Modernizing your Assets

Andy Symonds
Enterprise Modernization, Rational
symondsa@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software

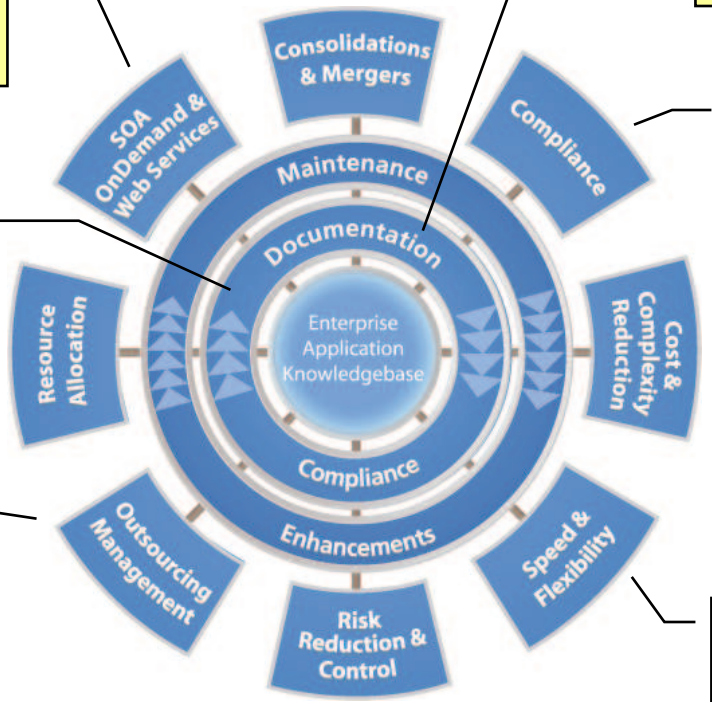
I want to move my enterprise applications to SOA – where do I start ?

We want to reuse key parts of our mainframe applications in a SOA. Where do we start?

We've accumulated decades of changes to our valuable applications, but we don't have reliable documentation of them

We've outsourced our application development; how do we keep intellectual control of our applications?

We can't be sure how changes in one part of an application will affect other parts, so maintenance is difficult and dangerous



Our applications staff are retiring or moving on, and we are losing the subject matter expertise

We need to change our applications to address industry or government mandates ... and it's going to take a lot of time and money

The cost to enhance our applications are holding back top-level business growth

We can't afford to rewrite our mainframe applications as new web applications

Rational's Analysis Tools – RAA and RTW

Both scan/parse code to build a repository

Have different meta-models

RAA focuses on high level object relationships

Program calls program, uses copybook, uses data

Coarse grained impact analysis - WHAT may be impacted

Intended to cover whole enterprise with an end to end view of mainframe, J2EE and future Microsoft support.

Browser based interface, centralised server.

RTW focuses on detail level parsing

Provides detailed analysis at system/sub-system and Project level

Enables data-flow, execution path analysis

Fine grained impact analysis – HOW things are impacted

Enables advanced functions for :

Component Extraction, Complexity reduction, Business rule extraction

Provides Rich Diagramming and Documentation

Workstation deployment

Specialises in Cobol, CICS, DB2 etc, support for VB6,C/C++ and Java in the near future.

Rational Asset Analyzer (Formerly WSAA)

High level scan - Immense scalability

One Client's meta-data:

200K programs, 140K batch jobs, 126K DB2 columns, 2.4M literals, 81M data elements

Over 250 million LOC scanned weekly.

Fast, Basic metrics, Program/data impacts

Keyword Query ability – control naming conventions

Low footprint

SAAS model – no client install

Server Scans Mainframe/Distributed

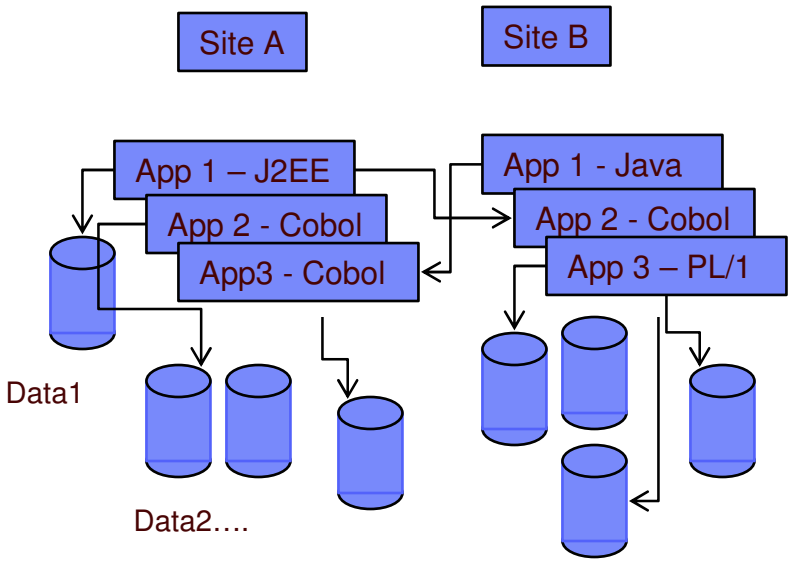
PVU pricing model – unlimited users.

