Call 425 284 2770

Call 425 284 2770

**The Completed Assessment and Transformation of VistA from MUMPS into Java is Available at No Charge as an Open Source Resource**

TSRI
the software revolution, inc.

# VistA Conversion Pilot and Aftermath : A "Game Changing Approach" to E.H.R. Modernization

**SOA in Healthcare Conference**
**July 14, 2010**

**Philip Newcomb, CEO & Founder**
philip.newcomb@tsri.com
**The Software Revolution, Inc. (TSRI)**
**Kirkland, Washington**
www.tsri.com

# Topics

- The Problem and The Alternative
- OpenVista MUMPS to Java Transformation Blueprint™
- About The Software Revolution, Inc
  - Company, Mission, Value Proposition, Customer/Partners, Methods, Track Record
- Modernization Case Study Overview:
  - Sensor, Eurocat, VistA
- VTS MUMPS to EGL Transformation Blueprint™ Demo
- Modernization Scenarios:
  - Application Portfolio Management, Language to Language and Platform to Platform Transformation, SOA Enablement.
- VHA MUMPS Conversion Pilot
  - March 17, 2005 Kickoff Meeting
  - November 10, 2005 Project Conclusion
- Open Source MUMPS to Java Transformation Blueprint™ of OpenVista®
  - http://www.softwarerevolution.com/blueprints/
- Chapter 12, Veterans Health Administration's VistA MUMPS Modernization Pilot in
  - Information Systems Transformation: Architecture-Driven Modernization Case Studies, Ulrich, Newcomb Morgan-Kaufman, 2010

# About The Software Revolution Inc., (TSRI) www.tsri.com

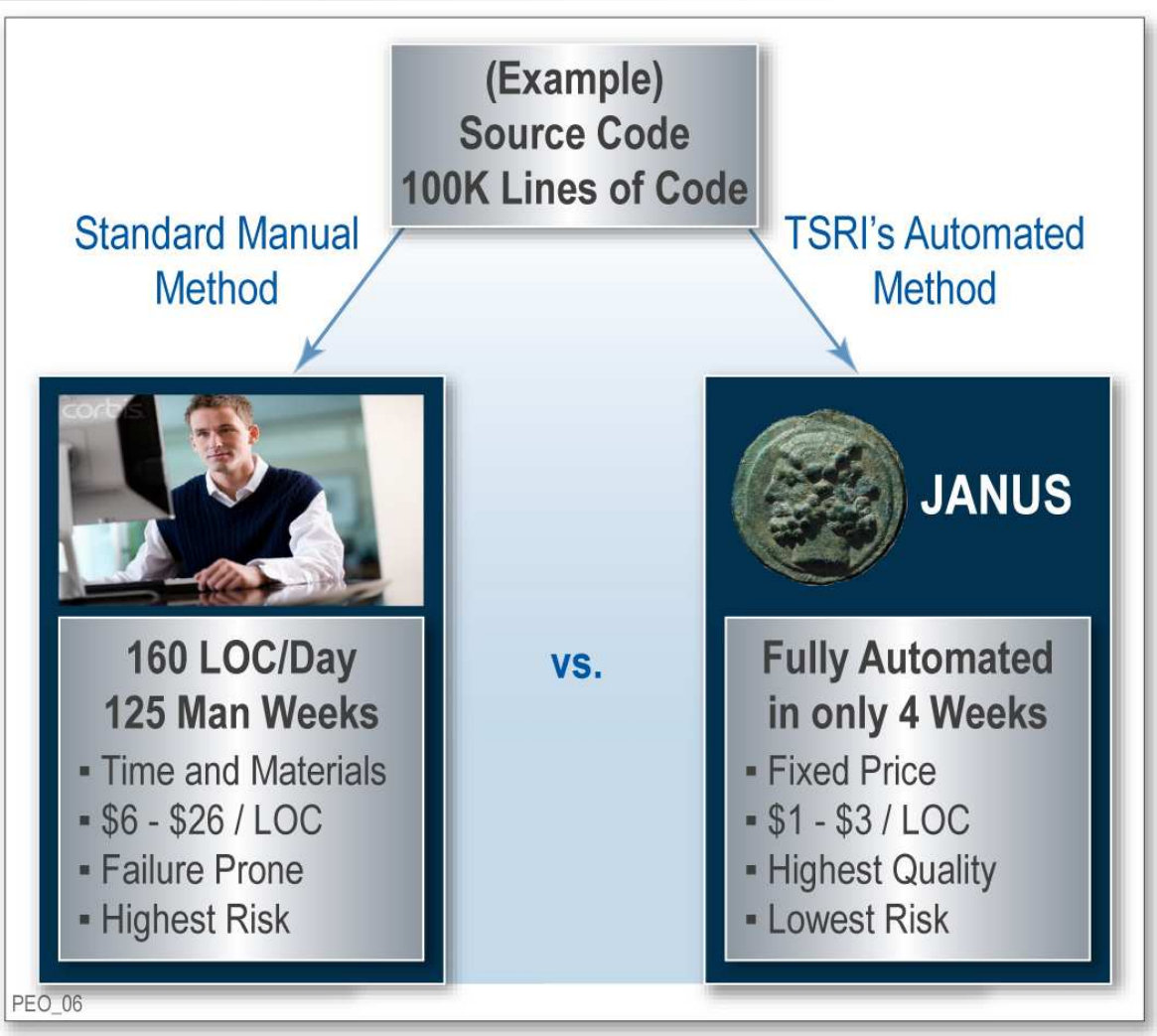
- Offices in Kirkland, Washington
- Technology Rooted in Early Artificial Intelligence Projects
  - 1983 USAF Knowledge Based Software Assistance (KBSA) Program
  - 1988-1994 Boeing Artificial Intelligence Lab
- Member company OMG Architecture Driven Modernization ADM Task Force participating in:
  - Abstract Syntax Tree Meta-Modeling (ASTM)
  - Knowledge Discovery Meta-Modeling (KDM)
  - Structured Metrics Meta-Model (SMM)
- Industrial Awards
  - Northrop Grumman "Small Business of Year" Award 2002
  - Raytheon "Supplier of Value" Award 2005
  - Small Business Administration's 2005 "Administrators Award of Excellence"
- Over 75 Automated Modernization Projects since 2000
  - 100% Project Successfully Completed
  - References Upon Request

# TSRI Mission

- **Modernize Valuable and Often Irreplaceable Legacy System Software and Databases Into**
  - **Modern Platform-Independent Target Languages and Computing Environments**
  - **Without the Requirement for Manual Intervention**

- **Conform To Standards-Based Architecture Driven Modernization (ADM) Technology and Services That Support**
  - **Highly Automated, Cost-Effective Legacy System Modernization**
  - **At a Fixed-Price and Guaranteed Quality**
  - **Between Any Practical Combination of Source & Target Languages**

# Cost, Quality, Risk of TSRI Automated vs. Manual Modernization



# TSRI Customers/Partners













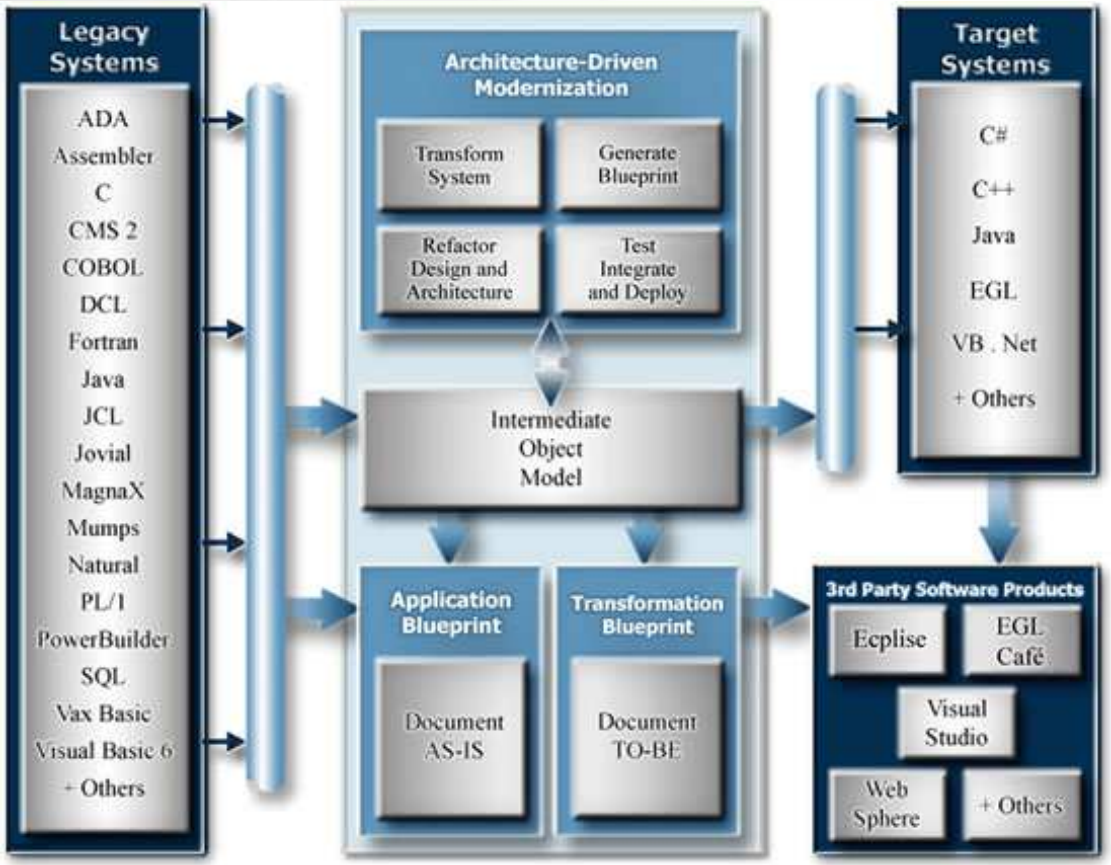




# TSRI Rigorous, Iterative, Agile Method

**Component-Oriented, Model-Based, Architecture-First, and Supports UML and MDA. All Change Is Iterative, Accomplished By Rules Applied To Models.**

**Delivers Highest Quality Modernized Code at 100% Automation Levels.**



# TSRI Track Record

| | Integrator | System | Code | SLOC | TTC |
|---|---|---|---|---|---|
| Civilian | SAIC & Open Source | Fileman, VistA Pilot and OpenVistA | MUMPS to Java | 2.5M | 6 mo |
| | Health Care Insurance Co | BlueCross Provider System | PowerBuilder & Magna X to Java | 170K | 3 mo. |
| | AMDOCS | Billing System | COBOL to C | 5.1M | 7mo. |
| | Thales Air Systems | French Air Traffic Management | Ada to Java | 495K | 12 mo. |
| | Thales Air Systems | Nordic Air Traffic Management | Ada to Java | 541K | 9 mo |
| | Thales Air Systems | Australian Air Traffic Management | Ada to Java | 638K | 9 mo |
| | Unisys | NY State Dept. of Criminal Justice | COBOL Documentation | 308K | 2 mo. |
| | NEA | Grant & Business Systems | COBOL to C++ | 656K | 7 mo. |
| | SAIC | Veteran's Health Administration | MUMPS to Java | 300K | 4 mo. |
| | State of OR | Employee Retirement System | COBOL to C# .Net | 250K | 4 mo. |
| | State of WA | Off. of Super of Public Instruct. | COBOL to C# .Net | 191K | 5 mo. |
| | TriGeo | Sim v4.0 (Internal Product) | Java Docs & Re-Fact. | 370K | 2 mo. |
| | EDS | Proof-of-Concept | P/L 1 to Java | 50K | 7 mo. |
| | CSC | Bureau of Immigration | COBOL to C++ | 17K | 3 mo. |
| | Boeing | ALCA - Czechoslovakia | Jovial to C++ | 9K | 2 mo. |
| Military | LMCO | P-3C | Ada - C++ | 656K | 14 mo. |
| | ITT | BMEWS - Cobra Dane | Ada/Fortran - C++ | 380K | 8 mo. |
| | Raytheon | Satellites | Ada/Fortran - C++ | 284K | 5 mo. |
| | L-3 | VTT | Ada - C++ | 77K | 3 mo. |
| | LMCO | SAC Strategic Planning System | Ada - C++ | 40K | 2 mo. |
| | DSR | E-2C ACFT | Ada - C++ | 20K | 5 mo. |
| | USAF | CAMS | COBOL Docs | 1M | 6 mo. |
| | NGC | REMIS | COBOL - C++ | 400K | 7 mo. |
| | Dyncorp | WCSRS | COBOL - C++ | 90K | 5 mo. |
| | ITT | BMEWS - ROSA | Fortran/C Docs | 2M | 4 mo. |
| | Raytheon | Patriot Missile | Fortran - C++ | 200K | 6 mo. |
| | Litton PRC | Strategic Air Command | Fortran - C++ | 50K | 4 mo. |
| | Raytheon | WDAC | Fortran - C++ | 40K | 1 mo. |
| | SAIC | EOSS | VAX Basic - Java | 38K | 5 mo. |
| | TRW | MILSTAR | Jovial to C++ | 143K | 1 mo. |
| | USAF | F-16 Decis | Jovial to C++ | 50K | 4 mo. |

