Utenti

Immettere le proprie informazioni.

Queste informazioni consentono di personalizzare il contenuto visualizzato. Per ulteriori informazioni, consultare [Criteri di riservatezza](#).

Base	Internazionale
* ID utente: <input type="text"/>	Lingua preferita: <input type="text" value="-nessuna selezione-"/>
* Password: <input type="text"/>	Formato internazionale preferito: <input type="text" value="-nessuna selezione-"/>
* Conferma password: <input type="text"/>	Fuso orario: <input type="text" value="-nessuna selezione-"/>
* Nome: <input type="text"/>	Tipo di diario preferito: <input type="text" value="-nessuna selezione-"/>
* Cognome: <input type="text"/>	Primo giorno della settimana: <input type="text" value="-nessuna selezione-"/>
* E-mail: <input type="text"/>	Primo giorno lavorativo della settimana: <input type="text" value="-nessuna selezione-"/>

Configurazione ACG

Utente <input type="text"/>	Password <input type="text"/>
Codice Agente <input type="text"/>	Codice Capo Area <input type="text"/>
Codice Cliente <input type="text"/>	Codice Fornitore <input type="text"/>

*Campo obbligatorio

Estensione x ACG

La Gestione degli Utenti: Assegnazione dei diritti del Portale per utente/gruppo

The screenshot shows the IBM WebSphere Portal administration interface in Microsoft Internet Explorer. The browser title is "IBM WebSphere Portal - Microsoft Internet Explorer". The address bar shows the URL: http://9.87.236.119:9081/wps/myportal/tut/pj_s.7_0_A/7_0_55/.cmd/ad/.ar/sa.T626213c8/.c/6_0_30/.ce/7_0_58/.p/5_0_3K/.d/1?PC_7_0_58_wchidc. The page title is "WebSphere Portal". The main content area is titled "Autorizzazioni risorsa" (Resource Authorizations). Under the "Portlet" section, there is a search form with a dropdown menu set to "Il titolo contiene" (Contains title) and a "Ricerca" (Search) button. Below the search form, there is a table of portlets with an "Assegna accesso" (Assign access) column. The table lists various portlets, including "Microsoft Exchange Contacts", "Microsoft Exchange Calendar", "Microsoft Exchange Mail", "Lotus Notes View", "Domino Web Access", "ACGactionPortlet", "Ad banner", "Portlet ACGBusinessFunction", "Segnalibri", and "Calcola fondo pensione". A blue callout box with white text is overlaid on the right side of the table, pointing to the "Assegna accesso" column. The callout text reads: "Gli utenti potranno accedere solo ai portlet a cui sono stati abilitati" (Users will be able to access only the portlets to which they have been enabled).

Gli utenti potranno accedere solo ai portlet a cui sono stati abilitati

La Gestione delle pagine: Creazione Nuova Pagina

IBM WebSphere Portal - Microsoft Internet Explorer

File Modifica Visualizza Preferiti Strumenti ?

Indietro - - - - - Cerca Preferiti Multimedia - - - - - Vai Collegamenti

Ingresso http://9.87.236.119:9081/wps/myportal!ut/p/s_7_0_A/7_0_36/.cmd/ad/ar/sa.page.detail.view/c/6_0_36/ce/7_0_39/p/5_0_35/d/0?pageTitle=

WebSphere Portal

Proprietà

Nuova pagina: ACG OnDemand

Titolo:

Opzioni avanzate

- Questa pagina può essere aggiunta all'elenco Preferiti di un utente.
- Altre pagine possono condividere il contenuto di questa pagina.