Broad Language coverage

COBOL, PL/I support, J2EE, Web Services

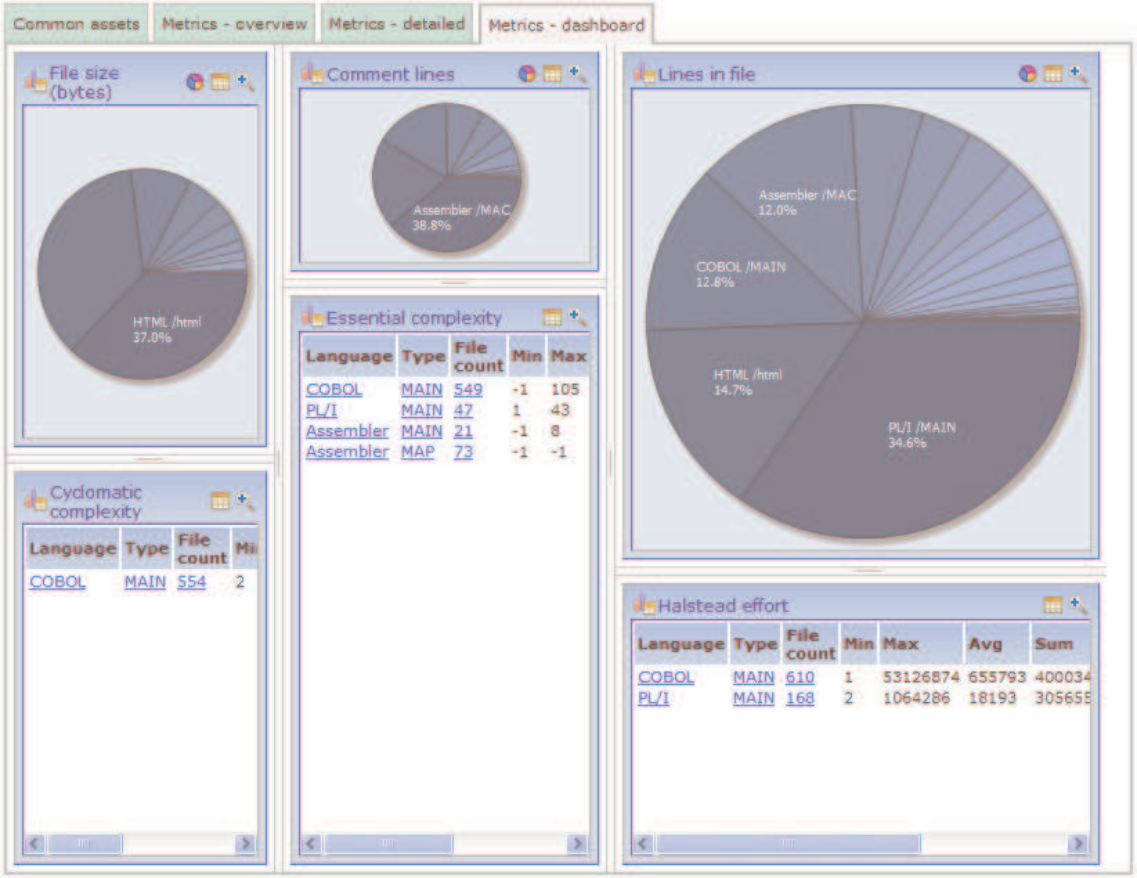
.NET and VB development priority for early 2009

Bridge to RTW for deeper analysis



Provides end to end impact analysis based on program call chains and data element usage

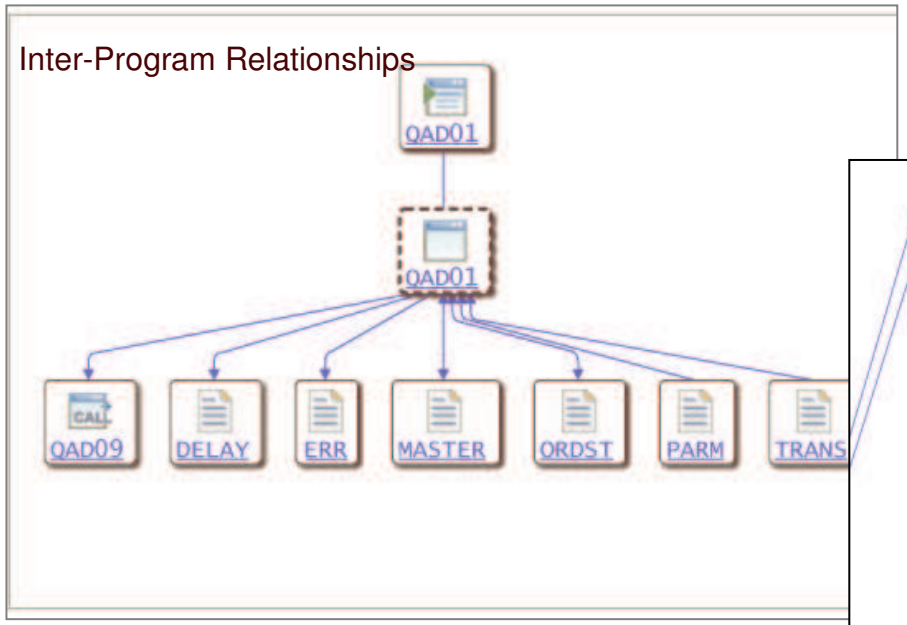
RAA -High level Inventory – Basic Metrics

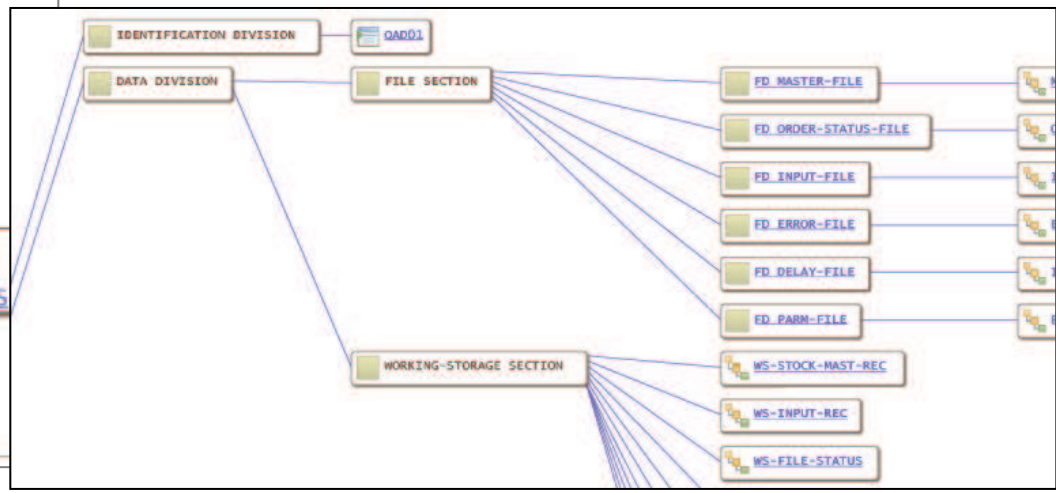


RAA – Application Scope

Quickly understand application scope

- Group artifacts into user-defined groups called Applications to limit scope to area of interest
- Use various types of diagrams for understanding how the application “hangs together”
- Create user-defined relationships for situations where known relationships cannot be determined
- Perform enterprise-level keyword searches