# Groundbreaking Architecture Driven Modernization Projects Case Study Overviews

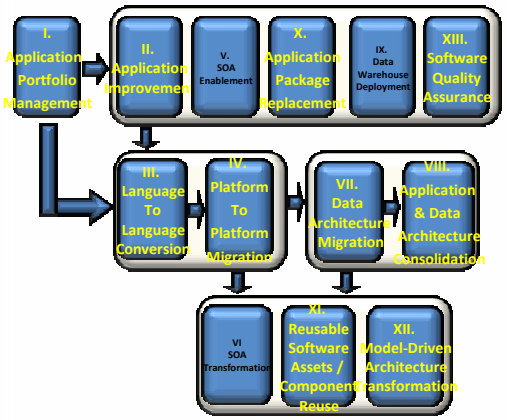
# TSRI Case Study: ITT Corporation SENSOR

## Project Summary

**ITT Corporation awarded a sole-source contract to The Software Revolution, Inc. (TSRI) for Assessment of Radar Open System Architecture (ROSA) C and FORTRAN and modernization of the COBRA DANE Ada Radar Calibration System (SCRS) of the Ballistic Missile Early System (BMEWS) under the Air Force's System Engineering & Sustainment Integrator (SENSOR) program.**

***Project Contacts:***

***Upon Request***




## Services

- Application Blueprint of Legacy "As-Is" Ada Code with UML Design Code
- Automatic Transformation and Re-Factoring of Ada into Object-Oriented Java and C++
- Architecture-Driven Semi-Automatic Incremental Re-Factoring of Target Design
- System Integration, Testing & Implementation Support
- Transformation Blueprint™ of Ada to Java "Showing Side-by-Side" Code with UML Design Documentation

## ADM Scenarios

**1, 2, 3, 4, 7, 8, 10, 11, 12, 13**

## Results

- Project successfully completed in several increments (Jul-04 Aug-05)
- Extended *JANUS*™ Ada, C and Fortran grammars systems.
- *JANUS*™ rules engine parser was tuned to address & ROSA C and FORTRAN and the Cobra Dane Ada
- Fully modernized 380,300 LOC of highly complex mission-critical Cobra Dane Ada to both C++ as well as Java (after Real Time Java performance was determined to be acceptable)

# TSRI Case Study: Thales EATMS

THALES

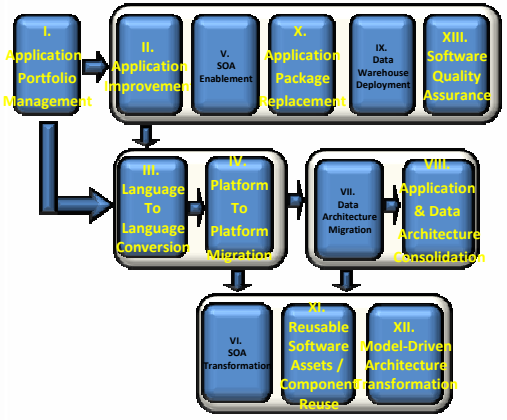
*Project Contacts:*

*Available Upon Request*

The project's detailed case study is published as Chapter 5: "Modernization of the Eurocat Air Traffic Management System"

in:

Information Systems Transformation: Architecture-Driven Modernization Case Studies By William M. Ulrich and Philip H. Newcomb Published by Morgan Kaufmann ISBN: 978-0-12-374913-0 Copyright Feb 2010




## Project Summary

**The Software Revolution, Inc. (TSRI) performed multiple contracts for Thales Air Systems to modernize several versions of Eurocat into the next generation European Air Traffic Management System (EATMS).**

## Services

- Application Blueprint of Legacy "As-Is" Code with UML Design
- Ada to C++ and Java comparison Pilot
- Transformation and Refactoring of Ada into RT Java
- Architecture-Driven Semi-Automatic Incremental Re-Factoring of Target Design
- System Integration & Test Support
- Transformation Blueprint™ of Final Ada to RT Java "Showing Side-by-Side" Code with UML Design Documentation

## ADM Scenarios

**1, 2, 3, 4, 8, 10, 11, 12, 13**

## Results

- Addressed the Flight Profile Library, Minimum Safe Altitude Warning System, and Air-Ground Data Processor modules of Eurocat
- *JANUS*™ rules engine parser was tuned to address the legacy Ada code of these EATMS modules
- Fully modernized nearly 1,700,000 LOC of Ada code to Real Time Java
- Effort included extensive automated and semi-automated re-factoring to meet precise, mission-critical coding standards
- Achieved Ultra High Assurance Air Traffic Control Software compliant with EATMS

# TSRI Case Study: VHA VistA Modernization Pilot

## Project Summary

**The Software Revolution, Inc. (TSRI) with SAIC performed a sole-source contract for the VHA in 2005 to demonstrate the feasibility of automated modernization of MUMPS into Java. TSRI subsequently (2010) transformed all 2.1 Million lines of OpenVistA® from MUMPS into JAVA, publishing it as an open source Transformation Blueprint™ accompanying release in Feb 2010 of Ulrich & Newcomb's Information System Transformation: Achitecture-Driven Modernization Case Studies book.**

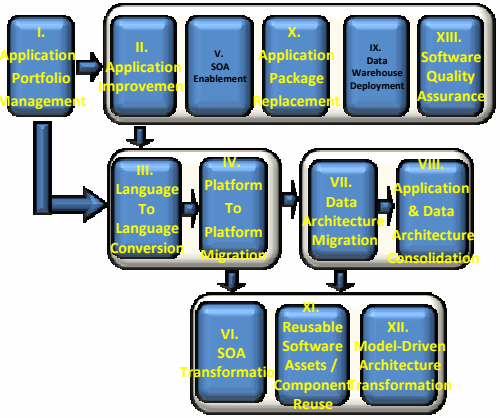
### *Project Contacts:*

***Upon Request***

The project's detailed case study is published as Chapter 12: "Veterans Health Administration's VistA MUMPS Modernization Pilot"

in:

Information Systems Transformation: Architecture-Driven Modernization Case Studies By William M. Ulrich and Philip H. Newcomb Published by Morgan Kaufmann ISBN: 978-0-12-374913-0
Copyright Feb 2010




## Services

- Application Blueprint of Legacy "As-Is" MUMPS Code with UML Design
- Automated Transformation and Re-Factoring of MUMPS into Java
- System Integration, Testing & Implementation Support
- Transformation Blueprint™ of MUMPS to Java Showing Side-by-Side Code with UML Design Documentation

## ADM Scenarios

**1, 2, 3, 4, 6, 7, 8, 11, 12, 13**

## Results

- **Project successfully completed in 5 months (ended Aug-05)**
- ***Created a new JANUS*™ MUMPS parser**
- ***JANUS*™ rules engine parser was tuned to address the VHA MUMPS**
- **Fully modernized 200,000+ LOC of highly complex MUMPS to Java during Pilot.**
- **Generated Transformation Blueprint™ for 2.1 MLOC of MUMPS of OpenVistA® as future open source EHR *reference* model.**

# The $11 Billion 9 Year Problem

## Source: http://www.gao.gov/products/GAO-08-805

### Veterans Affairs: Health Information System Modernization Far from Complete; Improved Project Planning and Oversight Needed
### GAO-08-805 June 30, 2008

**The Department of Veterans Affairs (VA), through its Veterans Health Administration (VHA), provides health care for more than 5 million veterans each year. In 2001, VHA began an initiative, HealtheVet, to modernize its current medical information system. GAO's objectives were to determine the status of the modernization, VA's overall plan for completing it, and how VA is providing oversight to ensure the success of the initiative. To conduct this review, GAO analyzed project documentation and interviewed officials responsible for the development and implementation of the new system. As of June 2008, the HealtheVet initiative has these eight major software development projects under way. One project is to further develop the Health Data Repository, a database of standardized health data. This database, which is currently operational, is not yet complete; additional types of health data remain to be standardized and added to the repository. Four application projects are currently in development. One application project is in the planning stage. Two projects are being pursued to enhance current systems, prepare them for transition to HealtheVet, and develop new applications. From 2001 through fiscal year 2007, VA reported spending almost $600 million for these eight projects. The time frame for completing the projects and the HealtheVet system as a whole was 2012, but the projected completion date has now been delayed until 2018. The department has a high-level strategy for HealtheVet, in which the remainder of the initiative is to be completed incrementally in phases (referred to as "blocks"), but it does not have a comprehensive project management plan to guide the remaining work. This work is considerable: the department plans to replace the 104 applications in its current medical information system with 67 modernized applications (of which 5 are currently in development, as described), 3 databases, and 10 common services (general software functions, such as messaging and security, on which application software can call as needed). In view of this scope, the importance is increased of developing a comprehensive project management plan that includes, among other things, an integrated schedule that considers all dependencies and defines subtasks to ensure that deadlines are realistic. Another important component of such planning is determining the resources necessary to accomplish tasks throughout the life cycle of the initiative. In April 2008, VA provided an $11 billion cost estimate for completion of HealtheVet; however, it has not yet independently validated this estimate. Having a validated cost estimate is essential to improve the accuracy of cost, schedule, and performance management. Without an integrated plan that includes independently validated cost estimates, VA increases the risk that HealtheVet could incur cost increases and continued schedule slippages and not achieve its intended outcomes. Various levels and types of oversight are currently being provided for the HealtheVet initiative by business owners, developers, and departmental information technology governance boards. However, the business owners have not yet implemented a complete governance structure, several key leadership positions within the developers' organization are either vacant or filled with acting personnel, and the governance boards have not yet scheduled critical reviews of HealtheVet projects. Until all elements of governance and oversight are in place, the risk to the success of the HealtheVet initiative is increased.**

**aa**

# Alternative *Proven* Standards-Based ADM Solution Scalability Demonstration

- ✓ **200KLOC Functional Equivalence Demonstration Was Completed in 2005 for The Volunteer Time Keeping System and Fileman into Java/J2EE/Oracle/Weblogic (demonstrated to MacFarland, Luigardt, Kolodner)**
- ✓ **Fully Automated Conversion of 2.1 Million Lines of OpenVistA® into Java in February 2009 – (Scalability Demonstration)**
- ✓ **Transformation Blueprint™ Published to Open Source in February 2010**
  - ✓ **99.999% Automated Conversion**
  - ✓ **Comprehensive VistA Software Metrics:**
    **Redundancy, Deadcode, Duplicate Code, Complexity. Size**
  - ✓ **The Complete UML Design for The Entire OpenVistA Syste**
  - ✓ **Side-By-Side MUMPS and Java with Hyperlinks between source and target code and design and architecture models.**