Questa pagina è:

Il contenuto di una pagina con queste proprietà:

Questa pagina supporta (deve essere selezionato almeno un elemento):

WML cHTML HTML

Pagina che utilizza il contenuto da una pagina condivisa:

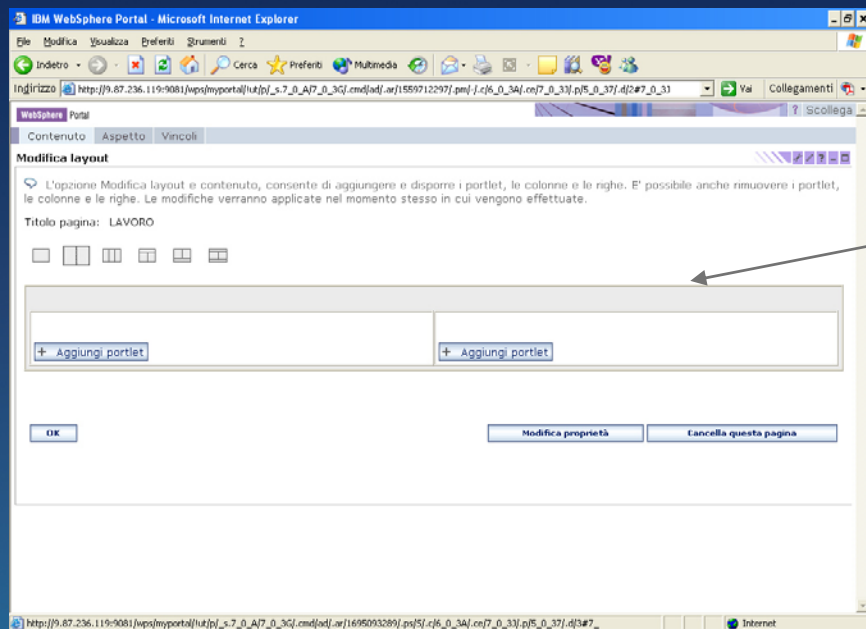
OK Annulla

Internet

L'amministratore definisce:

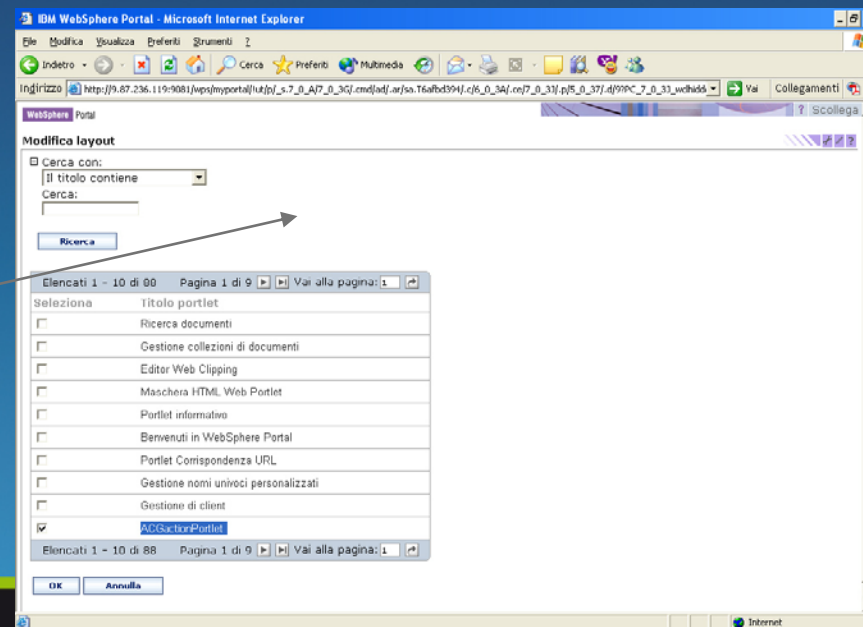
- Titolo
- Aspetto
- Mime

La Gestione delle pagine: Inserimento di Portlets

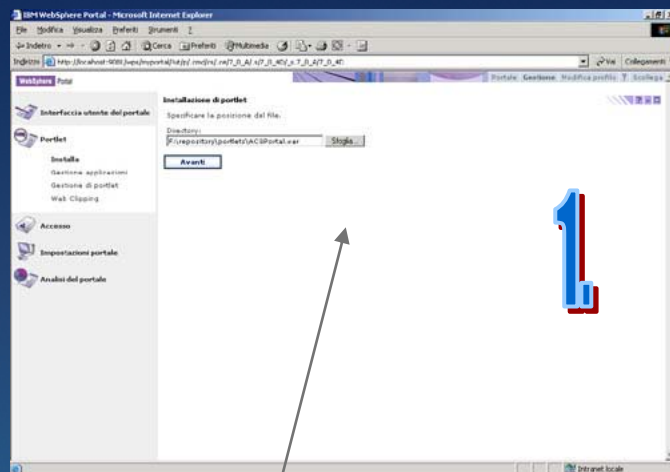


L'amministratore definisce la disposizione dei portlet

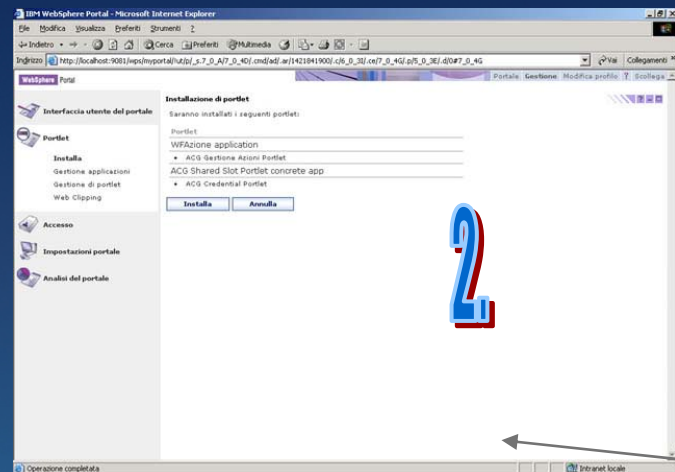
L'amministratore sceglie i portlet da inserire nella pagina



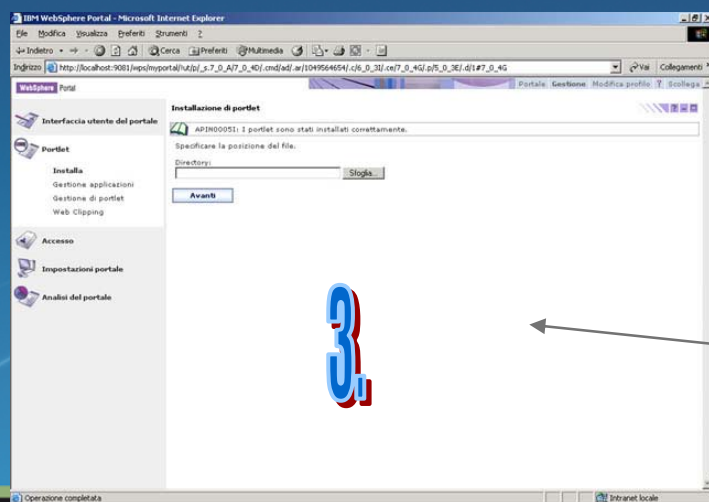
L'installazione dei Portlet



Scegliere il .war con la Web Application



Verificare la presenza dei Portlets



Corretta Installazione

SOA resources: IBM Software Group resources



alphaWorks Emerging Technologies Toolkit contains the latest code from IBM's R&D labs (ibm.com/alphaworks/tech/ettk)

**ETTK
(ibm.com/developerworks/webservices/demos/ettkalphaWorks)**

**alphaWorks Web services zone
(ibm.com/alphaWorks/webservices)**

SOA resources: Pubblicazioni ACG

A Scalable Service Oriented Framework for the Cooperation among e-Government Domains (GESTS Journal, International Transactions on Computer Science and Engineering - Vol.23, No. 01, November 30, 2005)

http://www.gests.org/online_index.html

A New Interoperability Framework for E-Government Services (IADIS International Conference, e-Society 2005, 27-30 June, Qawra, Malta) <http://www.iadis.org/es2005/>

A Flexible Mining Architecture for Providing New E-Knowledge Services (Proceedings of the 38th HICSS, Hawaii International Conference On System Sciences, January 3-6, 2005 Big Island, Hawaii, IEEE Computer Society Press)

<http://csdl.computer.org/comp/proceedings/hicss/2005/2268/03/2268toc.htm>