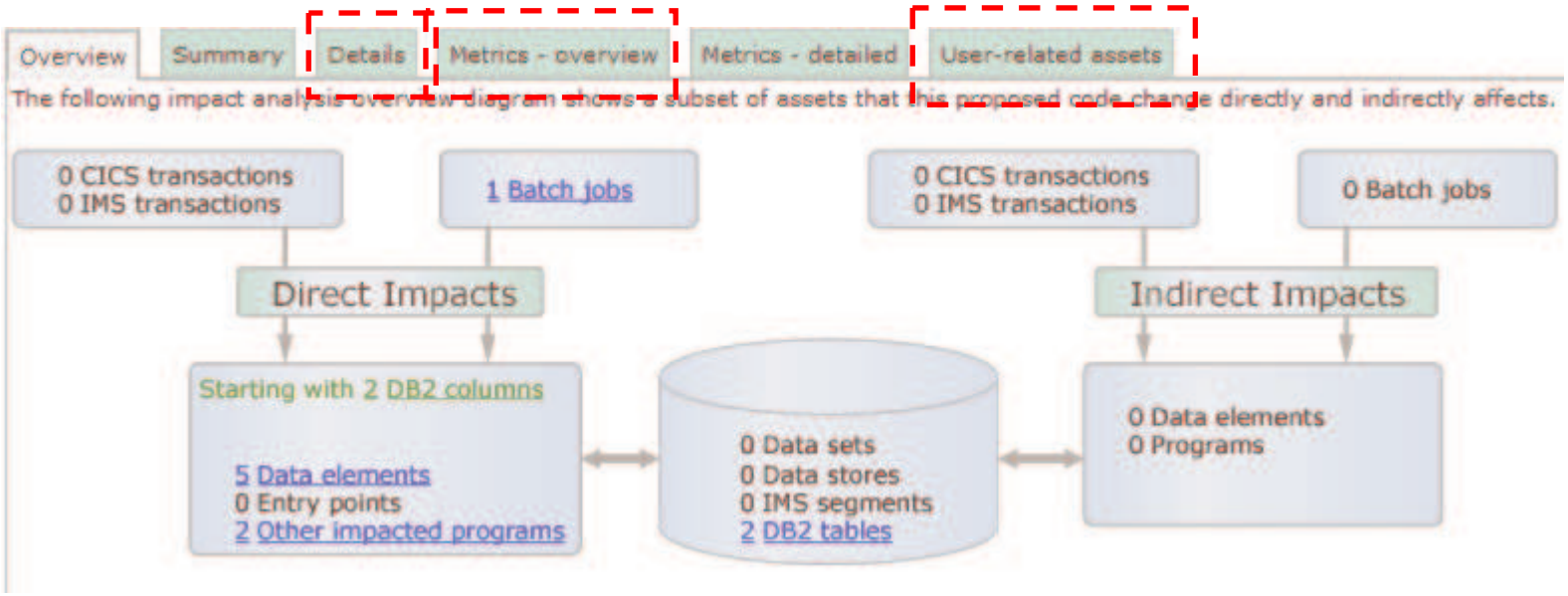




RAA – End to End Impact Analysis

Reduce time to market & risk of downtime by understanding change impact upfront

- Reduce time to determine **scope of change** whether for new enhancements, or even maintenance efforts
- View the metrics for impacted artifacts to determine the **risk of change** i.e. cyclomatic complexity, lines in file, etc.
- Traverse user-defined relationships to determine **impacts across platforms** i.e. follow dependencies from mainframe to J2EE and back.
- Create a "**bill of materials**" of impacted artifacts by evaluating the details page



How RAA can help

Identify Scope of potential services by understanding data usage and high level relationships of the entire portfolio.

High level inventory of all assets

"impact of change" at data element level to scope overall enterprise dependencies

Direct : Data used by Program

Indirect : Other programs use same Data

Challenges to Evolving Enterprise Applications

Complexity, Volume and Coupling

Code Quality

Abstraction – so much detail its hard to see the big picture

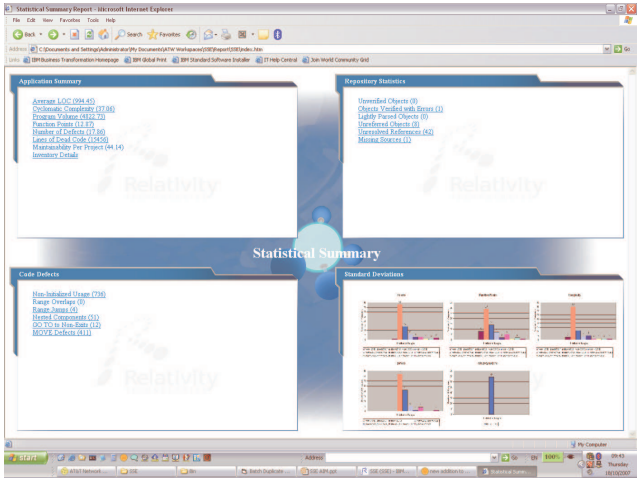
Traceability

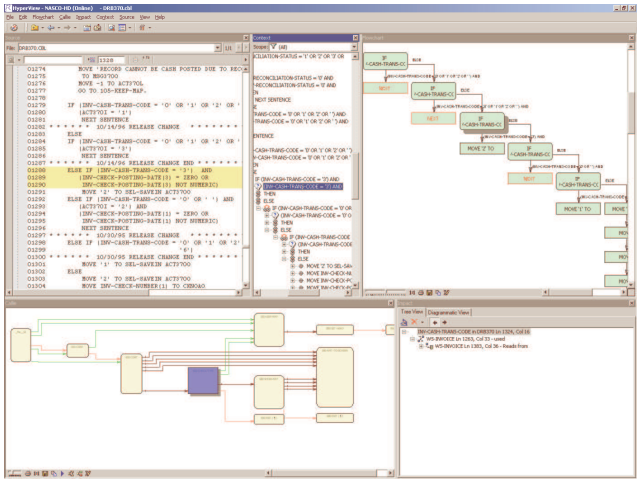
Dead Code , Obsolete Code

Architecture – Custom System Software vs Business Logic

Rational Transformation Workbench

- **RTW analyzer** provides deep insight into even the most complex application
  - Detailed Inventory and Metrics
  - Automated Diagrams and Documentation
  - Impact Analysis – cause and consequence of change
  - Data usage, Data flow
- **Business Rule Extension** automates the discovery and governance of business logic
  - Adds High Level Business Function/Rule Documentation
  - Can be linked to Process/Requirements
  - Glossary function links Business Names to their technical implementation
- **Architect Extension** enables the restructuring of programs to increase their maintainability and quality
  - Automated Dead Code removal
  - Extract Logical Components