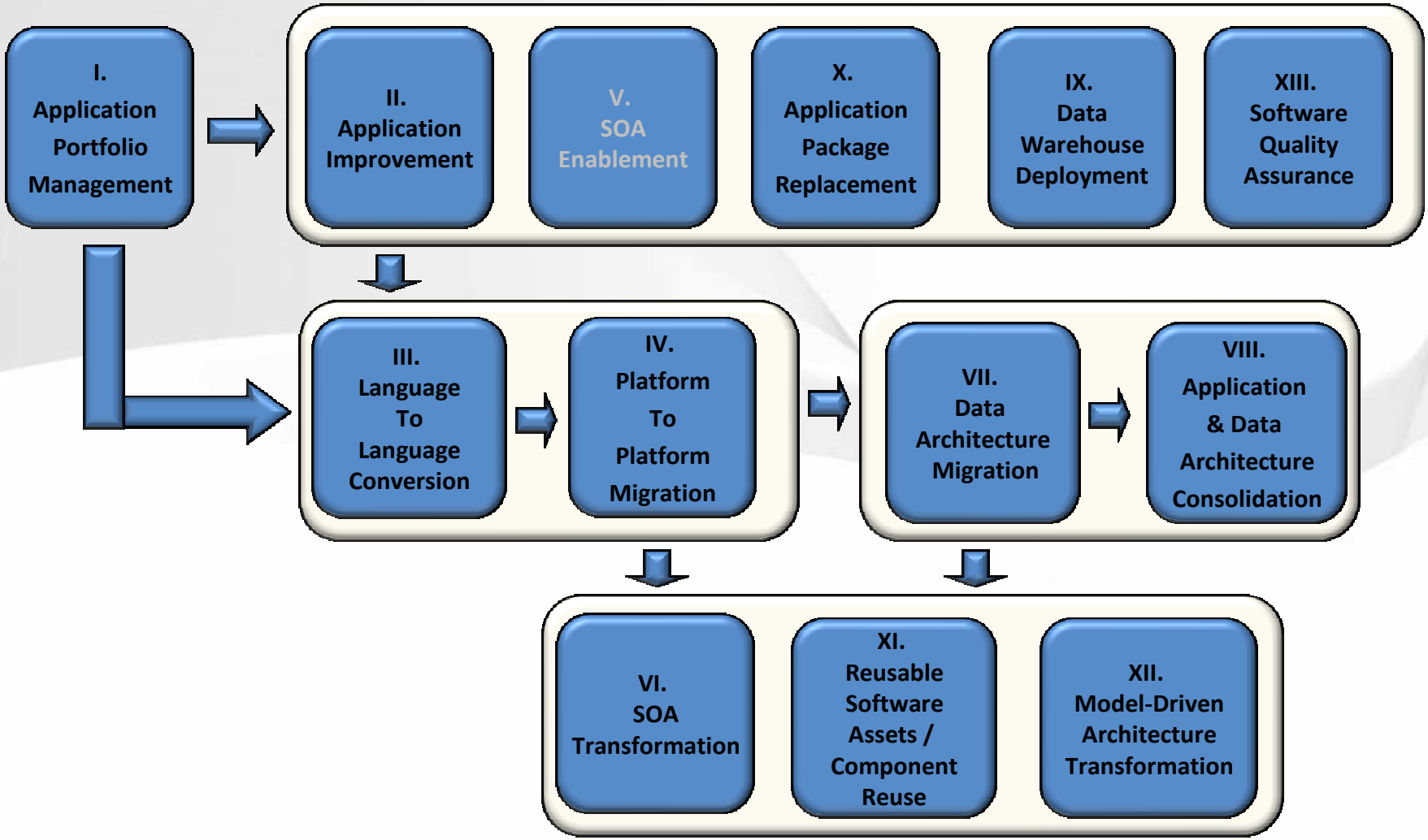
➢ **Where to Get The Free Solution**

**http://www.softwarerevolution.com/blueprints/index.php/open-source-blueprints/o-section/vha-open-vista-mumps-to-java**

**http://www.softwarerevolution.com/blueprints/**

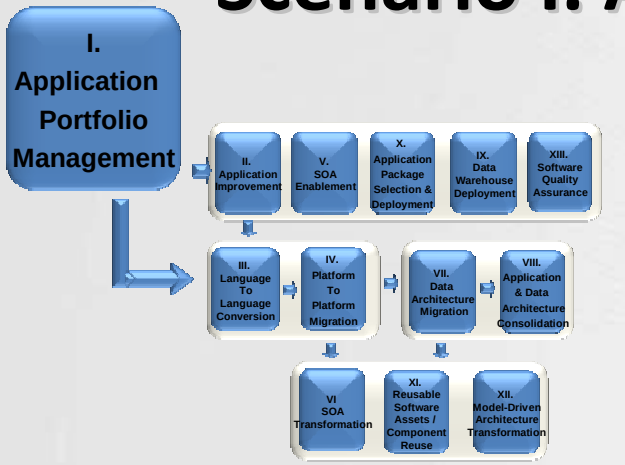
**WAIT STOP!!! Free is Bad … Don't Bother With It. Spend the $11Billion and Take 9 Years Instead.**

# Combine Modernization Scenarios



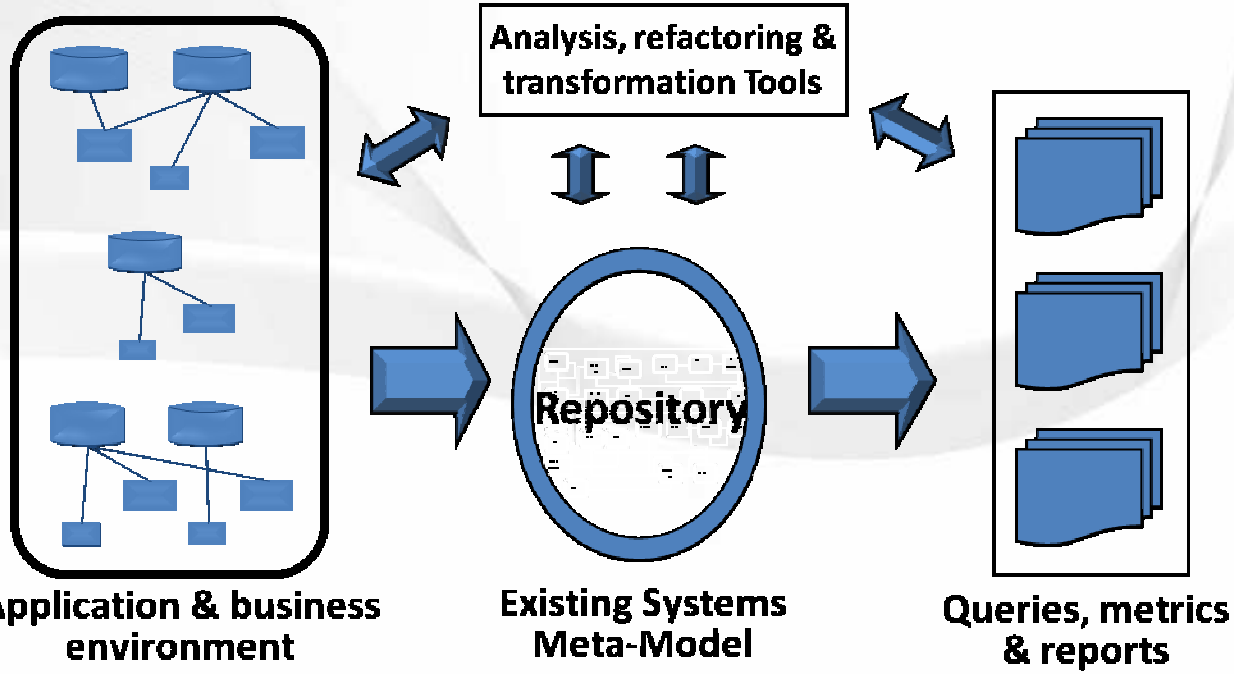

* Object Management Group (OMG) Architecture Driven Modernization Task Force (ADMTF) Modernization Scenarios

# Scenario I. Application Portfolio Management



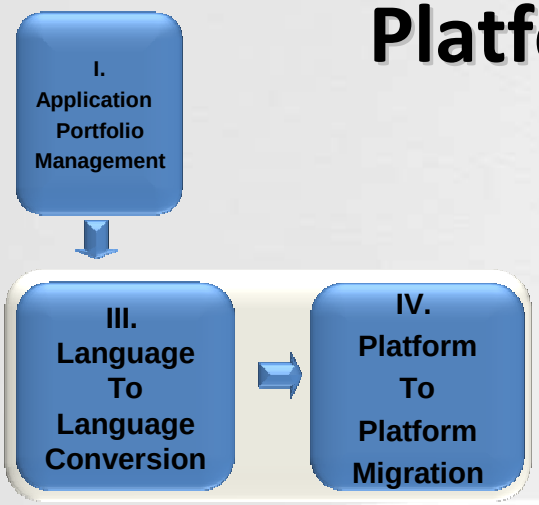

**Objective: Establish multi-dimensional knowledge base for managing & transforming applications.**

| Tasks |
|---|
| Catalog Application Inventory, Structure and Relationships between Components |
| Catalog Architecture of Application Layers, Flow and Interactions among Components and Application Boundaries |
| Catalog Dead Code, Obsolete Code and Dead-End Code |
| Catalog Definitions, Facts and Rules about Data |
| Catalog Derivation, Triggering and Validation Rules |
| Catalog Business Processes and Usage Scenarios |
| Catalog Flaws and Vulnerabilities |
| Group Applications with Like Data, Functionality and Requirements |
| Create Portfolio with Step-By-Step Migration Scenario and Roadmap, Work Break Down Structure, Rough Order of Magnitude Estimates and Cost Estimates. |




Source: William Ulrich / TSG, Inc

# ADM Scenario III & IV: Language-to-Language and Platform to Platform Migration

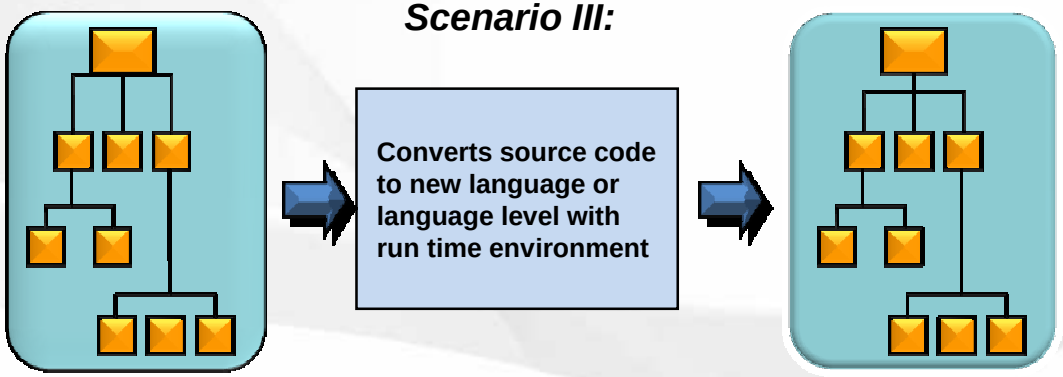



**Objective: Convert applications to new language and / or run time platform. Scenarios can be performed separately or together.**

