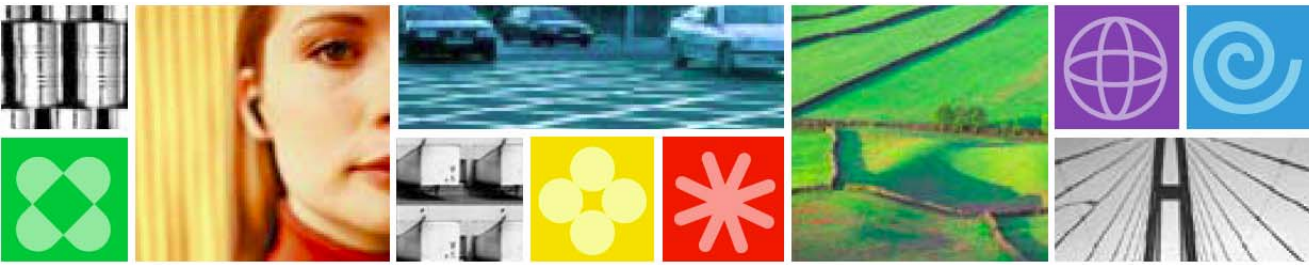
Let’s tie it all together and play in the sandbox



Updated Jan, 26, 2010

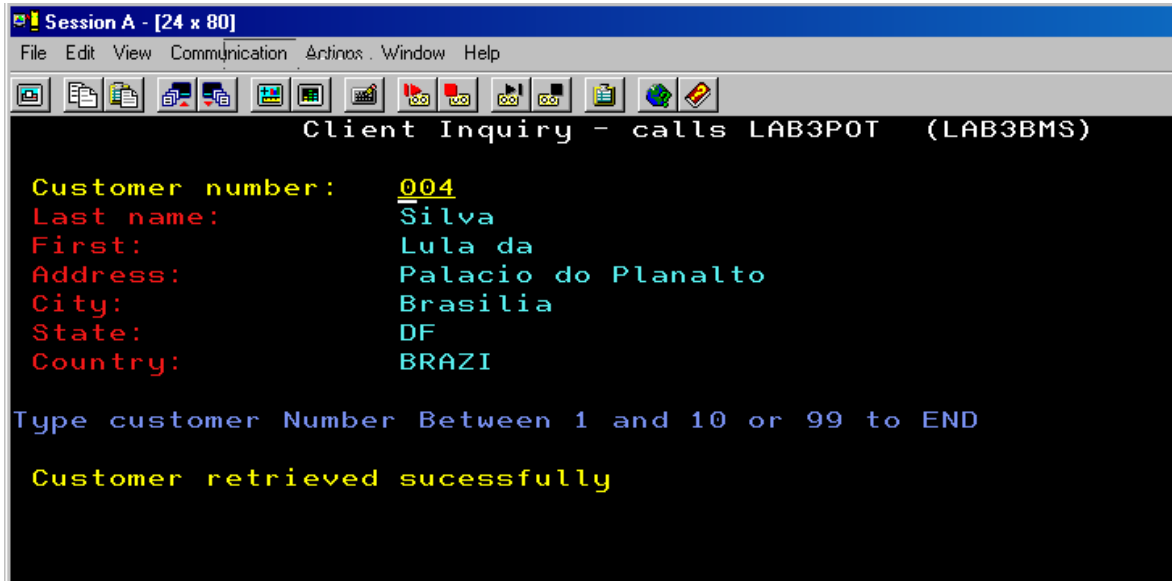
Agenda

08:40 - 09:40 - Build a smarter foundation for future investments
09:40 - 09:50 - Break (10 min)
09:50 - 10:50 - Smart Reuse- Transform green screens to Web, SOA, mobile, and portal
10:50 - 11:00 - Break (10 Min)
11:00 - 12:00 - Speed the development of multiplatform applications

12:00 - 01:00 - Lunch (1 hour)

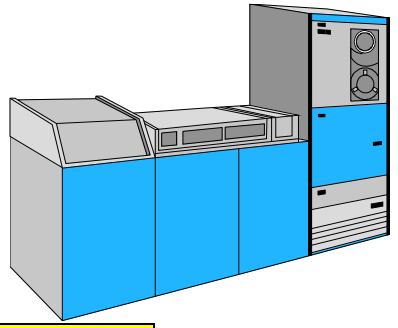
1:00 - 2:00 - Developing Web 2.0 applications using Mashup Tools
2:00 - 2:10 - Break (10 Min)
2:10 - 3:10 - Smart Work on System z: Enhance teamwork with multiplatform SCM tools
3:10 - 3:20 - Break (10 Min)
3:20 - 4:20 - Let's tie it all together and play in the sandbox
3:20 - 4:30 - Close

Existing **COBOL/CICS 3270 Application**



z/OS Texas

ZSERVEROS



EXEC CICS LINK LAB3POT

EXEC CICS READ
FILE ('POTVSAM')
INTO (WORK)

EXEC CICS RETURN

VSAM

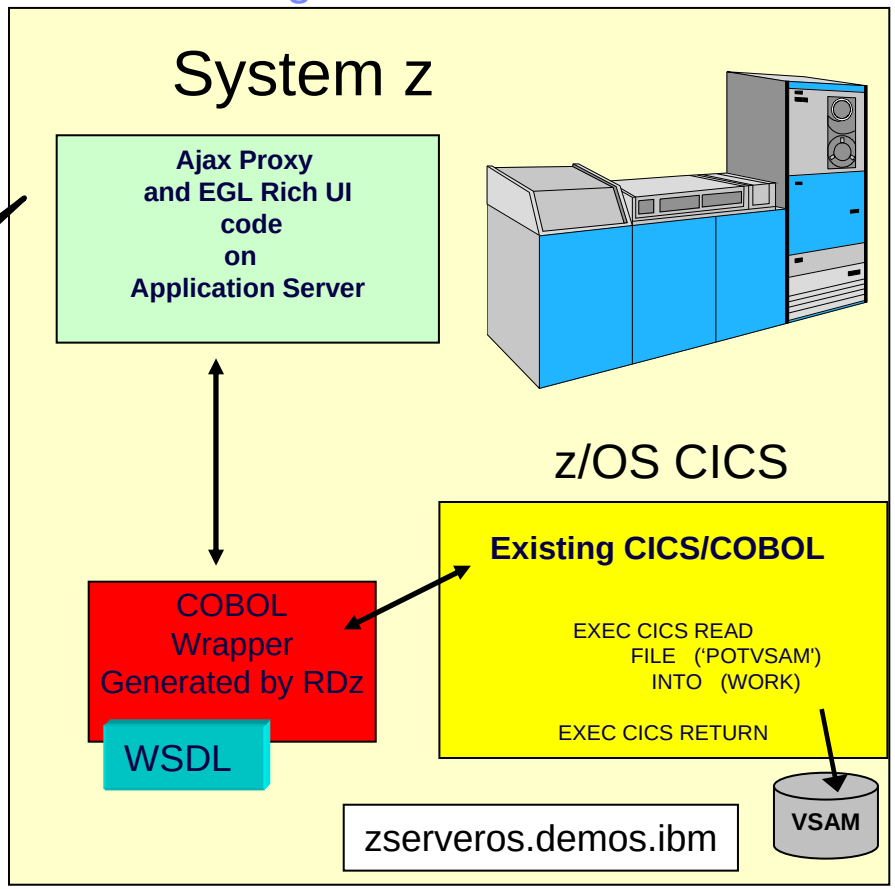
COBOL Client

COBOL Server

Link to PCOM

Invoking CICS Web services from iPhone

Using COBOL/CICS/VSAM

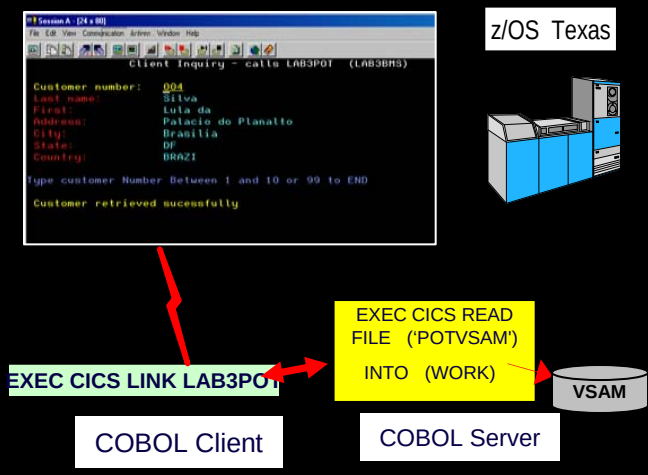


rbarosa@us.ibm.com

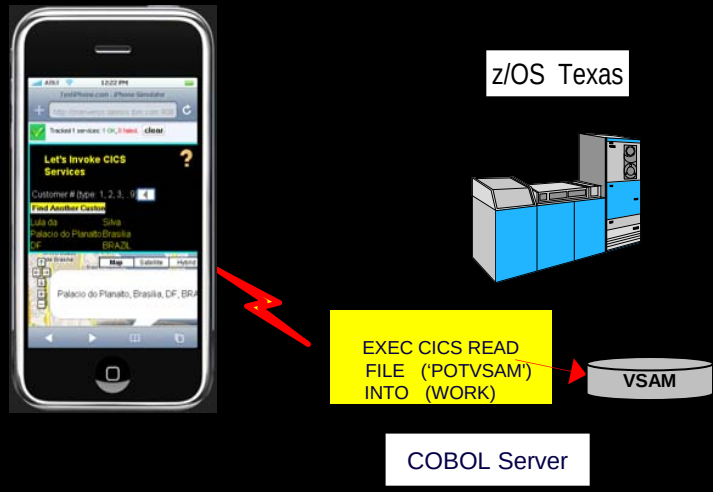
http://zserveros.demos.ibm.com:9080/iPhone/egl.html

More at:
http://www.ibm.com/developerworks/websphere/techjournal/0909_barosa/0909_barosa.html

LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (RDz)
(30 Minutes or less)



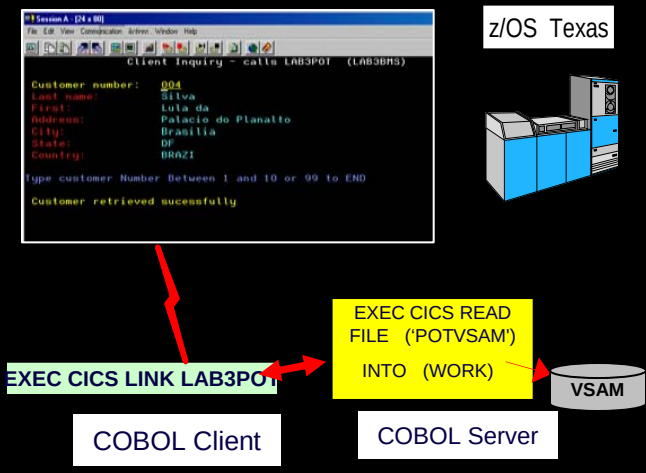
LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL)
(30 Minutes or more – no optional)



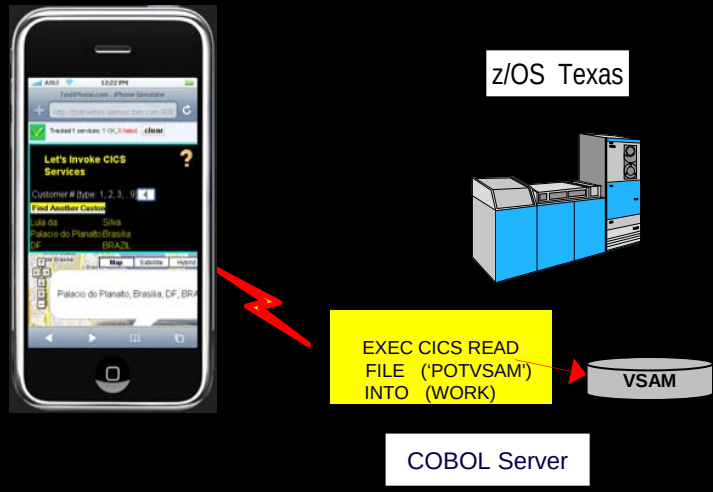
You can choose both or just one

LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (**RDz**)

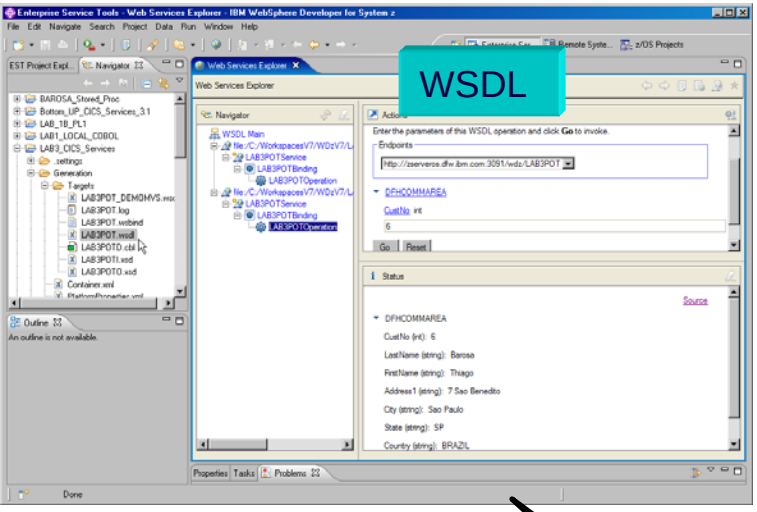
(30 Minutes or less)



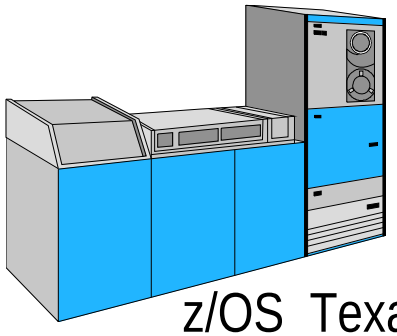
LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL)



LAB #1 Create CICS Services, deploy test and debug using RDz



- Create the COBOL/CICS Web Services
- Test the Web services



z/OS Texas
ZSERVEROS

COBOL ADAPTER

Converter Driver

EXEC CICS READ
FILE ('POTVSAM')
INTO (WORK)

EXEC CICS RETURN

VSAM

COBOL Server

Link to RDz

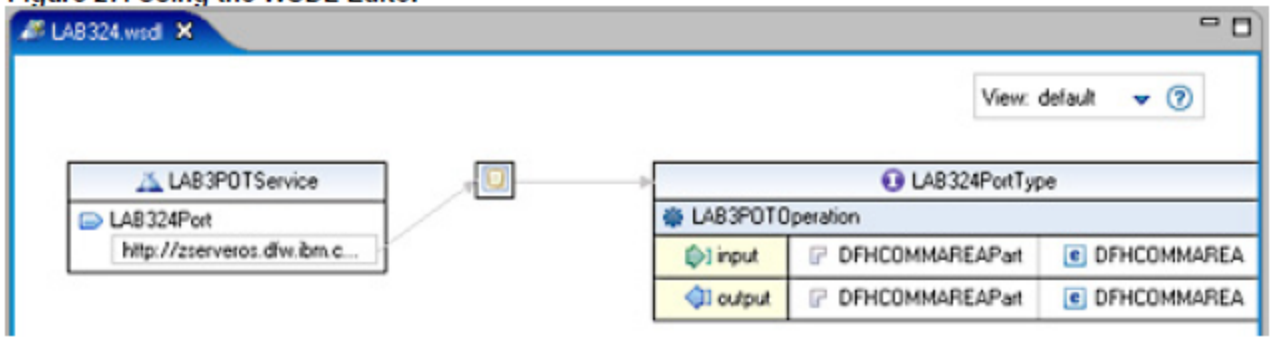
LAB1 #1 Create CICS Services, deploy test and debug using RDz

http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/emz/Wrap_existing_COBOL_program_as_Web_Services.pdf

IBM Enterprise Modernization for System z: **Wrap existing COBOL programs as Web Services with IBM Rational Developer for System z**

23. Verify the WSDL that was generated: double-click **LAB324D.wsdl** and use the WSDL Editor to visualize the input and output messages, as shown in Figure 27.

Figure 27. Using the WSDL Editor

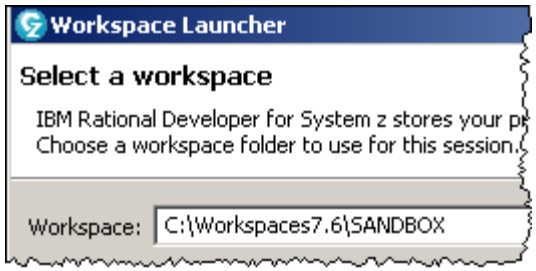


1. Start RDz

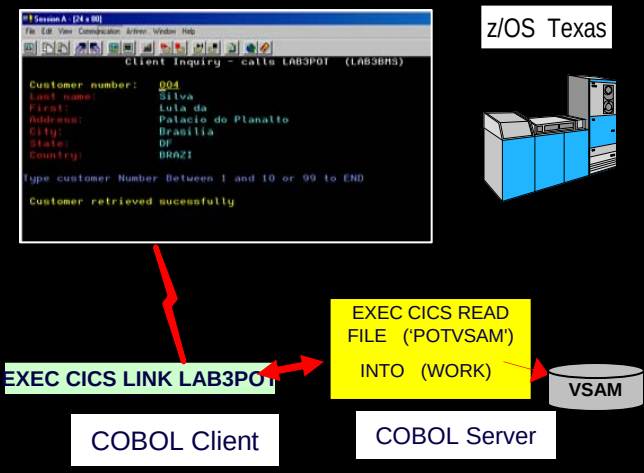
Start → Programs → IBM Software Development Platform for RDz7.6 → IBM Rational Developer for System z V7.6 → IBM Rational Developer for System z 7.6

or click on the icon in the windows quick launch:

2. Point to:

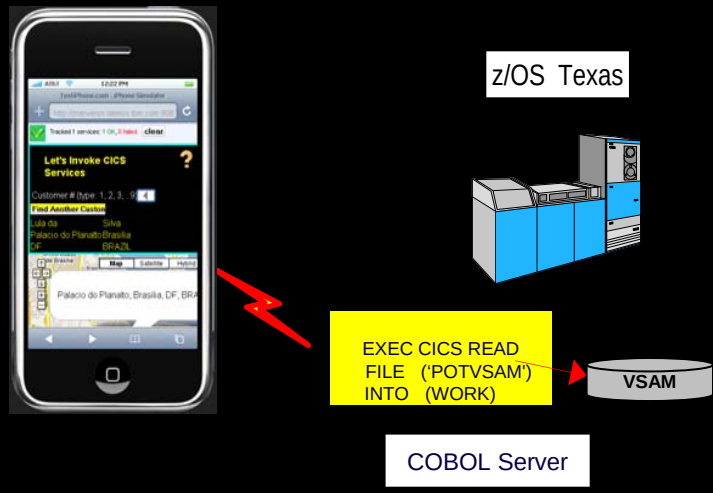


LAB #1. Create a CICS Web Service and WSDL using Rational Developer for System z (RDz)