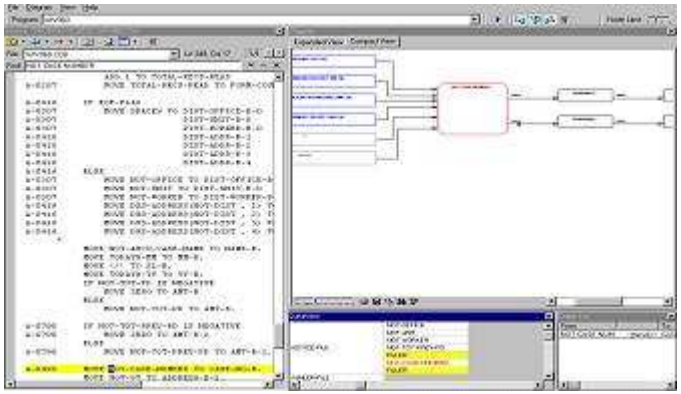




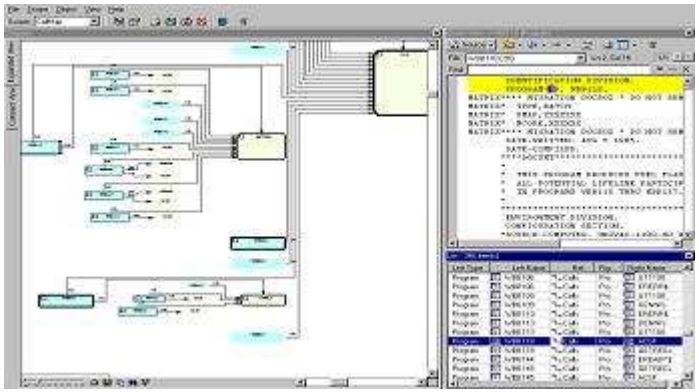
RTW – Base Capabilities

Metrics

Impact Analysis



Data Flow Analysis



Call Maps

RTW - Understanding Software Assets

Group and tag software assets to enhance business and technical insight

- *Map business terms onto technical concepts to ease communication challenges*
- *Start to see how larger components actually interact*

Richly understand technical reality to reduce time, cost, and risk of development activities

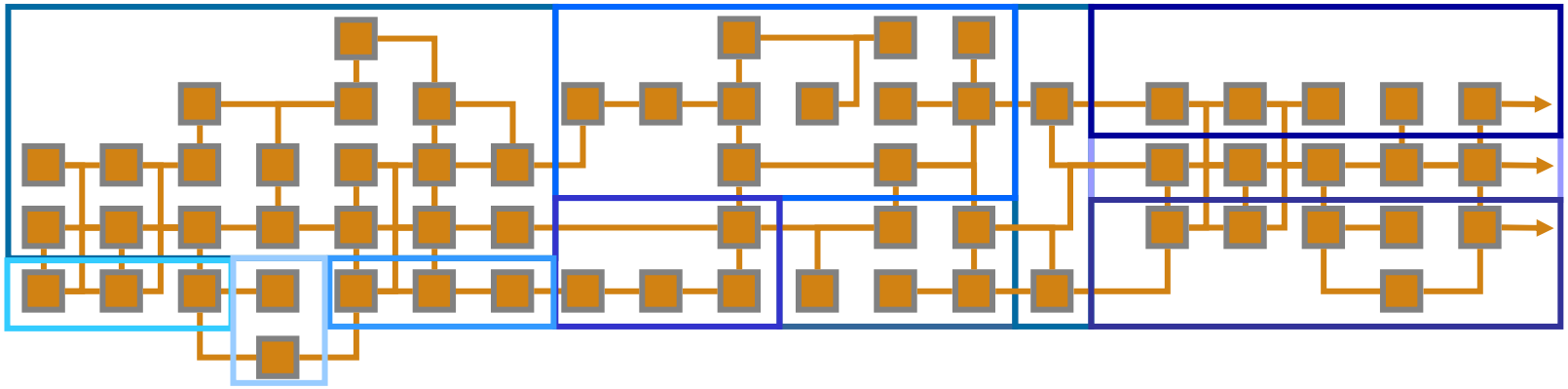
- *Powerful querying to locate elements of interest*
- *Synchronized visualizations boost insight and improve communication across distributed teams*
- *Identify architectural and technical inefficiencies*