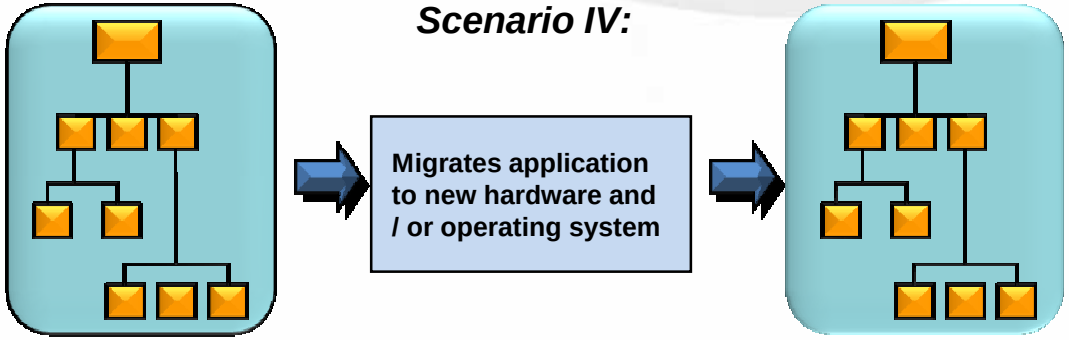
## Tasks

| | | |
|---|---|---|
| **Top Down Language To Language (L2L) Conversion Scenarios** | | |
| 5GL to 4GL | | |
| 5GL to 4GL to 3GL | 4GL to 3GL | |
| 5GL to 4GL to 3GL to 2GL | 4GL to 3GL to 2GL | 3GL to 2GL |
| | | |
| **Bottom Up Language To Language (L2L) Conversion Scenarios** | | |
| 2GL to 3GL | | |
| 2GL to 3GL to 4GL | 3GL to 4GL | |
| 2GL to 3GL to 4GL to 5GL | 3GL to 4GL to 5GL | 4GL to 5GL |

***Scenario III:***

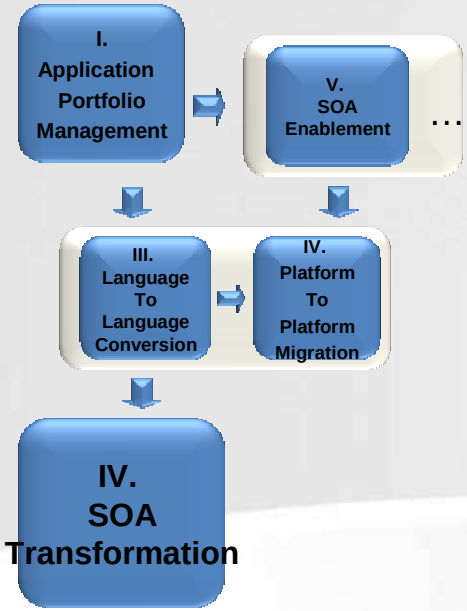



***Scenario IV:***



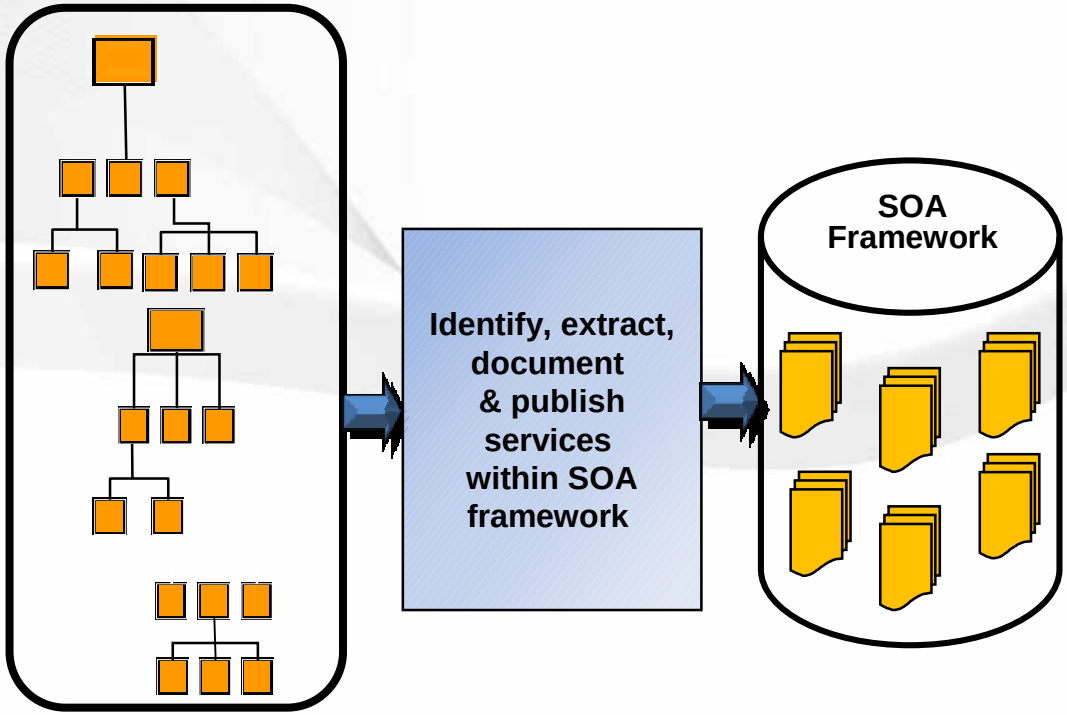

Source: William Ulrich/TSG,Inc

# ADM Scenario VI: Services Oriented Architecture (SOA) Transformation




**Objective: Create a framework for constructing and interlinking back-end systems with the goal of making applications more agile.**

| Tasks |
|---|
| **Transform by Product Lines (what system produces)**<br>Modernize Functions that Create Products into SOA |
| **Transform by Application Layers (how system is layered)**<br>Modernize Application Layers into SOA |
| **Transform by Business Process**<br>Modernize Functions supporting Business Process into SOA |
| **Transform by Business Rules**<br>Modernize Application Logic into Business Rule Engines and SOA |
| **Transform by Data Sets (how data is organized)**<br>Map Legacy Data Bases into SOA Enabled Repositories |




**Existing applications**

**Repository identifying reusable services**

Source: William Ulrich/TSG,Inc

# 2005 VHA MUMPS Conversion Pilot
# Project Kickoff Meeting March 17, 2005


## Topics

- Introductions
- Relevant Projects
- Project Goals
- Roles
- TSRI Transformation Process
- Progress to Date from R&D
- Schedule and Deliverables
- Software and GFE Requirements

## Introductions

Call Participants

- SAIC
  - Alex Bravo — SAIC Project Manager
  - George Hou — VA Account Manager
  - Bruce Custis — Deputy Account Mgr
  - Cheryl Campbell — Operations Manager
  - Frank Wilcox — Chief Architect
  - Frank Schillinger — Mgr, Requirements and Modeling
  - Dennis Eisenstein — Testing
  - Wade Brown — VA Contracts Officer
- The Software Revolution Incorporated (TSRI)
  - Philip Newcomb — CEO
  - Randy Doblar — COO
  - Roger Knapp — Conversion Task Lead

## Relevant Projects

SAIC/TSRI's experience:

- >25 successful legacy system automated conversions
- Support of the Military Health System HIS
- Support of the Bi-directional Health Information Exchange (BHIE) – DoD/VA Data Sharing program
- Pre-Pilot VistA MUMPS Conversion R&D project

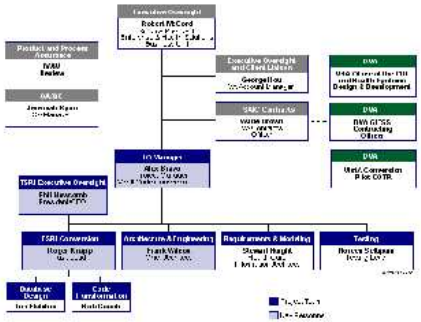
# March 17, 2005 Kick-off Summary (continued)

## Project Goals

- Fusion of respective company strengths to meet VA's modernization objectives
- Automated 100% MUMPS Code Converted to Java
- Clear separation of application logic, data handling elements and presentation elements
- Document transformed VistA system
- Web-enable Front End / User Screens
- Integration of transformed system into VA's Service Oriented Architecture and HealtheVet applications

## Project Organization and Roles

- SAIC
  - Provide M/FileMan/Cache Technology Domain Expertise
  - Project Management
  - IV&V
- TSRI
  - Provide Modernization Expertise
  - Transformation Process and Tool Set



# March 17th Kick-Off Summary (continued)

## Progress Report on VistA Transformation R&D Project

- Phase I and Phase II code has been provided to TSRI
- 100% of Phase I and Phase II code is parsing, constraining and mapping into the IOM
- Challenges overcome:
  - **Parsing**
    - Spaces and new lines can have significance
    - Case insensitive and non-reserved keywords
    - First/Follow conflicts
    - Undocumented commands
  - **Constraining**
    - Extended scope for routine labels
    - Forward references to routines and their locals
    - Argument-less DO's with inner blocks
    - References to undefined routines and their locals
  - **Mapping**
    - Indirection required IOM extension
    - Argument-less DO's with inner blocks
      - Due to tree structure, special code required to avoid retranslation
    - For commands
      - Iteration over an explicit list
      - Termination state
      - Simplifications for loops over statically known quantities

## Schedule and Deliverables

Phase I – Voluntary Timekeeping

## Software and GFE Requirements

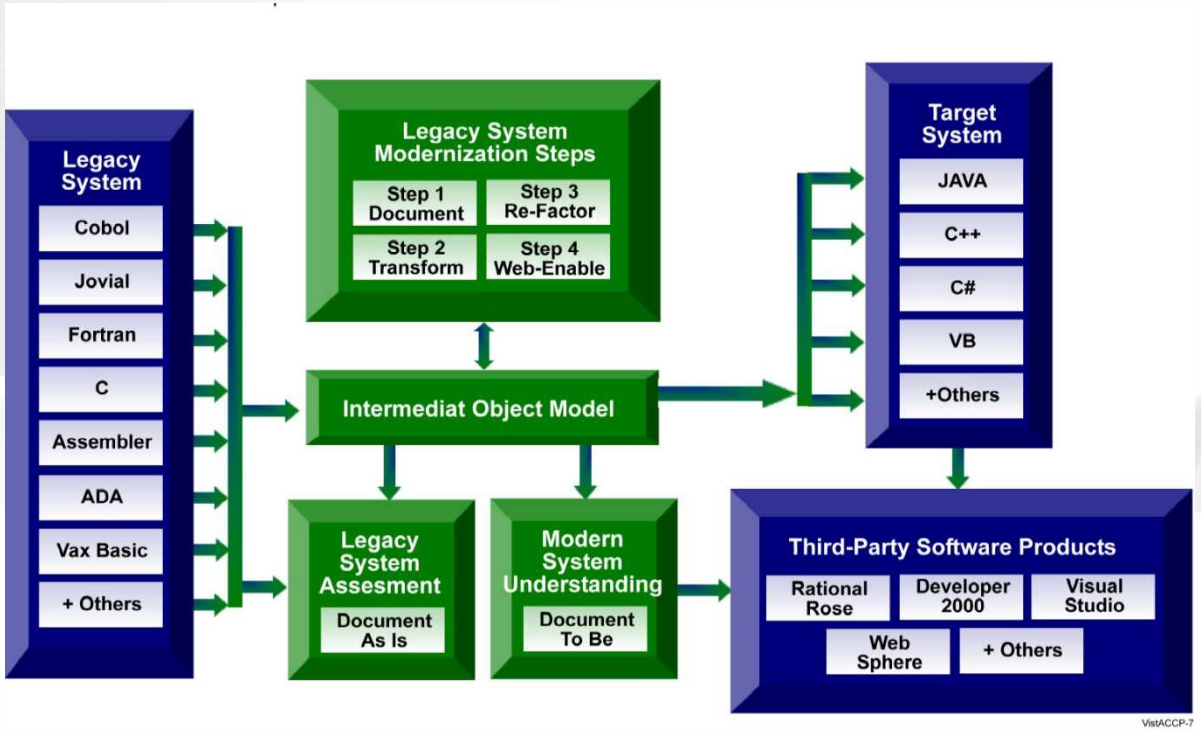
- SAIC will obtain required Rational Rose, Requisite Pro, WebLogic and Red Hat Linux licenses
- All documentation on the VistA products to be converted through provision of the full VistA suite of software as issued under the Freedom of Information Act (FOIA).
- "Core Specifications for Rehosting Initiatives".
- Layout and performance attributes of the VA WAN.
- Access to a test environment in the OI Test Lab to allow the contractor to either review or create test scripts that are applicable to the VistA application targeted for conversion.
- Access to the Visual Architecture that describes the HealtheVet-VistA system, including all identified services.
- Provide sufficiently mature Common Services components.
- Most recent version of Cache and the Cache M Data Access Repository Tool (M-DART) from Intersystems Corporation.
- Most current versions of the user and technical manuals for the Voluntary Time Keeping and Employee Time & Attendance applications for the purposes of extracting requirements and preparing use cases and test scripts.
- Make OI technical staff available as may be required for consultation and issue resolution.