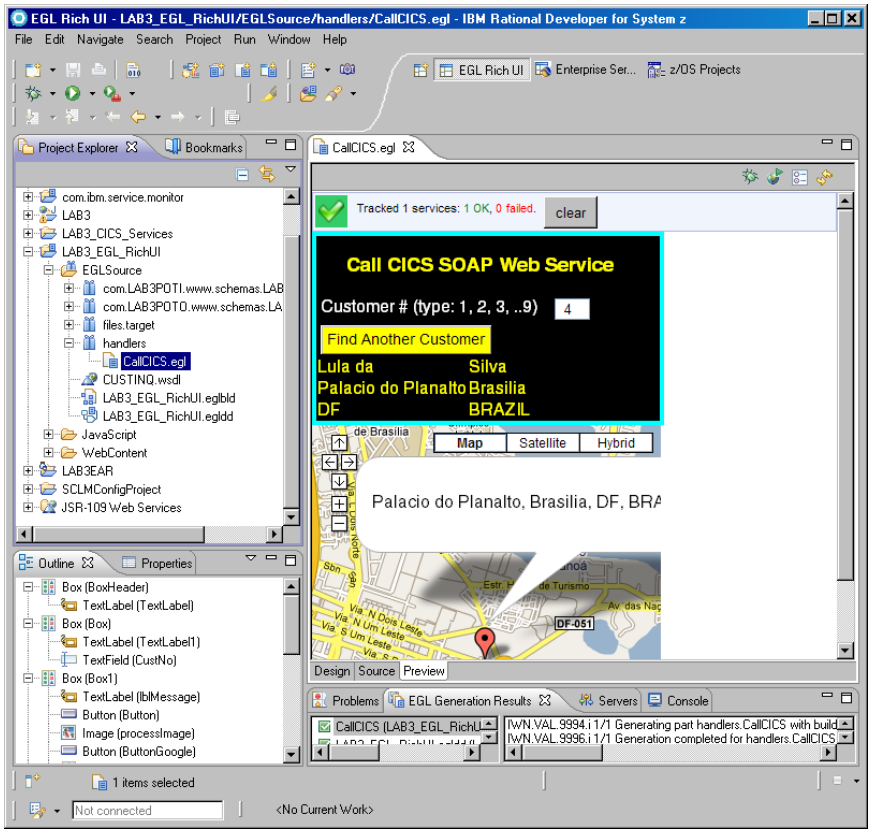


LAB #2. Create a Web 2.0 Interface using Rational Developer for System z with EGL (RDz/EGL)

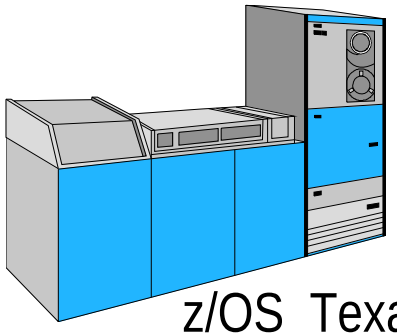
(30 Minutes or more – no optional)



LAB #2 Using RDz with EGL create the Web 2.0 Interface



- Create the Rich UI interface with EGL
- Test/deploy the Rich UI



z/OS Texas

ZSERVEROS

WSDL

```
EXEC CICS READ
     FILE   ('POTVSAM')
     INTO   (WORK)

EXEC CICS RETURN
```

VSAM

COBOL Server

LAB #2 Using RDz with EGL create the Web 2.0 Interface

http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/emz/Accessing_Existing_CICS_VSAM_application_from_iPhone.pdf

IBM Enterprise Modernization for System z:
Accessing existing CICS/VSAM application using an iPhone

Part 1 - Creating the service requester using Web 2.0 (User Interface).

1. Create a Rich UI EGL project and import the WSDL
2. Test the COBOL/CICS Service already deployed
3. Create the Rich UI EGL handler
4. Create the EGL Rich UI widgets
5. Create the EGL code to consume the CICS Service
6. Complete the EGL Rich UI code to invoke the CICS Service
7. Testing the Rich UI Application
8. Measuring the Service Call response time

Part 2 – Extending the Web 2.0 to invoke Google map services.

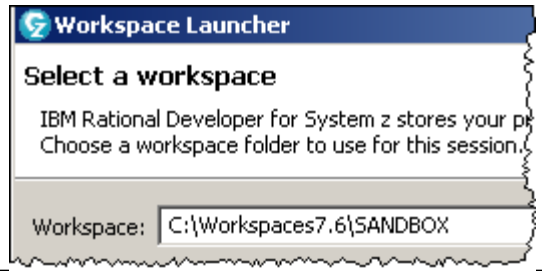
9. Modify the EGL Rich UI to add more controls and invoke the Google Map Service.
10. Testing using external Web Browser
11. Using an iPhone Browse emulator

1. Start RDz

Start → Programs → IBM Software Development Platform for RDz7.6 → IBM Rational Developer for System z V7.6 → IBM Rational Developer for System z 7.6

or click on the icon in the windows quick launch:

2. Point to:



Agenda

08:40 - 09:40 - Build a smarter foundation for future investments
09:40 - 09:50 - Break (10 min)
09:50 - 10:50 - Smart Reuse- Transform green screens to Web, SOA, mobile, and portal
10:50 - 11:00 - Break (10 Min)
11:00 - 12:00 - Speed the development of multiplatform applications

12:00 - 01:00 - Lunch (1 hour)

1:00 - 2:00 - Developing Web 2.0 applications using Mashup Tools
2:00 - 2:10 - Break (10 Min)
2:10 - 3:10 - Smart Work on System z: Enhance teamwork with multiplatform SCM tools
3:10 - 3:20 - Break (10 Min)
3:20 - 4:20 - Let's tie it all together and play in the sandbox
3:20 - 4:30 - Close

Next Steps: How to get started?

Resources to help kick-start your Enterprise software innovation

System z Sandboxes

Example assets and best practices providing low-risk, practical, hands-on path to leveraging IBM solutions

- Full version software trials
- 'Try online' hosted System z environments
- Hands-on-exercises

System z Starter Solutions

Solution to help you incrementally evolve core IT systems towards modern architectures and technologies

- Discover, reuse and grow
- Analyze and modernize
- Develop and manage
- Test and track

Jump-start your modernization projects!

Education Series

Modern Application Architecture for COBOL Developers

Learn how to design and integrate composite applications across CICS and WebSphere – leveraging existing COBOL processes

EGL Distance Learning

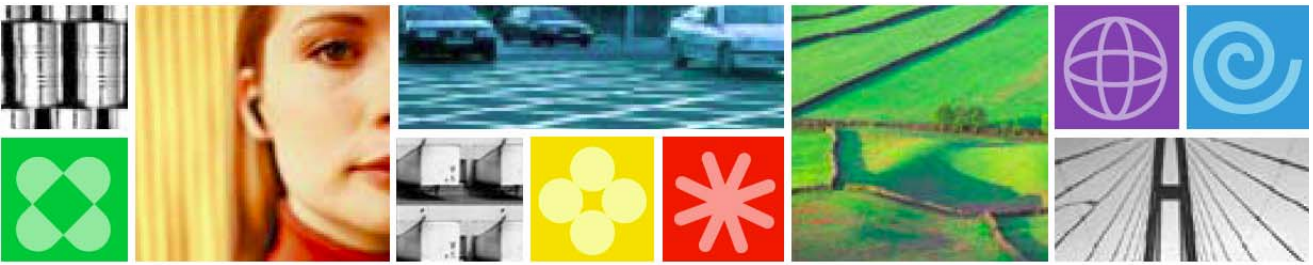
8 days of training free electronically - lectures, labs, and electronic support

Learn more at:
www.ibm.com/rational/modernization

http://w3-103.ibm.com/software/xl/portal/viewcontent?type=doc&srcID=R9&docID=X983555G31195K78

Bank Scenario – Originals slides

(not Ready as Jan 26,2010)



Scenario: JKHL Bank Organization Overview and Objectives

Who we are

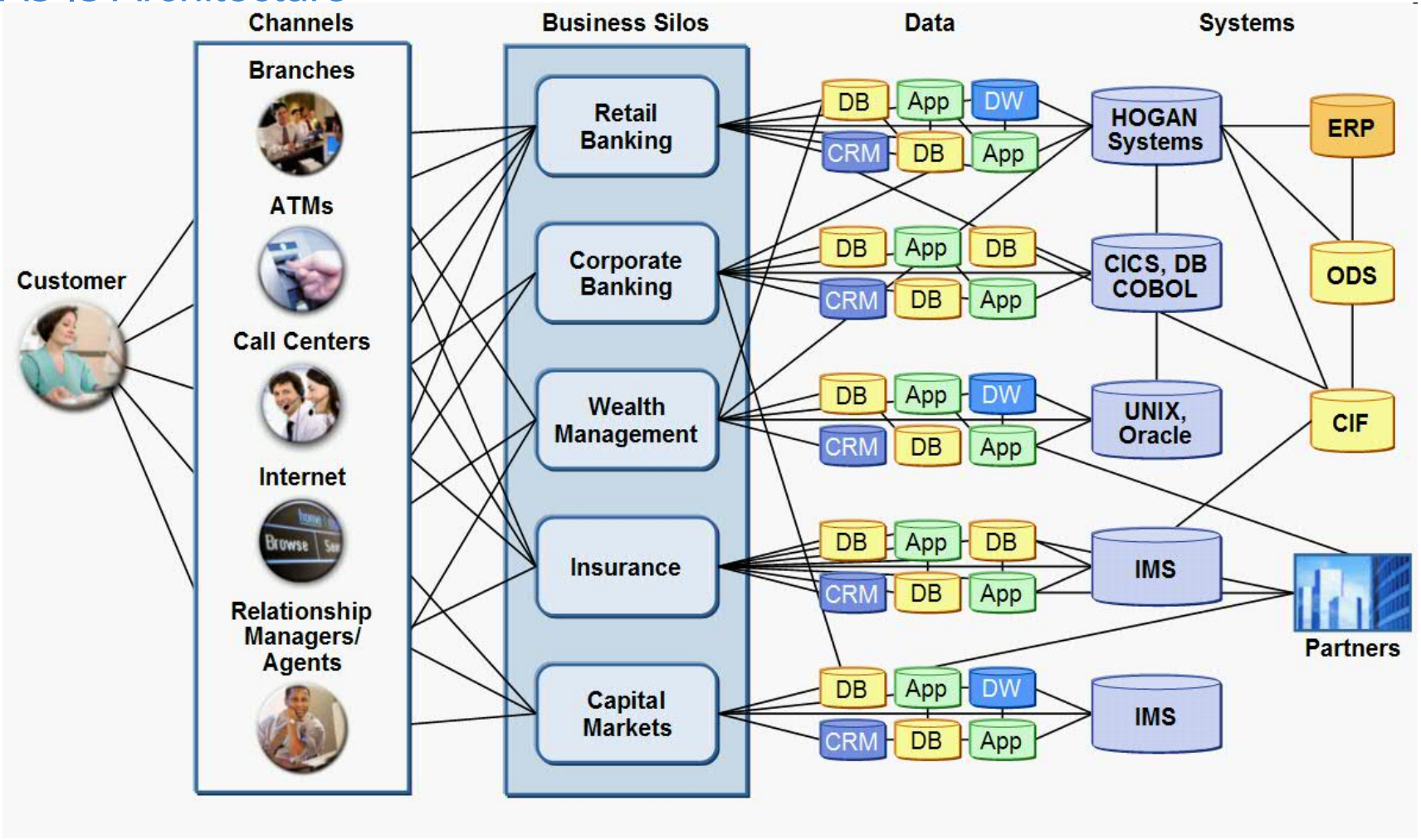
- **JKHL Bank is a tier 2 bank, who, over the last several years has acquired 5 smaller regional banks to create a broader geographic presence.**
- **Our business started in 1935, we have a well recognized brand and a solid brand image. By 1999, we had 500 branches in 6 countries providing retail, corporate, commercial and private banking.**
- **We have achieved remarkable success in migrating banking transactions from branches to alternative distribution channels. Over 50% all our banking transactions are carried out via alternative channels including ATMs, Kiosks, internet and the call center, a ratio that is above most banks.**
- **We win in the marketplace with a high-touch approach to our customers. We pride ourselves with understanding our customer segmentation and providing differentiated service based on customer profitability. We are known for our innovative products**