Mapping Business Context

Business Context View

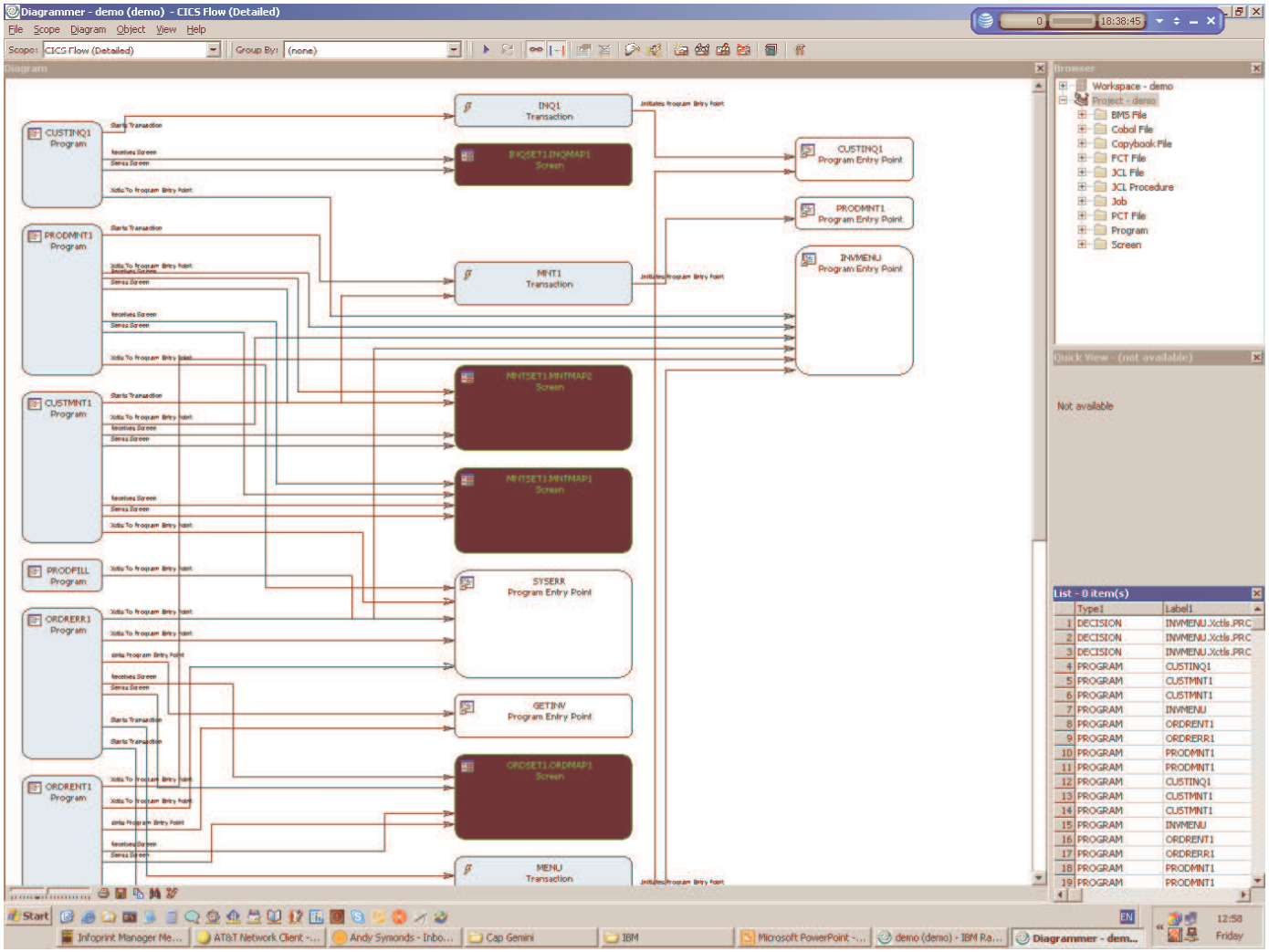
Order Entry
Insurance Product Catalog
Claims Management
Underwriting
Order Validation
Client Management
Client Eligibility
Route for Underwriting
Add Client
Edit Client
Financial Management