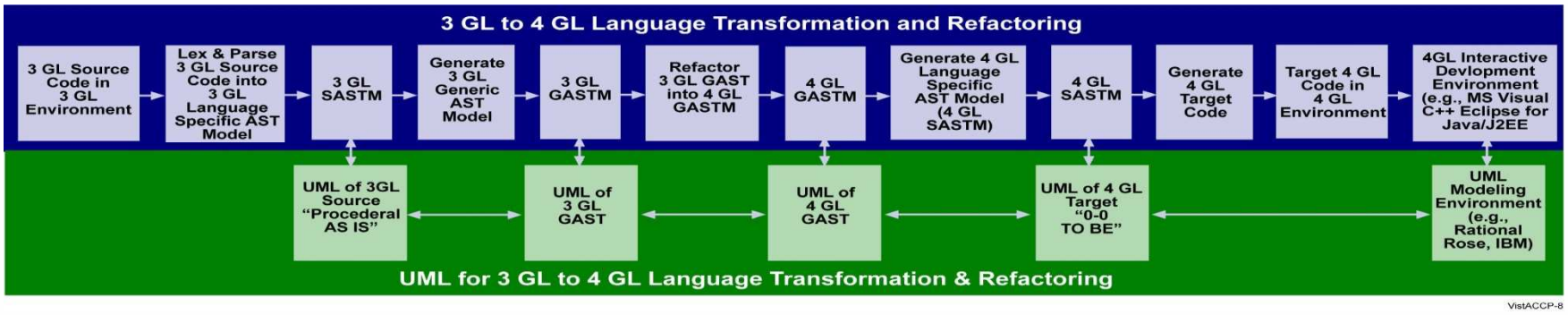
# March 17, 2005 Project Goals

- Fusion of respective company strengths to meet VA's modernization objectives
- Automated 100% MUMPS Code Converted to Java
- Clear separation of application logic, data handling elements and presentation elements
- Document transformed VistA system
- Web-enable Front End / User Screens
- Integration of transformed system into VA's Service Oriented Architecture and HealtheVet applications

# March 17, 2005 Transformation Process

- Assessment
- Transformation
- Refactoring
- Web-Enablement
- Service Oriented Architecture



# March 17, 2005 Progress Report on VistA Transformation R&D Project

- Phase I and Phase II code has been provided to TSRI
- 100% of Phase I and Phase II code is parsing, constraining and mapping into the IOM
- Challenges overcome:

**Parsing**

- Spaces and new lines can have significance
- Case insensitive and non-reserved keywords
- First/Follow conflicts
- Undocumented commands

**Constraining**

- Extended scope for routine labels
- Forward references to routines and their locals
- Argument-less DO's with inner blocks
- References to undefined routines and their locals

**Mapping**

- Indirection required IOM extension
- Argument-less DO's with inner blocks
  - Due to tree structure, special code required to avoid retranslation
- For commands
  - Iteration over an explicit list
  - Termination state
  - Simplifications for loops over statically known quantities

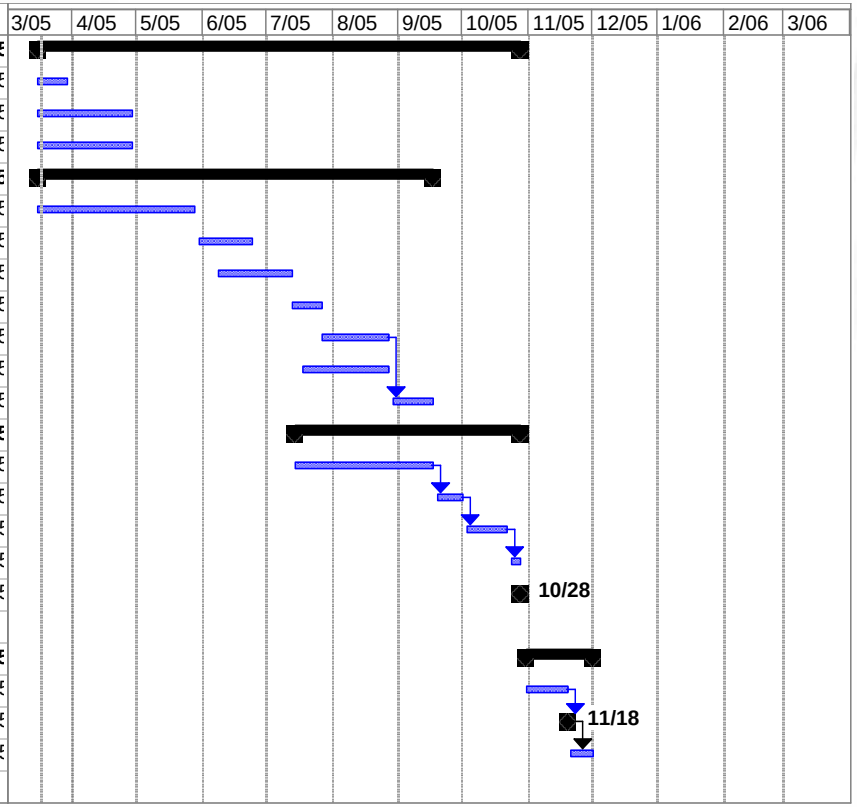
# March 17th 2005 Software and GFE Requirements

- SAIC will obtain required Rational Rose, Requisite Pro, WebLogic and Red Hat Linux licenses
- All documentation on the VistA products to be converted through provision of the full VistA suite of software as issued under the Freedom of Information Act (FOIA).
- “Core Specifications for Rehosting Initiatives”.
- Layout and performance attributes of the VA WAN.
- Access to a test environment in the OI Test Lab to allow the contractor to either review or create test scripts that are applicable to the VistA application targeted for conversion.
- Access to the Visual Architecture that describes the HealtheVet-VistA system, including all identified services.
- Provide sufficiently mature Common Services components.
- Most recent version of Cache and the Cache M Data Access Repository Tool (M-DART) from Intersystems Corporation.
- Most current versions of the user and technical manuals for the Voluntary Time Keeping and Employee Time & Attendance applications for the purposes of extracting requirements and preparing use cases and test scripts.
- Make OI technical staff available as may be required for consultation and issue resolution.

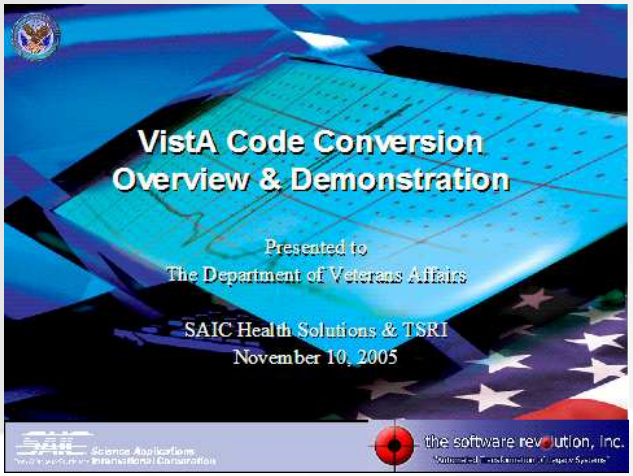
# March 17, 2005 Schedule and Deliverables

## Phase I – Voluntary Timekeeping

| ID | Task Name | Start | Finish |
|---|---|---|---|
| 1 | **Phase 1** | **3/15/05** | **10/28/05** |
| 2 | **CLIN 1001 - Management Plans** | 3/15/05 | 3/28/05 |
| 3 | **CLIN 1002 - Requirements** | 3/15/05 | 4/28/05 |
| 4 | **CLIN 1003 - Modeling Use Case Diagrams** | 3/15/05 | 4/28/05 |
| 5 | **CLIN 1004 - Code Conversion** | **3/15/05** | **9/16/05** |
| 6 | Transformation Set Up (CLIN 1004a) | 3/15/05 | 5/27/05 |
| 7 | Code Transformation (CLIN 1004b) | 5/30/05 | 6/23/05 |
| 8 | Auto Re-Factoring (CLIN 1004c) | 6/8/05 | 7/12/05 |
| 9 | Integration (CLIN 1004d) | 7/13/05 | 7/26/05 |
| 10 | CSOW tasks 2,3,4,8,21 (CLIN 1004e) | 7/27/05 | 8/26/05 |
| 11 | Documentation (CLIN 1004f) | 7/18/05 | 8/26/05 |
| 12 | Integration Testing (CLIN 1004g) | 8/29/05 | 9/16/05 |
| 13 | **CLIN 1005 - Testing** | **7/14/05** | **10/28/05** |
| 14 | Prepare Test Plans and Scripts | 7/14/05 | 9/16/05 |
| 15 | Functional Testing | 9/19/05 | 9/30/05 |
| 16 | Prepare test reports | 10/3/05 | 10/21/05 |
| 17 | IV&V review and certification | 10/24/05 | 10/27/05 |
| 18 | Deliver Initial Converted Code to Government (CLIN 1005a) | 10/28/05 | 10/28/05 |
| 19 | | | |
| 20 | **Government IV&V/Exercise Period** | **10/31/05** | **11/30/05** |
| 21 | Government conducts IV&V and reviews test results | 10/31/05 | 11/18/05 |
| 22 | Government issues defect report to SAIC | 11/18/05 | 11/18/05 |
| 23 | SAIC corrects defects and delivers final converted code (CLIN 100 | 11/21/05 | 11/30/05 |
| 24 | | | |



# November 10, 2005 MUMPS Conversion Pilot Project Conclusion




## Topics

- Introductions and Administrative Logistics
- Project Goals
- Team Overview
- How did we get here – Conversion as a Modernization Option
- TSRI's Transformation Methodology and JANUS™ toolset
- CSOW Tasks and Challenges of a MUMPS Conversion
- Schedule and Accomplishments
- Conversion Demonstration using TSRI's JANUS™ Tool Suite
- Next Steps in the Conversion Process
- Phase II Considerations
- COBOL Demonstration of TSRI's JANUS™ Tool Suite
- Case Studies

2

## Project Goals

- **Demonstrate Feasibility of Converting MUMPS Application Code into Java/J2EE**
- **Achieve 100% Automated Conversion**
- **Create Clear Separation Between:**
  - Application Logic
  - Business Rules
  - Data Handling
  - Presentation Elements
- **Document Transformed VistA System**
- **Investigate Automated Generation of Use Case & Requirements Documentation**
- **Web-Enable Front End / User Screens**

3

## Demonstration of the TSRI *JANUS*™ Tool Suite

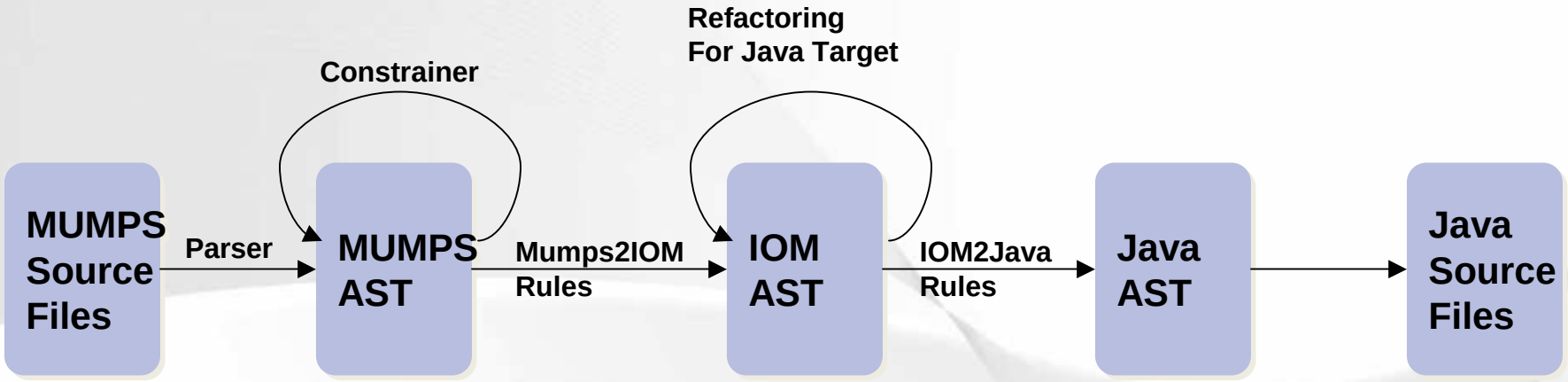
Selected Web-enabled Modules of the Voluntary Timekeeping Application