Who we want to be

- **In the short term, given recent changes in financial markets, we are reducing costs and leveraging existing assets and processes, while at the same time,** providing robust capabilities to manage dynamic issues of risks as well as regulatory compliance
- **In the long term, we want to** achieve differentiation **through improved customer experience and innovative self-service offerings**
- **We want to show that we really** know our customers **through the use of integrated data and analytics, and by having more meaningful customer interactions that can develop profitable customers, who go beyond loyalty to become advocates.**
- **We want to build on our ability to deliver** innovative products **and services as well as provide a seamless experience across the different LOBs and channels**

In order to make their banking operations more profitable, JKHL Bank recruits IBM to analyze its existing business processes and provide recommendations for a business transformation.

As-Is Architecture

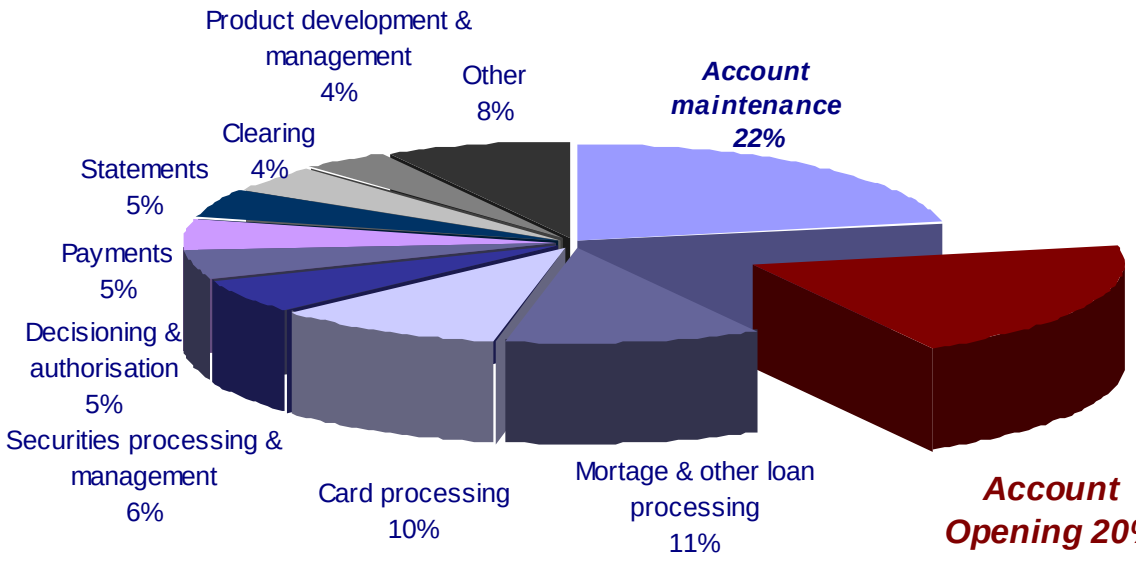


- Inconsistent View of Customer
- Silos of Information
- Information Locked in Repositories
- Inconsistent Data
- Applications are hard-wired
- Poor Channel Communication

Why JKHL Considers Account Opening to be a Key Initiative

Cost Breakdown of a Bank's Activities

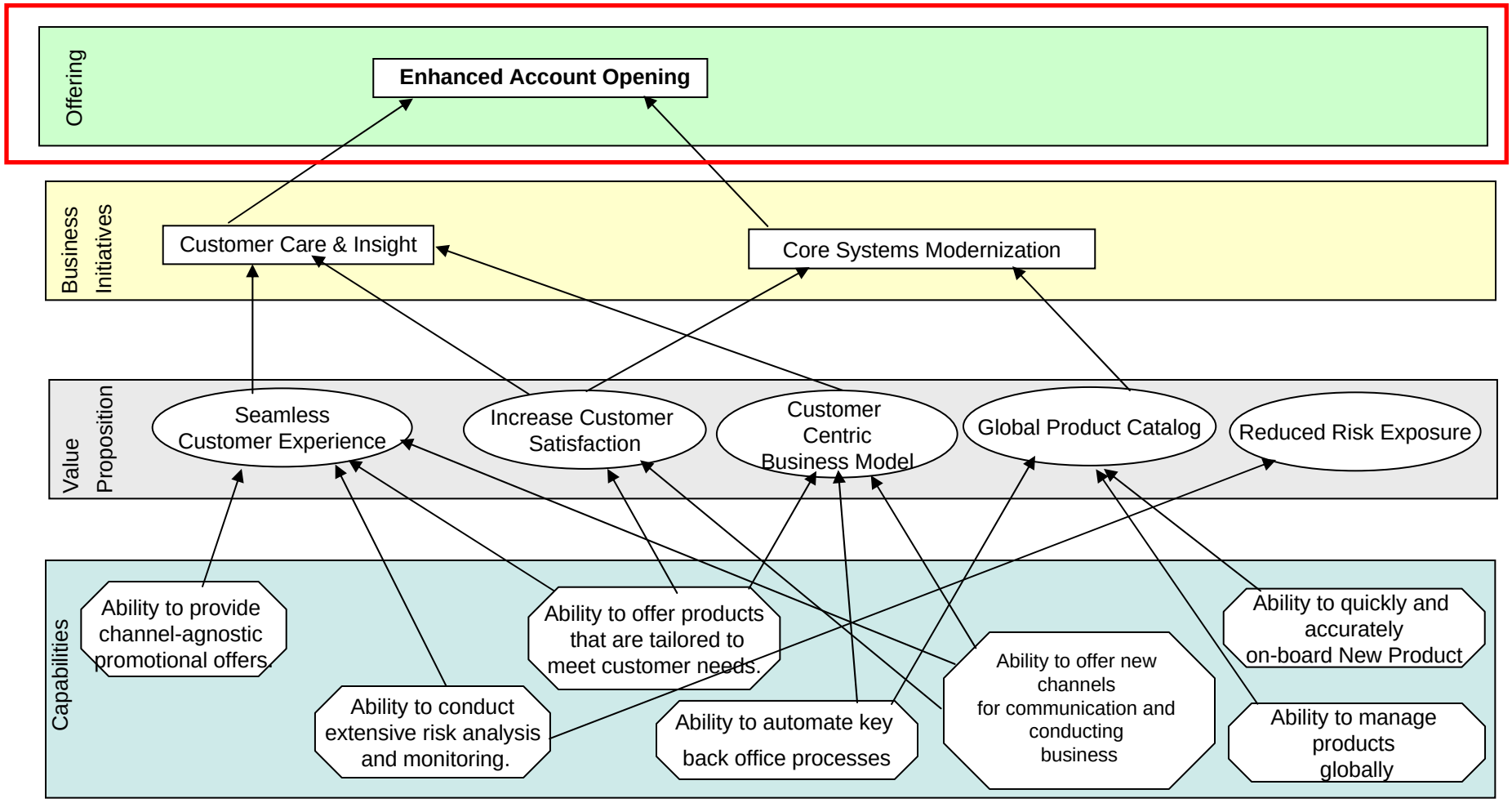
Total Manufacturing & Processing Cost Base



1. ***Account Opening is a major item of the cost base***
2. ***Duplication of Efforts and Systems put pressure on AO costs***
3. ***Maintaining interfaces among disparate systems is the major reason behind costs***

Source: IBM Banking Industry Thought Leadership products; IBM client engagements

Capability Model Ties Business Capabilities to Initiatives



IBM verifies that the core capabilities which JKHL Bank identified as being vital to the company, are mapped to its key business initiatives, which in turn is addressed by the company's enhanced offerings.