Technical View

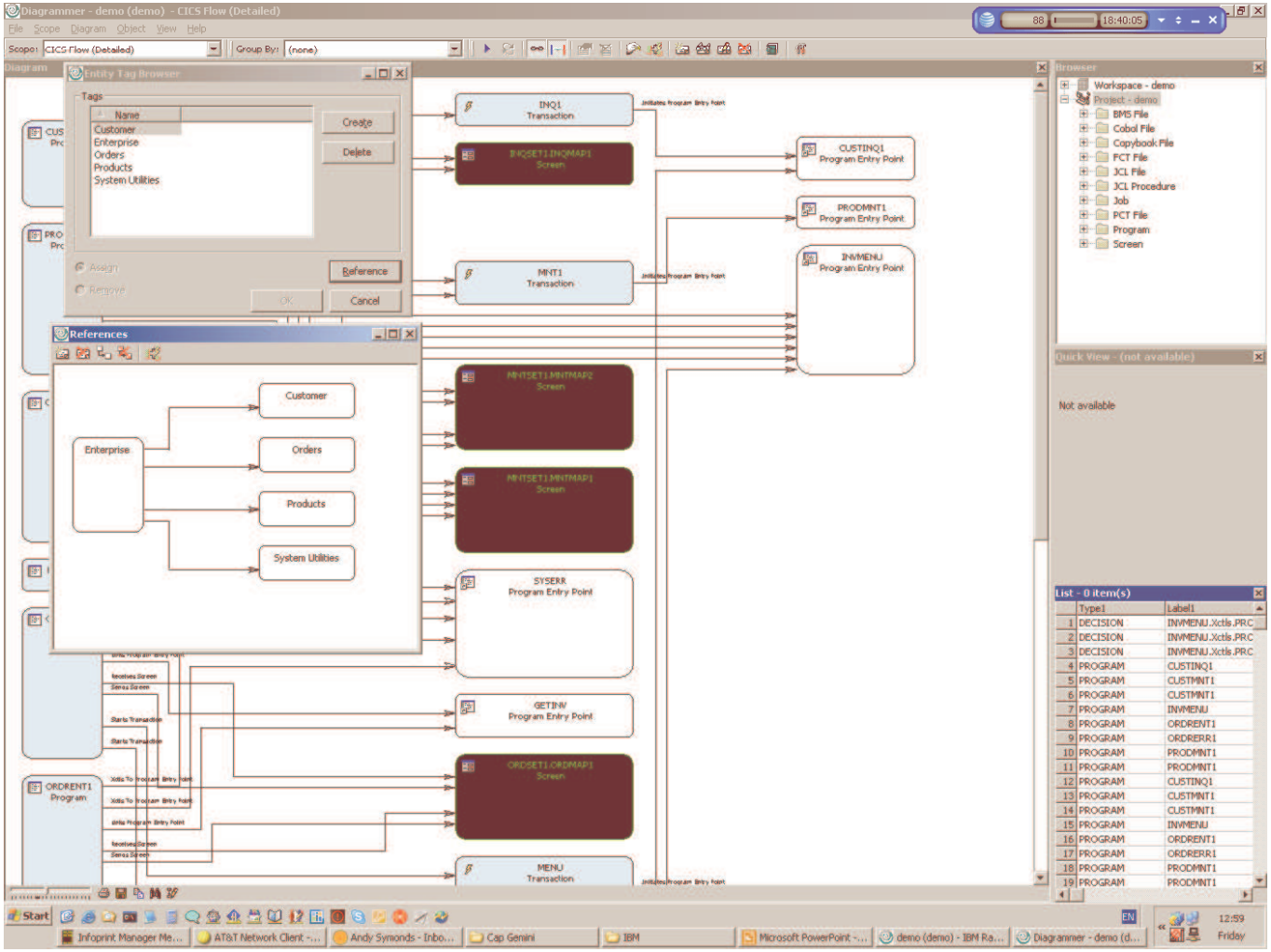


Insurance Order Management System

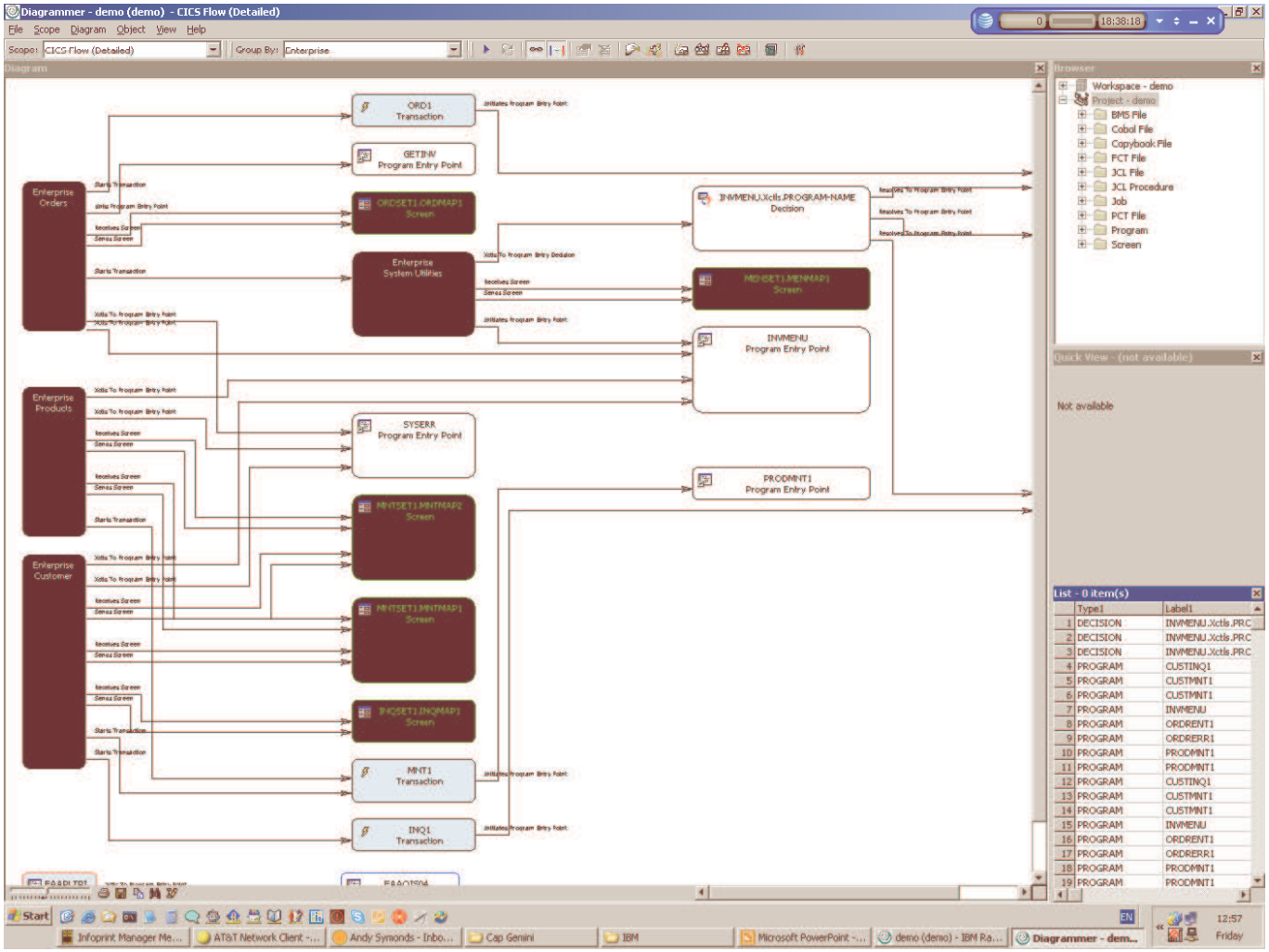
A Standard CICS Flow Diagram



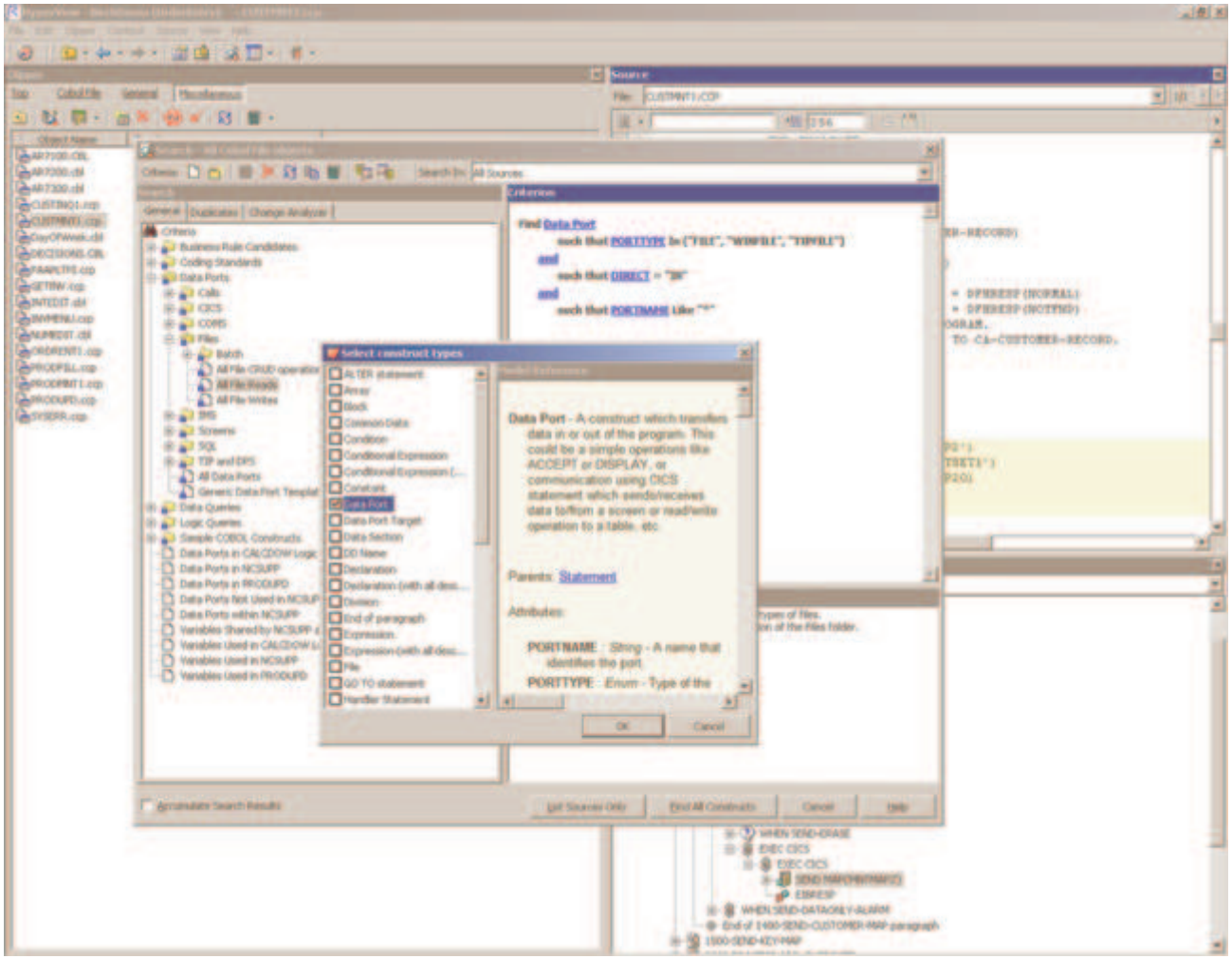
Applying Business Context