12

# November 10, 2005 Project Goals

- **Demonstrate Feasibility of Converting MUMPS Application Code into Java/J2EE**
- **Achieve 100% Automated Conversion**
- **Create Clear Separation Between:**
  - Application Logic
  - Business Rules
  - Data Handling
  - Presentation Elements
- **Document Transformed VistA System**
- **Investigate Automated Generation of Use Case & Requirements Documentation**
- **Web-Enable Front End / User Screens**

# November 10, 2005 Transformation Process




- **AST – Abstract Syntax Tree**
  - In Memory Tree Base Representation of System
- **Constrainer**
  - Adds Symbol Linkage & Expression Types to AST Nodes
- **IOM – Intermediate Object Model**
  - TSRI Developed Language Neutral Model into Which All Source Languages are Mapped & Target Languages Emitted

# November 10, 2005 Major CSOW Tasking

- **Convert Two MUMPS Applications into Java**
  - Phase I – Voluntary Timekeeping
    - Small Simple Application
    - Relatively Independent and Self-Contained
    - Roll and Scroll User Interface
    - Device Handler Interfaces
    - Uses FileMan ^DIC and ^DIE APIs extensively
  - Phase II (optional) – Employee Time and Attendance
    - Large Complex Application
    - Multiple Points of Integration with Other VistA Applications
    - ScreenMan User Interface
- **Produce Requirements and "Use Case" Documentation**
- **Document Business Rules**

# November 10, 2005 MUMPS Challenges

- **For Loop Constructs**
- **MUMPS Indirection**
  - Use of Do and Goto with Indirection Arguments
  - G @X, D @X
  - Indirection Assignment of Variables
  - S @X=@Y
- **Xecute Command**
  - Embedded Goto Statements in Execute String
- **FileMan APIs**

# November 10, 2005 Accomplishments Since Inception of Pilot

- **100% Automated Conversion of MUMPS to Java**
- **Web Enabled Front End for Voluntary Timekeeping**
- **Relational Data Base Implementation**
- **Implementation of M Language Challenges**
  - MUMPS Indirection
  - Xecute Command
- **Conversion of FileMan APIs**
  - ^DIC
  - ^DIE
  - Requires Transformation of Nearly All of MUMPS Language Features

# November 10, 2005 Demonstration of the TSRI JANUS™ Tool Suite

Selected Web-enabled Modules of the Voluntary Timekeeping Application

# November 10, 2005 Considerations for Phase II

- **Automated Generation of Use Case Documentation**
- **Software Architecture Refactoring**
  - Java Servlets - Moving Application Logic/Business Rules into Client Side Web Forms
  - FileMan Re-Engineering
  - RDBMS Schema Optimization
- **Transform a Clinical Application**

Call 425 284 2770

Call 425 284 2770

# *BACKUP SLIDES :*

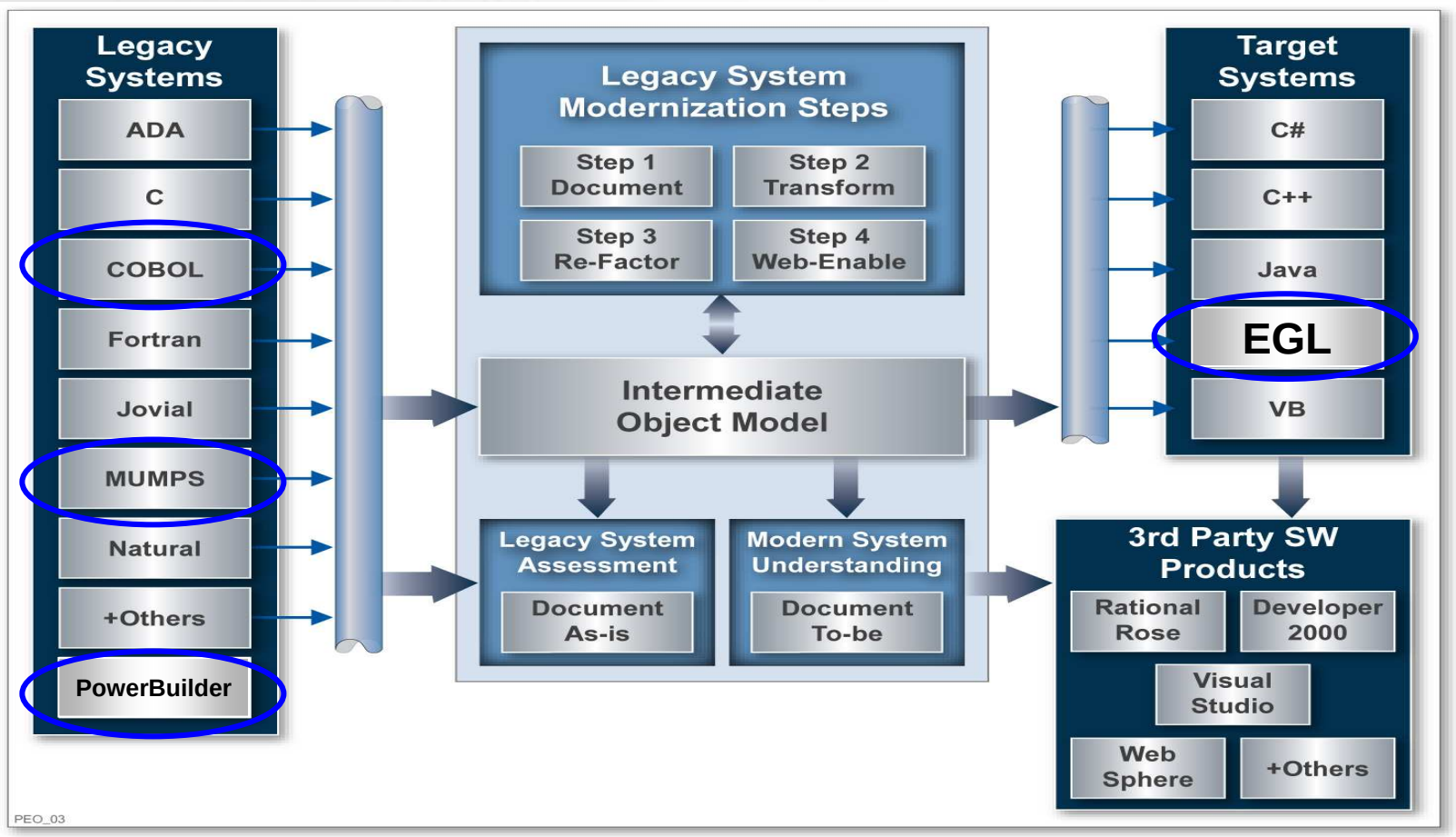
# *Why CONVERT VistA from MUMPS TO EGL?*

**Philip Newcomb, CEO & Founder**
philip.newcomb@softwarerevolution.com

**The Software Revolution, Inc. (TSRI)**
**Kirkland, Washington**
www.softwarerevolution.com

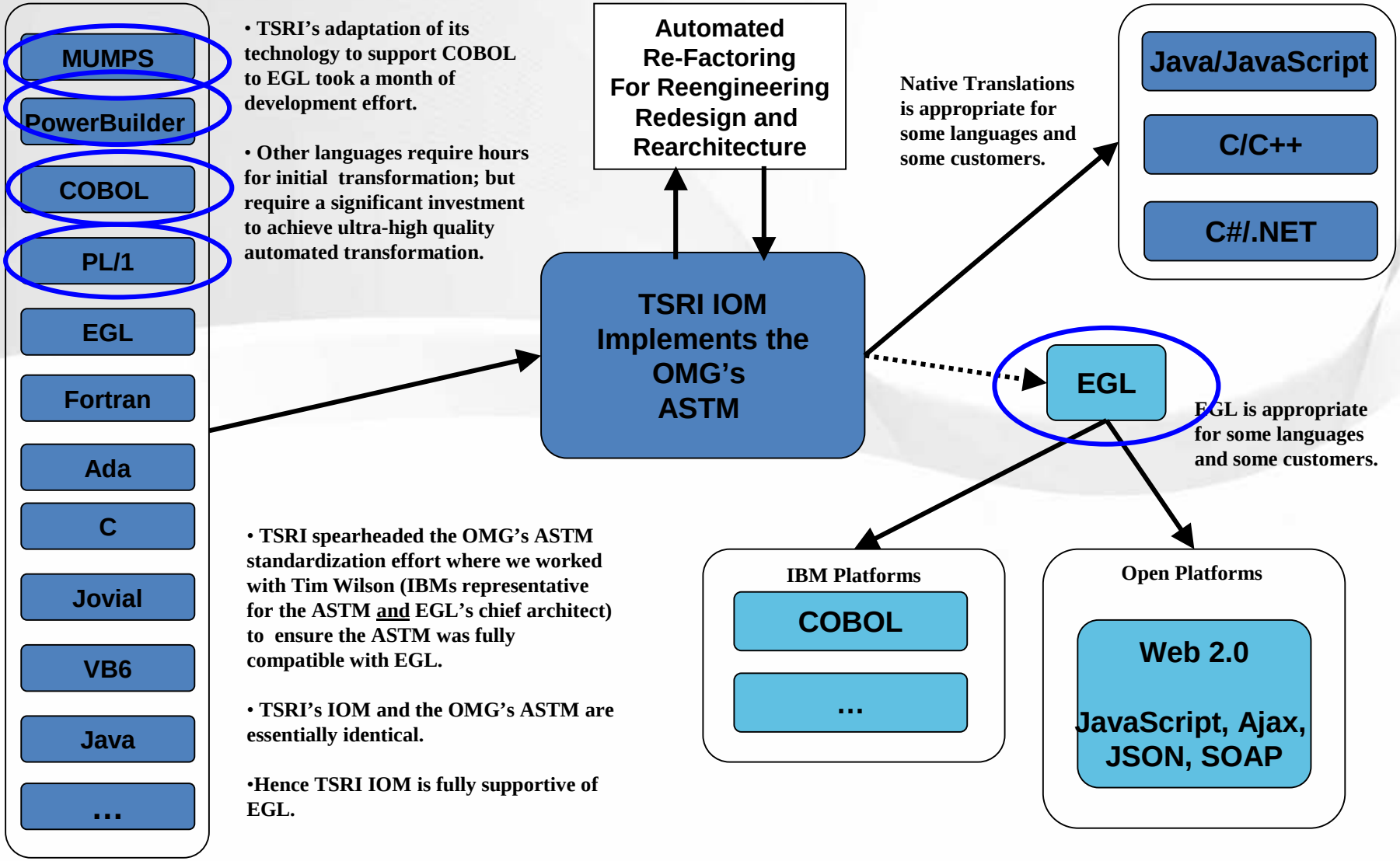
# TSRI's Multi-Scenario Modernization Framework

**TSRI has demonstrated 100% automated conversion into EGL from MUMPS, PL/1, Powerbuilder and COBOL. Additional languages will be added very quickly.**