JKHLE - Current State Assessment / General Observations

Accounts:

- **Account information is duplicated and inconsistent within multiple systems and LOBs**
- **All pre-requisites for a new account cannot be satisfied via any channel (of choice)**
- **Promotions are not consistently offered across delivery channels**
- **Limited risk management during account open process**

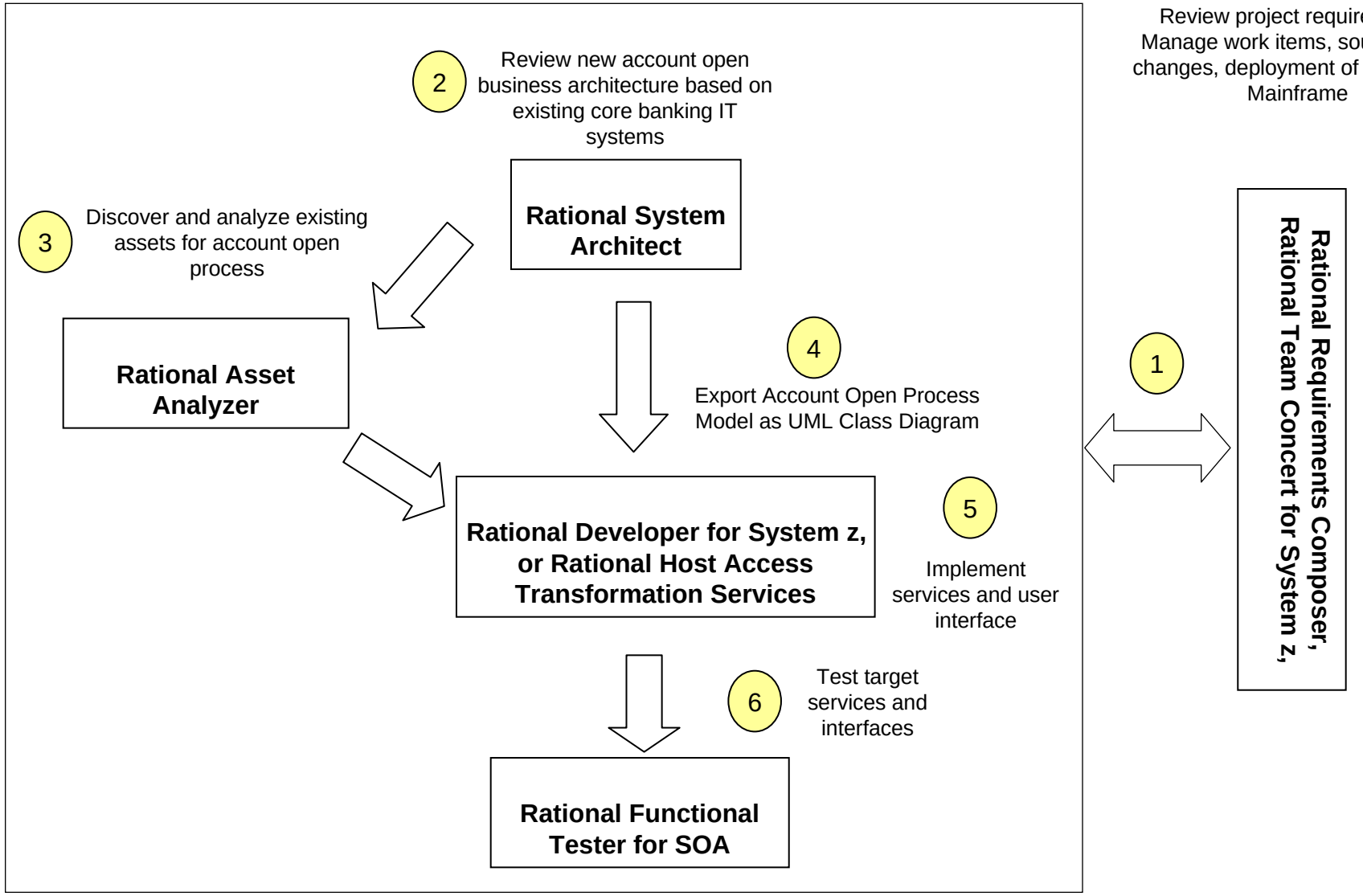
Customers:

- **Lack of a single presentation of information across channels**
- **Unable to exchange information with bank via channel of choice**
- **Customers are forced to re-supply personal information for every new account they open with the bank**
- **Customers do not understand why all accounts and services are not available using a single user interface**

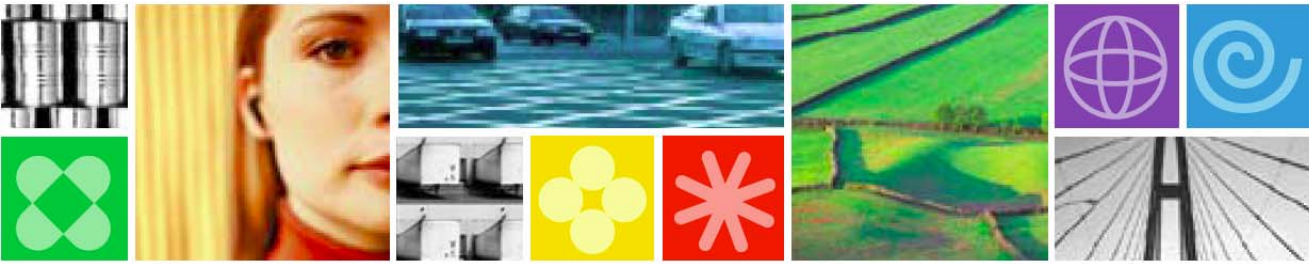
Systems:

- **IT systems designed to support various channels and lines of business as separate systems Web system, Catalog/Call Center system and Store system are on very different infrastructures**
- **Many legacy applications**
  - Less flexible than newer technologies – monolithic, home grown applications which are not easily adaptable to open standards
- **Numerous technologies supported**
  - Some supported long past end of life
  - Attempts by a few IT departments within the company to adopt SOA
  - No enterprise-wide SOA strategy
- **Some use of newer, web-based technologies**