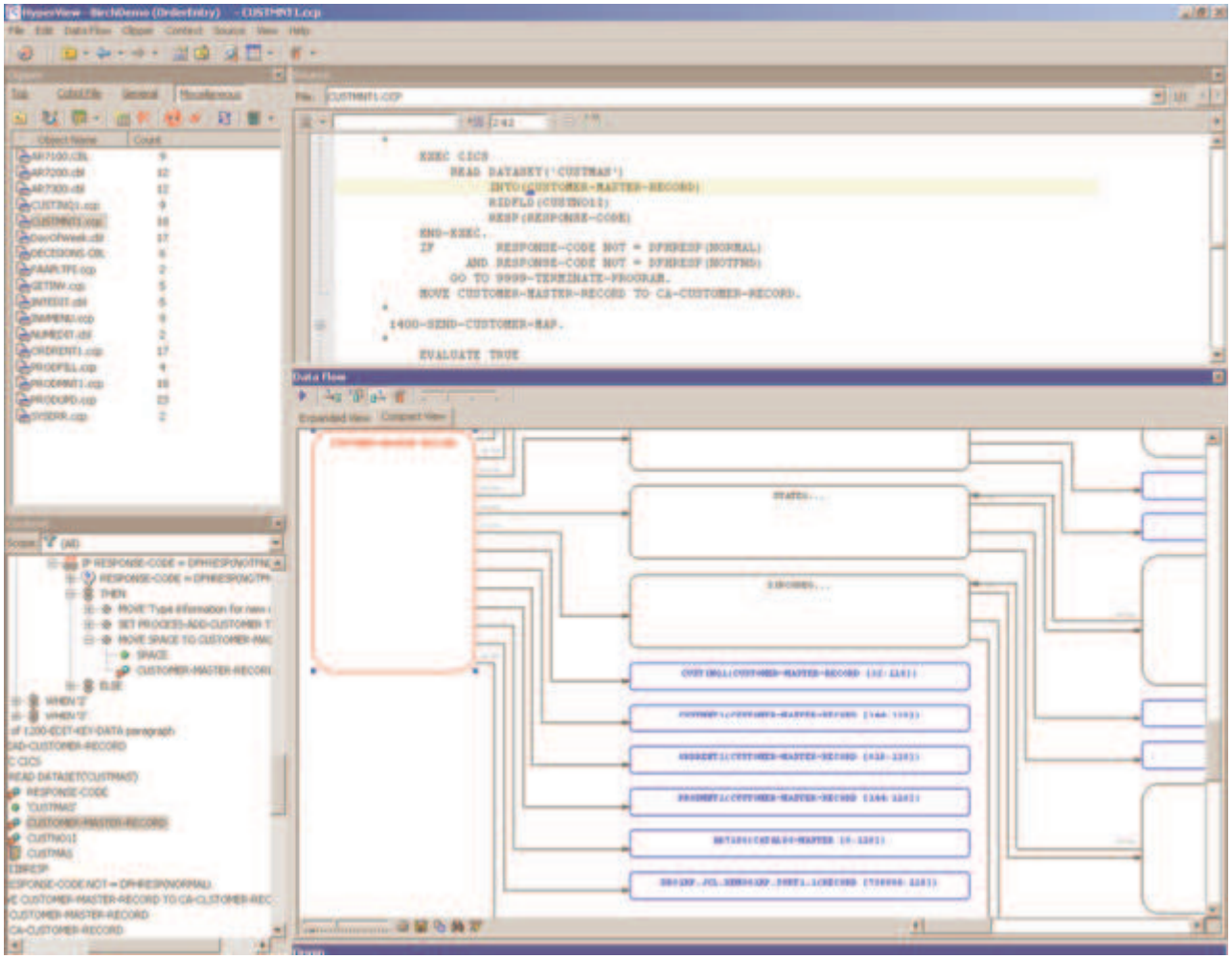
Abstracted view



Powerful Querying to Locate Elements of Value



Trace Data-flow through Systems



Business Rule Extension

Automated Rule Discovery Speeds Collection:

Highly automated paths to uncovering business logic hidden deep within complex applications

Analyst-Centric Functionality Accelerates Identification

Interactive environment dramatically reduces the amount of time to locate logic

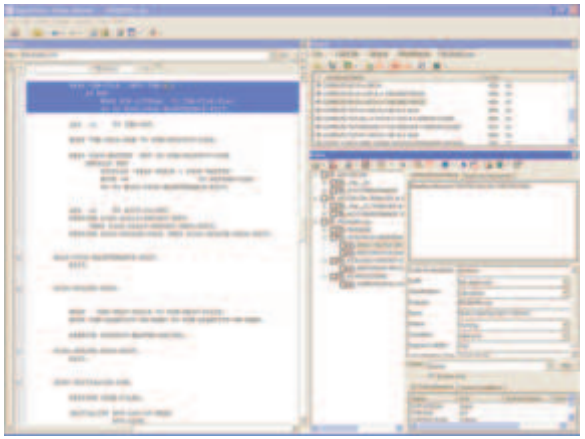
Effective Management of Business Rules:

Organize and document rules to govern operations

Extension of Value of the Rules Repository:

XML-based repository enables rules to be leveraged by other technologies such as Requirements Management or Business Process Modeling

Discovery of hidden business rules is significantly accelerated



Powerful management tools help capture and model processes

Dead and Obsolete Code

There are two types of dead code that exist in application systems:

- *Code that is logically unreachable*
- *Code that is obsolete based on data conditions*

For dead code analysis RTW provides the following:

- *Reports on the dead code and dead data as part of the complexity analysis*
- *Distinguishes between dead code found in programs and dead code found in copybooks*
- *Enables the user to perform sophisticated semantic-based queries to interactively view the dead code*
- *Enables automatic 'surgical' removal of the unwanted code*

For obsolete code RTW is the only tool on the market that can automatically remove it based on data conditions

RTW Architect Extension

Extracting pieces of logic

Structure based - RTW externalizes paragraphs into a new program and makes the links from the old to the new.

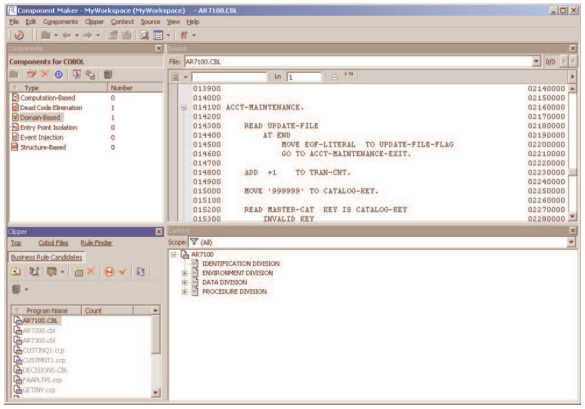
Unraveling Spaghetti

Computation based ("Bottom-Up") – RTW will extract all code that is needed to compute the value of a specific variable into a new component.

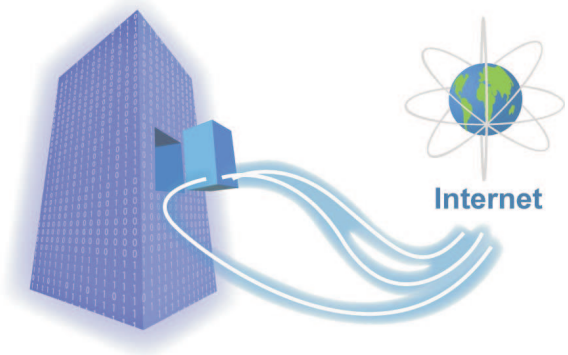
Extracting Specific Business Logic

Domain Based ("Top-Down") – RTW allows the user to extract logic that is executed depending on a specific set of variables having specific values.

Key to decommissioning obsolete business logic



Multiple patented componentization



Componentized business processes can be extended as services

Application Management ROI

Customer-led ROI studies reveal significant savings

Transfer knowledge and encourage resource pooling

Developers are able to get up-to-speed more quickly, allowing even new developers to become productive

Maintain quality and limit risks

Enables developers to ask 'what-if' type questions to avoid cascades of errors

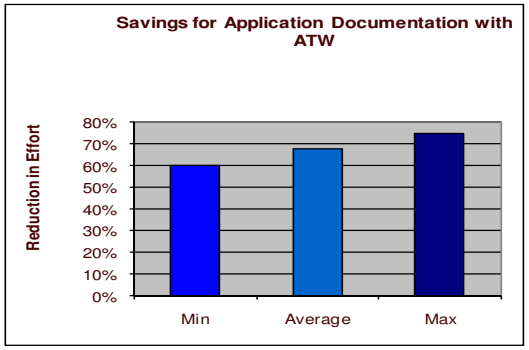
Apply pre-defined queries to maintain coding standards

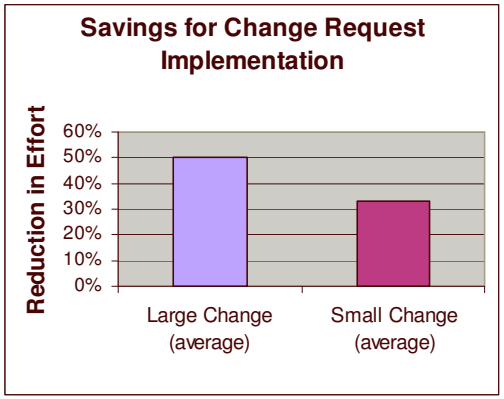
Impact analysis conducted over 87% more rapidly

Increase the effectiveness of change requests

Large change requests required an average of 50% less effort

Overall Cost Savings around 15%-20% can be achieved. More with perfective maintenance.





Modernizing your Assets

Andy Symonds
Enterprise Modernization, Rational
symondsa@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software