# Why Target EGL?

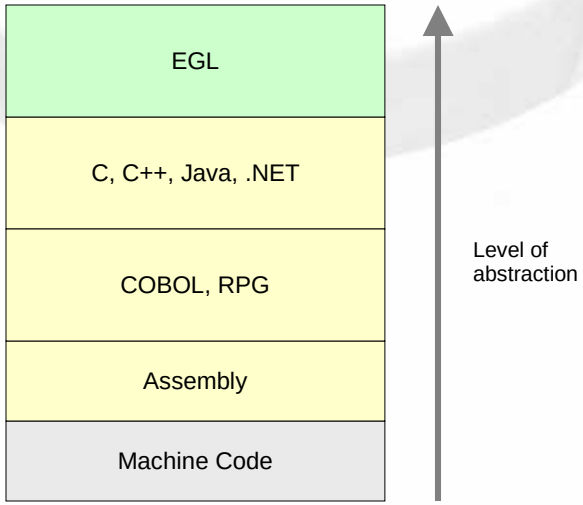
**Because EGL Expands migration options and EGL supports platform-neutral application generation per the OMG's ADM approach to modernization**



# How will VA developers benefit from programming in EGL?

**Because EGL Simplifies Application Development. Developers Can Use (*and learn*) Just One Really Powerful Language and Generate All The Other Languages from EGL.**

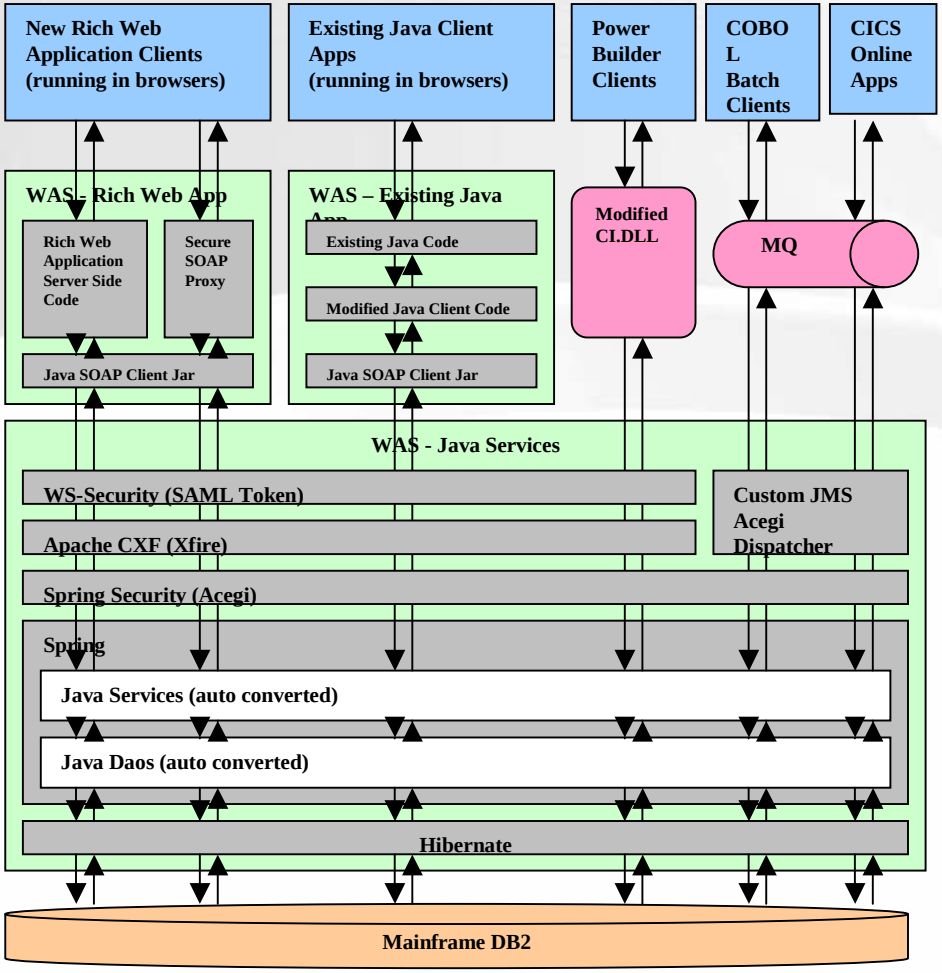
- **Web 2.0 and SOA ares built-in into EGL**
- **EGL is a higher-level programming language designed for developing powerful, modern applications**
  - **Its goal is to shield developers from complexities typically associated with Web 2.0 application development**
- **EGL is a true cross-platform, cross-tier language targeted at all types of developers**
- **EGL Provides flexible deployment options**
  - **Compiles to Java and JavaScript**

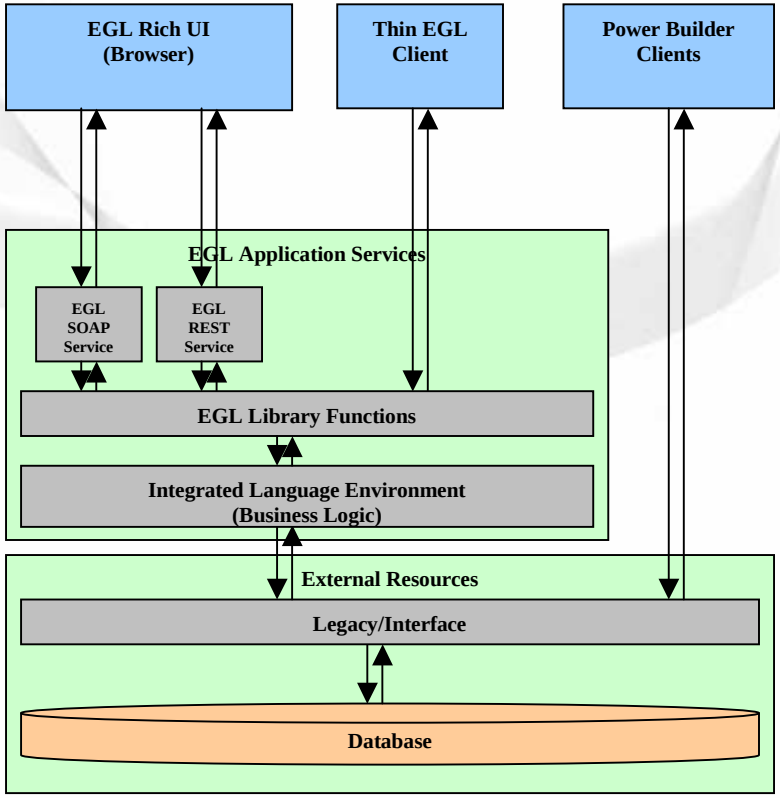
# How does EGL a Simplify Web 2.0 and SOA Enablement?

**Because for some languages, such as Power Builder, which are used for developing client server applications, the resulting Web 2.0 code is vastly easier to maintain when expressed as EGL. EGL eliminates the need for developers to understand Web 2.0 and SOA complexities.**

## PB to Web 2.0 *Without* EGL



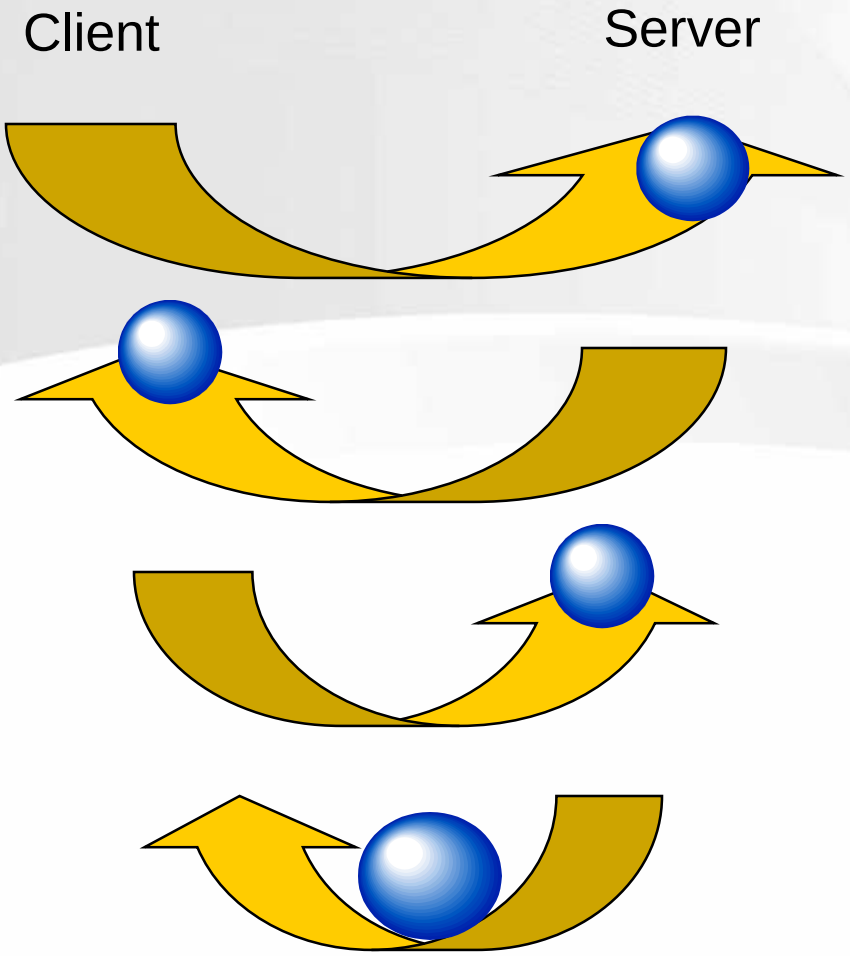

## PB to Web 2.0 *With* EGL



# Why target EGL for Web 2.0 and SOA Development?

**Because Web 2.0 and SOA Applications are notoriously complicated, and EGL makes Web 2.0 applications development simple.**

**Web 2.0 makes Web applications as powerful as desktop applications.**




## Mainframe computing

"Dumb" green screen clients
Omnipotent big mainframe servers

## Client-server computing

"Smart" Personal Computer clients
Simple file and database servers

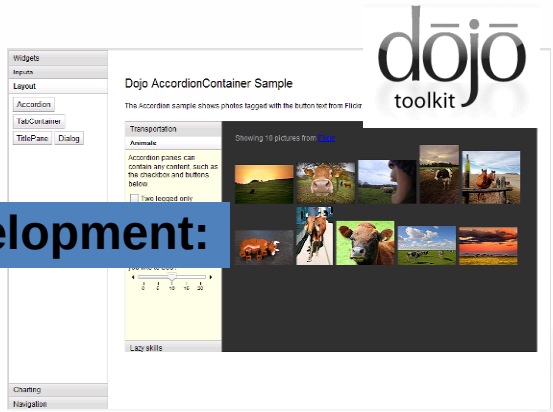
## Web (1.0) computing

Light Web Browser clients
Rich application and database servers

## Web 2.0 computing

Rich Internet Application clients
Lighter application and database servers

# Do developers really like EGL for Web 2.0 and SOA development?

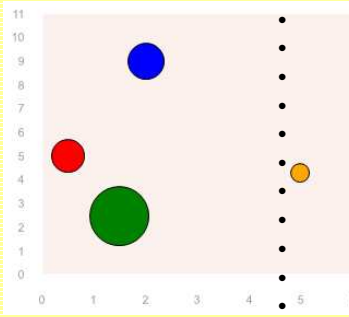



## Yes. Because EGL greatly simplifies Web 2.0 widget development:

- Dojo is a the most popular and powerful open source JavaScript library used for Web 2.0 development
- IBM has created a sample EGL Dojo widget library that enables developers to easily use Dojo widgets within their EGL applications
  - No knowledge of Dojo or JavaScript required
  - Fits within the EGL programming model
  - Demonstrates extensibility of EGL architecture
  - Enables faster development
  - Available as a sample on the EGL Café and is included in EGL Community Edition