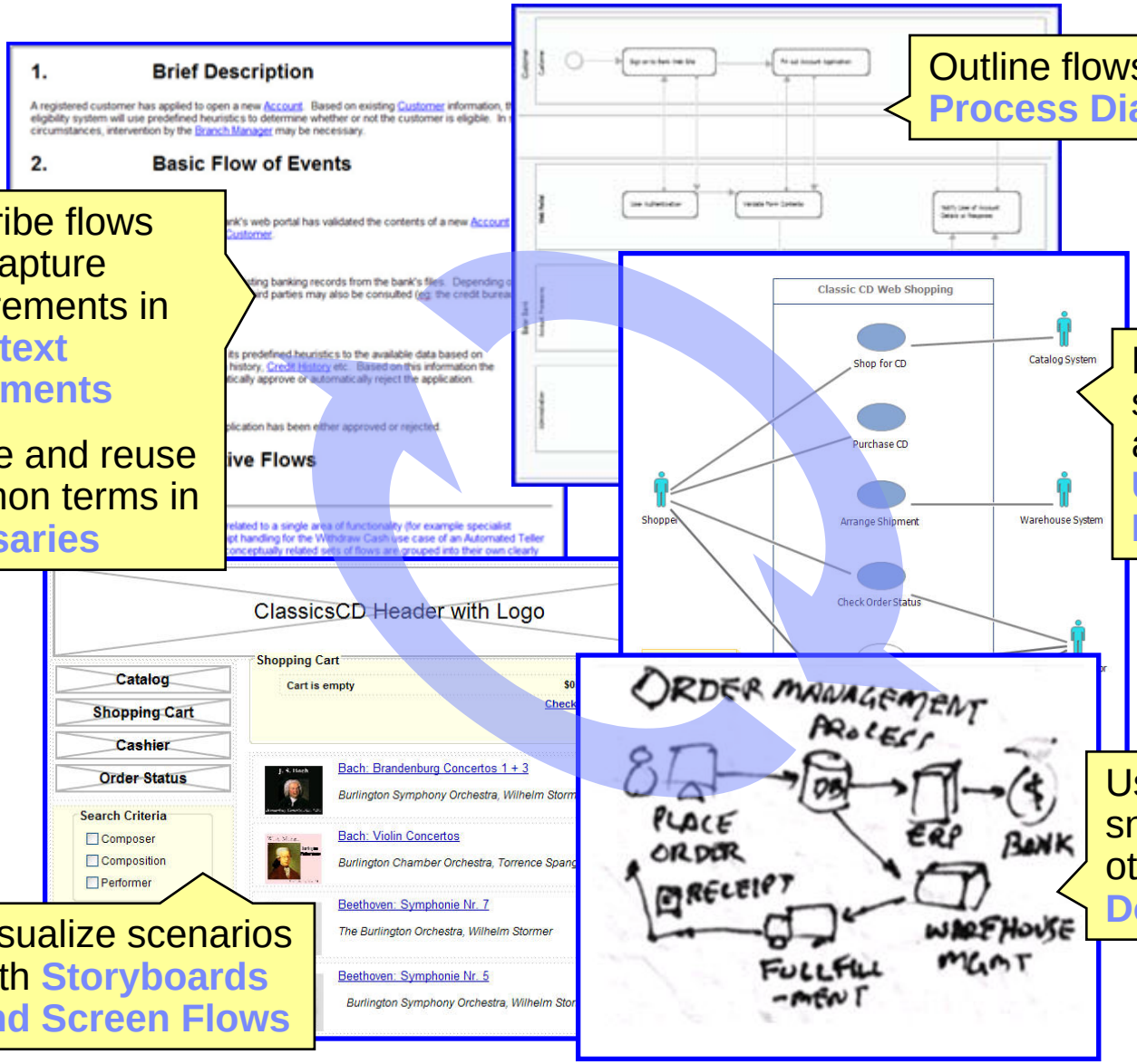
Rational Banking Modernization



Backup



Use scenarios to uncover customer needs



Outline flows with **Process Diagrams**

Describe flows and capture requirements in **Rich-text Documents**

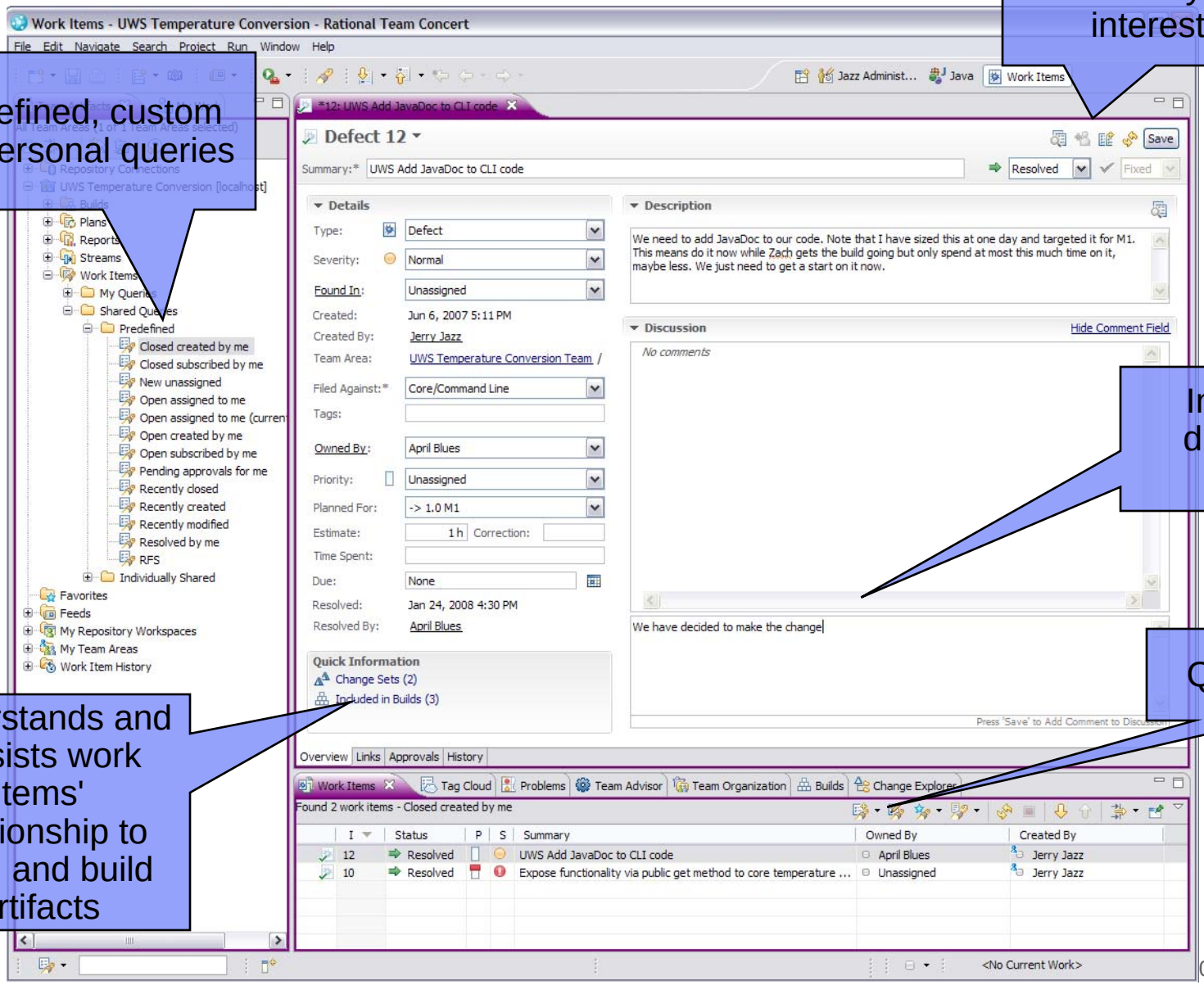
Define and reuse common terms in **Glossaries**

Describe actors, system boundaries and user goals in **Use Case Diagrams**

Use whiteboard snapshots and other **Informal Documentation**

Visualize scenarios with **Storyboards and Screen Flows**

Team Concert Work items



Predefined, custom and personal queries

Subscribe to work items you're interested in

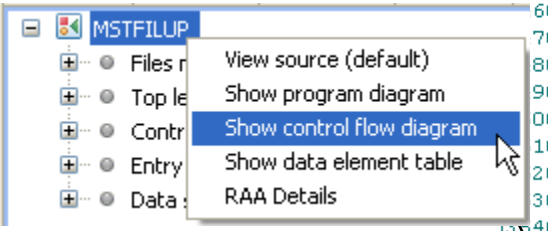
Integrated discussion threads

Query results

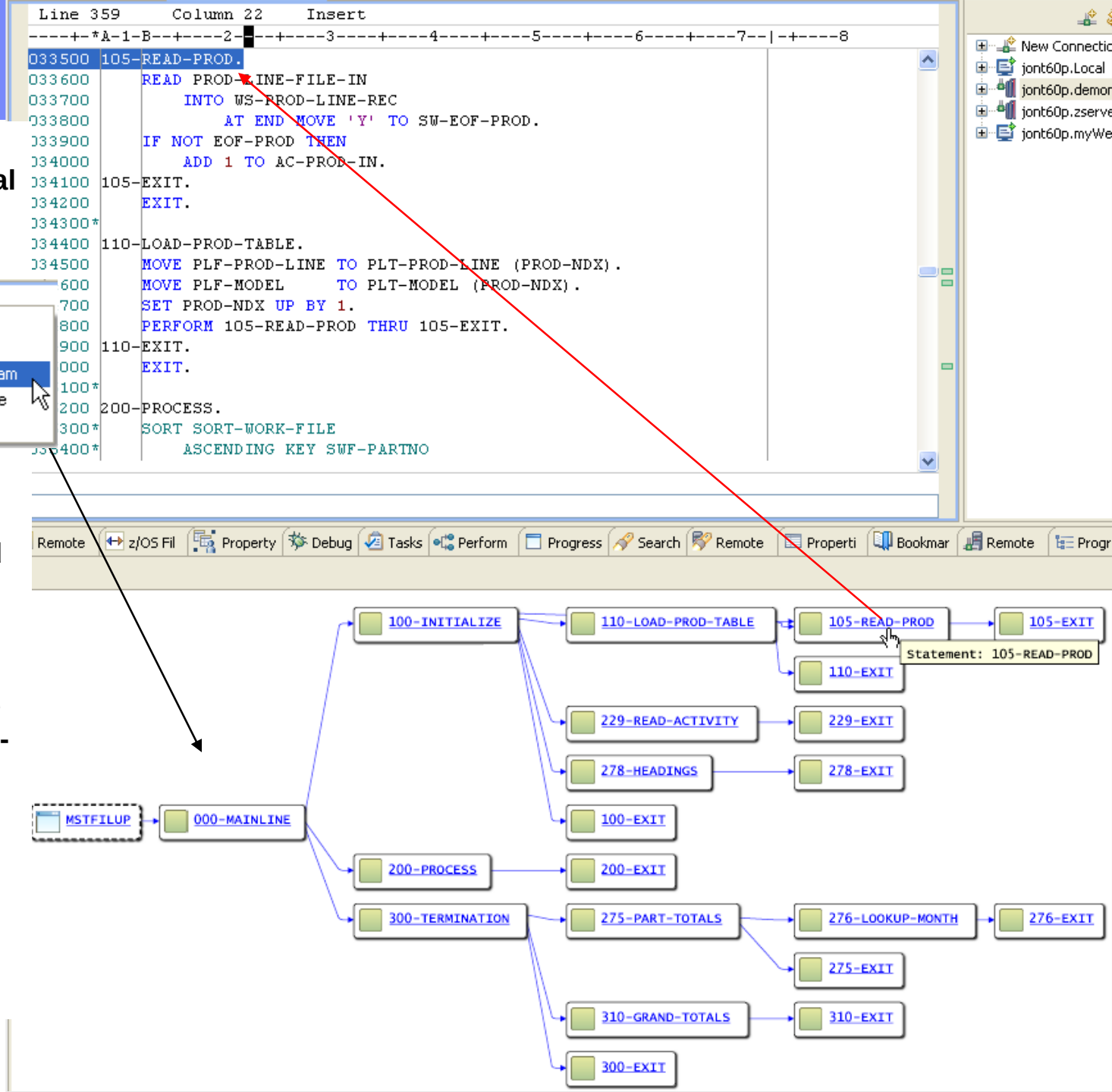
Understands and persists work items' relationship to SCM and build artifacts

RAAi Control Flow Model

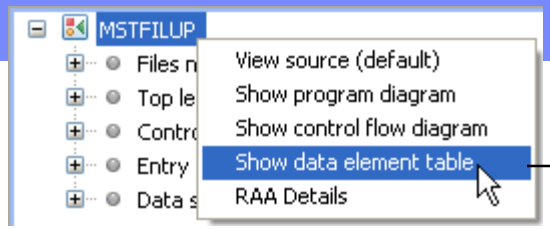
- **Graphically reveals program's procedural structure**

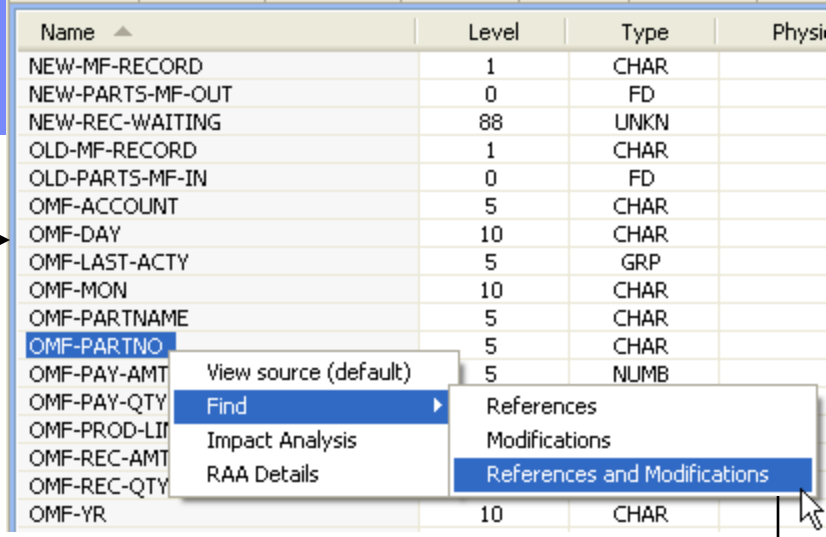


- **Can click a paragraph name and source code within the LPEX editor aligns**
- **Provides unbeatable combination of "top-down" and "bottom-up" code learning model**



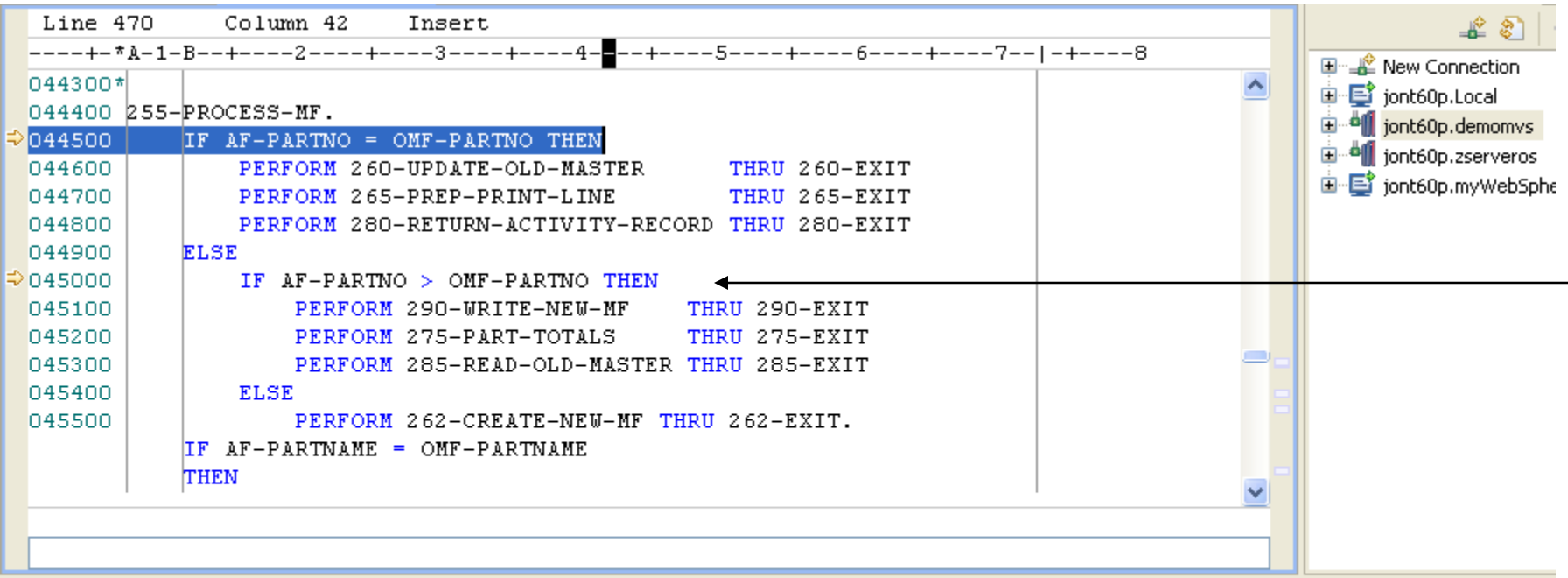
RAAi – Search for Variables References and Modifications

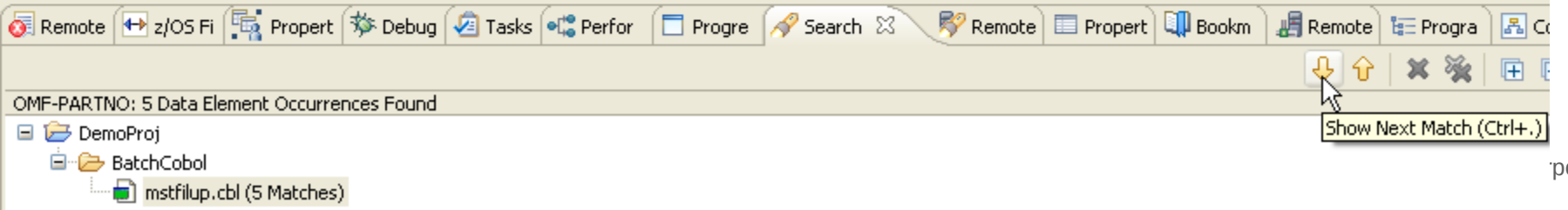




Show sorted X-Ref of program variables

When found can use to answer critical and important Maintenance Questions



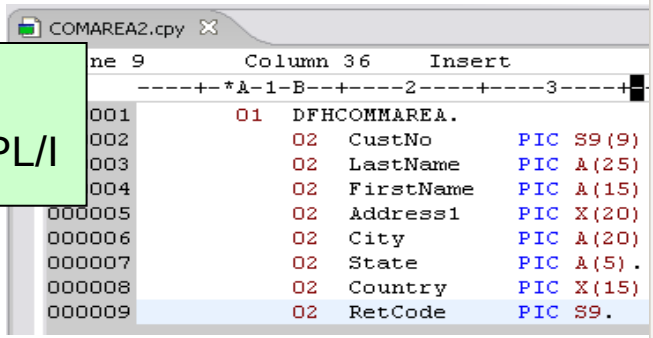


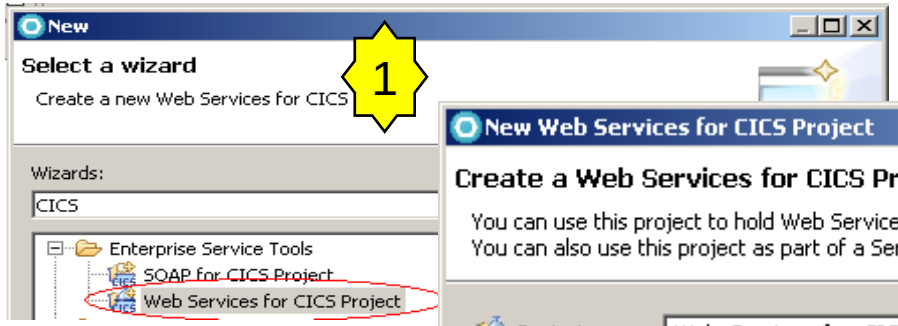
Web Services for CICS 3.x

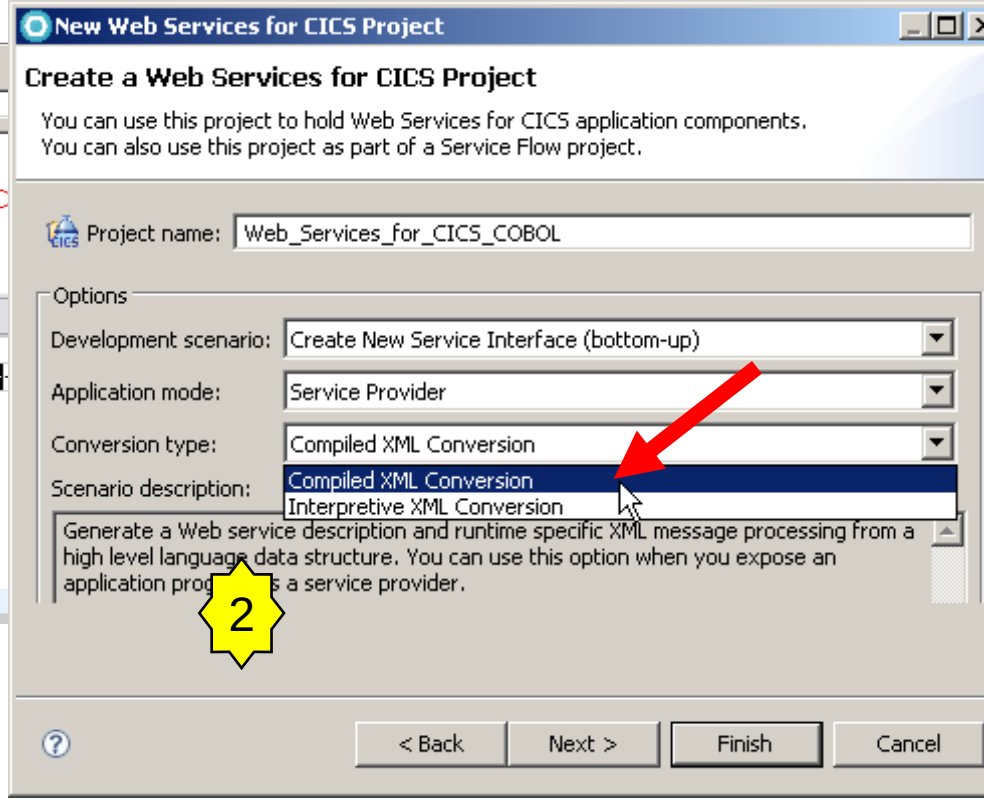
Example Bottom-up **Compiled** XML Conversion - 1 of 2

Web service WSDL & Converters

Existing COBOL or PL/I

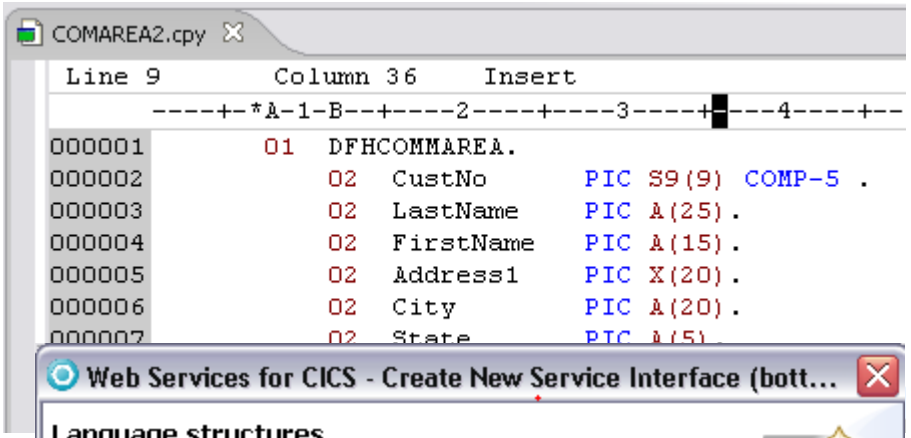




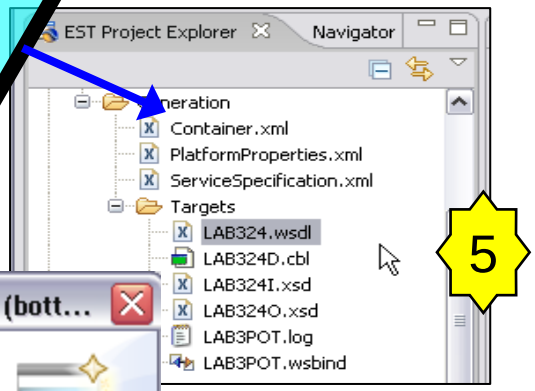


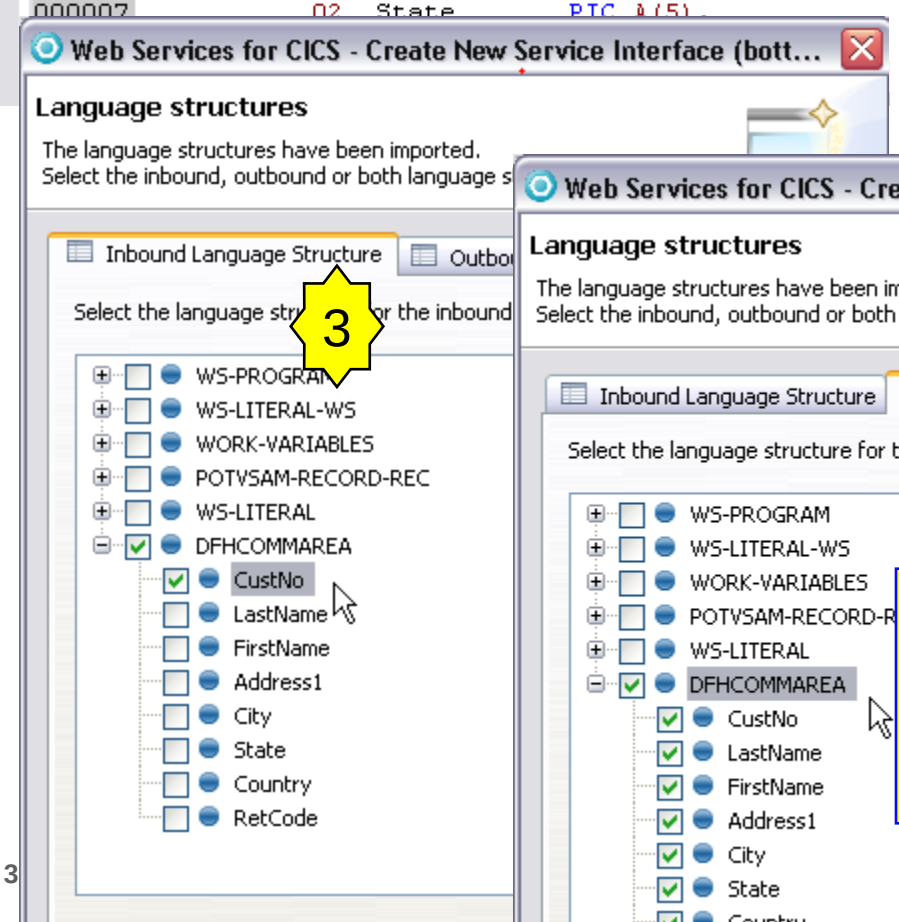
Bottom-up

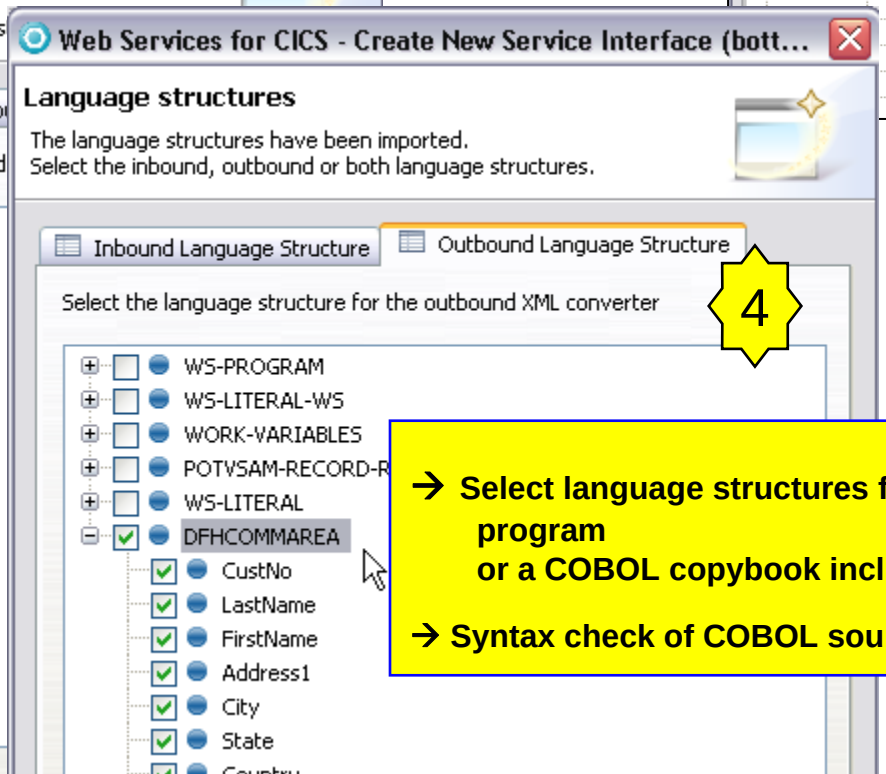
CICS Web Services Compiled Conversion – Example – 2 of 2



RDz XML Enablement (*Bottom-up*)







→ **Select language structures from a complete COBOL program or a COBOL copybook included by the program**

→ **Syntax check of COBOL source during import.**

Recording Scripts

Verification Points

Software
Trials and betas
Support
Training and certification
Library
Events
News

Functional Tester Sees Data

You See…

Property	Value
.bounds	Rect[84,100,787,809]
.class	Html.HtmlDocument
.cookie	IBMSurveyTest=isEnabled
.offsetHeight	809
.offsetWidth	787
.text	IBM Rational Software Home
.title	IBM Rational Software
.url	HRef(http://www.ibm.com/software/rational/)

Functional Tester Sees Properties

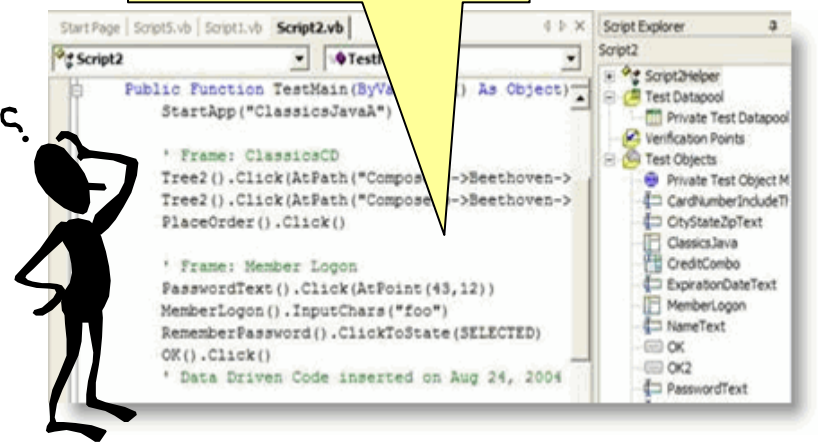
- Automated Validation
  - Functional Tester captures data and properties that can be invisible to users
  - During script execution, current results are compared to stored baselines
  - Discrepancies are flagged and reported to user in an HTML based test log

Simplified Test Representation

Scenario: My business SME must contribute to the QA activities

Today the business user is overwhelmed by technical depth

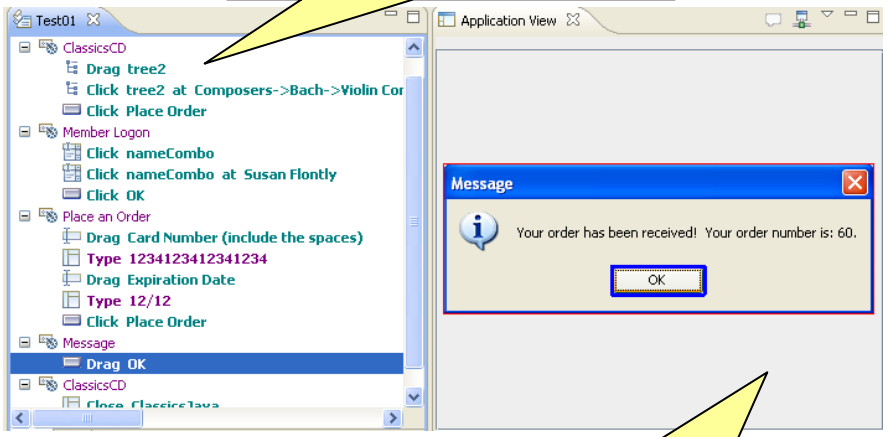
Business user is faced with a programming language …



… and ‘Objects’ and other advanced options

With 8.1 we will speak the language of the business user

Natural language statements grouped by window/screen touched



Application view synchronized with group/step in the Test View