### Example of a EGL widget

```
new DojoBubbleChart{
    themeColor = dojo.widgets.DojoLib.CHART_COLOR_THEME_ORANGE,
    minX = "0", maxX = "7",
    minY = "0", maxY = "11",
    width = "400",
    height = "300",
    data = data
}
data BubbleChartData[] = [
    new BubbleChartData { x=0.5, y=5,   size=1.4, color="red",    tooltip="Gas" },
    new BubbleChartData { x=1.5, y=2.5, size=2.5, color="green",  tooltip="Mortgage" },
    new BubbleChartData { x=2,   y=9,   size=1.5, color="blue",   tooltip="Electric" },
    new BubbleChartData { x=5,   y=4.3, size=0.8, color="orange", tooltip="Cable" }
];
```



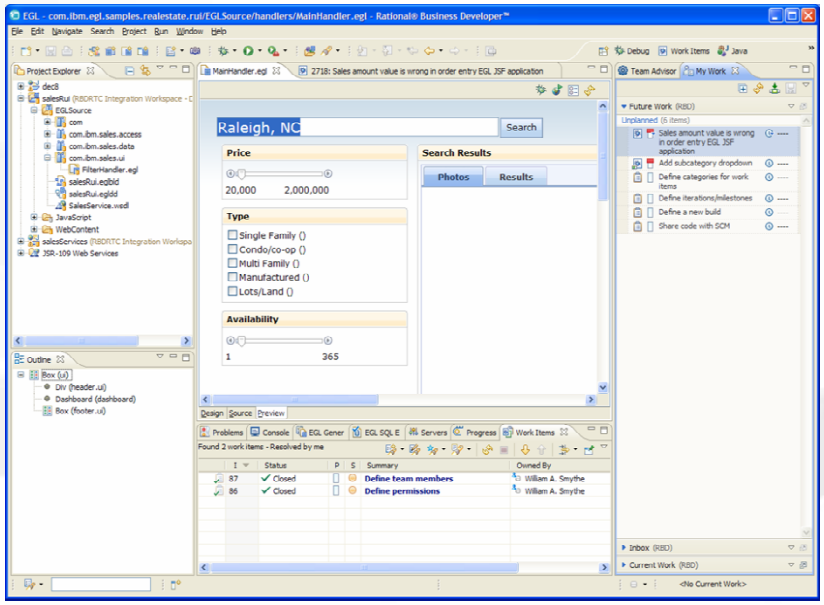
### EGL RUI provides simple versions of many Web 2.0 Widgets

- Accordion Container
- Bar Graph
- Bubble Chart
- Button
- Check Box
- Color Palette
- Combo Box
- Content Pane
- Context Menu
- Currency Text Box
- Date Text Box
- Dialog
- Grid
- Horizontal Slider
- Line Graph
- Menu
- Pie Chart
- Progress Bar
- Radio Group
- Tab Container
- Time Text Box
- Title Pane
- Tree

# How Do Open Source Developers Benefit From EGL?

**Because EGL is the most powerful Eclipse-based Open Application Generator available and Eclipse is the most used Open Source IDE.**

- **EGL is available in multiple Rational offerings:**
  - Rational Business Developer (RBD)
  - Rational Developer for System z with EGL
    - IDE for System z development; includes COBOL, PL/I editing tools, and more
  - Rational Developer for System i for SOA Construction
    - IDE for IBM i development; includes RPG and COBOL editing tools, and more
- **EGL's recent integration with Eclipse propels EGL into the Open Source Arena through its tight integration with the most down-loaded platform neutral IDE. EGL gets powerful feature sets from IDE, including:**
  - Visual and source editors
  - Code completion, templates, and snippets
  - Service generation for database tables
  - SQL visualization and editing
  - References and declarations
  - Open on selection
  - Refactoring
  - Cheat sheets and dynamic help

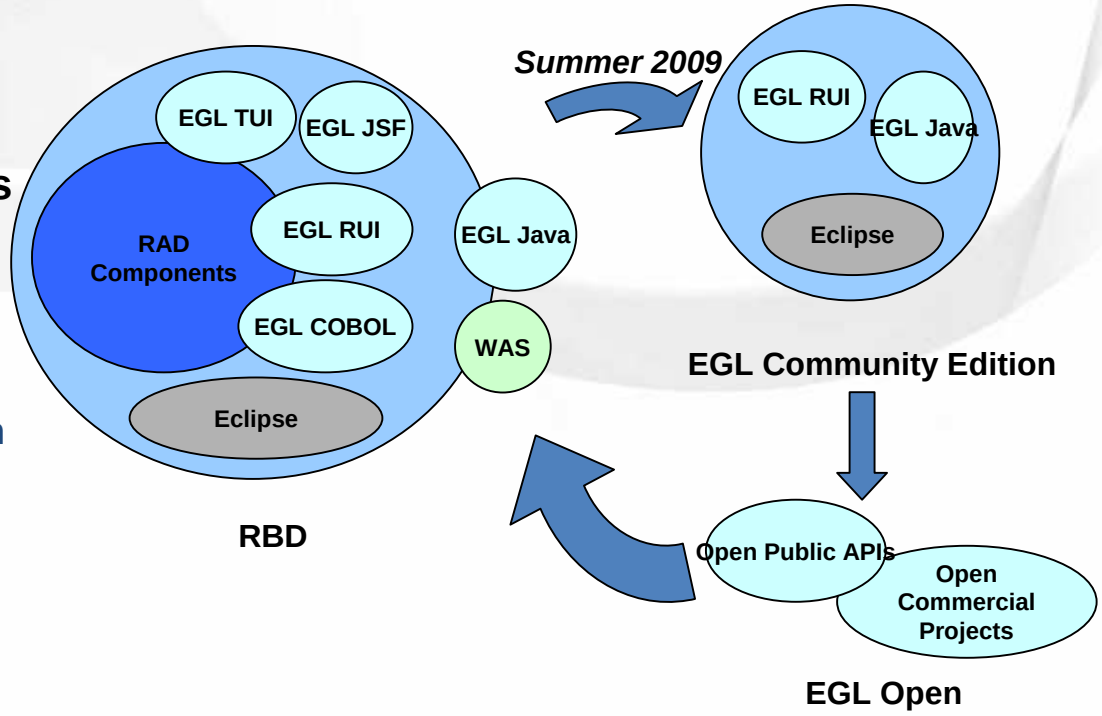



- **EGL is tightly Integrated with Rational Team Concert / Jazz**
  - So EGL development teams can take advantages of the capabilities provided by Jazz/Rational Team Concert to manage EGL development projects.

# Is EGL Really an Open Standard?

**EGL is Open Sourced. IBM released the EGL Community Edition in the Summer of 2009. EGL went fully open source at the 2010 Innovation Conference..**

- **Fully open and extensible**
  - Utilize existing Java or JavaScript libraries if needed
- **Rich UI based on Web Standards**
  - REST, SOAP, JSON, OpenAjax, Dojo, etc
- **UI Libraries at the EGL Café**
  - Download third-party libraries
  - Write your own and upload them
  - Import into the visual editor palette
- **Plans for open implementation**
  - Allow third parties to extend EGL, develop their own version



# How does EGL Simplify Web 2.0 Development?

## There is no better proof than a side-by-side code comparison

### Here is the EGL Rich UI

```
handler MyRuiHandler type RUIhandler { initialUI = [ addressForm, map ] };

 addressField TextField { text = "1600 Pennsylvania Ave, Washington
width = 250 };

 goButton Button { text = "Go!", onClick ::= goButton_clicked };
 addressForm Box { children = [ addressField, goButton ] };

 map GoogleMap { width = "500px", height = "300px" };

  function goButton_clicked (e Event in)
    addresses String[] = [ addressField.text ];
    map.showAddresses(addresses, addresses);
  end
end
```

All code, including UI and controller logic, is written completely in EGL.

The complexity of the Google Map APIs are hidden from the developer, so the developer can focus on the actual business requirement and not technical complexities.


### the Dojo HTML and JavaScript

```
tp://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-

 equiv="content-type" content="text/html; charset=UTF-8"/>
oogle Maps API Example: Simple Geocoding</title>
<script src="http://maps.google.com/maps?file=api&amp;v=2.x
<script type="text/javascript">

var map = null;
var geocoder = null;

function initialize() {
 (GBrowserIsCompatible()) {
ap = new GMap2(document.getElementById("map_canvas"));
ap.setCenter(new GLatLng(37.4419, -122.1419), 13);
eocoder = new GClientGeocoder();


unction showAddress(address) {
  if (geocoder) {
    geocoder.getLatLng(
      address,
      function(point) {
        map.setCenter(point, 13);
        var marker = new GMarker(point);
        map.addOverlay(marker);
        marker.openInfoWindowHtml(address);
      }
    );


</script>
</head>

<body onload="initialize()" onunload="GUnload()">
  <form action="#" onsubmit="showAddress(this.address.value); return
false">
    <p>
      <input type="text" size="60" name="address" value="1600 Pennsylvania
Ave, Washington DC" />
      <input type="submit" value="Go!" />
    </p>
    <div id="map_canvas" style="width: 500px; height: 300px"></div>
  </form>
</body>
</html>
```

Developing RIAs by hand requires developers to become experts in multiple technologies – HTML and JavaScript. Neither was designed for the kinds of applications being developed today!

# Dispelling the EGL Myths. What is the Reality?

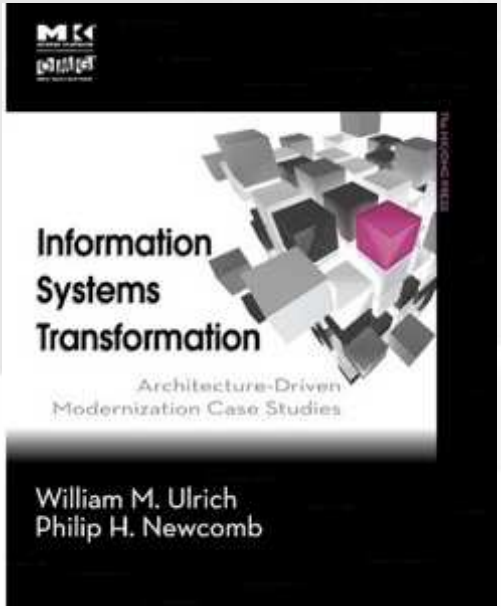
| EGL Cons | EGL Reality |
|---|---|
| • Current developer support is lacking, and the community is small in comparison with other languages. | • The community is probably bigger than people think, but is still relatively small compared to languages like PHP and Java. EGL has been freely available for only a few months - PHP and Java have been available to everyone for over a decade. It's will take some time to catch up.. EGL is easy to learn by existing developers and IBM has trained thousands of EGL programmers via its free distance learning. |
| • There is a perception, right or wrong that this language is not mature and is still new to market. | • Its been in the market in one form or another for the last 25+ years. It has grown and matured as the marketplace has changed...to encompass solving very hard software problems like multi-tier application development, component-oriented development and Web 2.0 development, but EGL has always handled the basic computing scenarios. EGL is running businesses all over the world. EGL is one of most mature enterprise-languages available today. |
| • IBM is sending mixed modernization messages. The IBM Rational Group is pushing Java … then EGL pops up on the radar as a Java replacement. Why not C# or .NET? | • This is an oranges and apples comparison. EGL does not compete with Java or C# or any other executable language because EGL is a pure specification language that uses generators to create code that is executable. Developers creates EGL specifications that are platform neutral and EGL generators create platform specific code that runs on multiple enterprise platforms (hence the name Enterprise Generation Language). |
| • The language lacks many of concepts surrounding Object Oriented Programming, e.g. It lacks inheritance. | • This is by design to simplify the construction of business applications around relational databases. EGL is a business-oriented language, not a general purpose language. Because of this, it's not necessary to expose this layer of complexity to developers. (EGL actually does support OO concepts, but EGL is not a pure OO language). |
| • Extending RUI widgets is not intuitive and the walkthroughs on EGL Café result in marginal understanding. | • IBM's philosophy is that general business developers should not be writing widgets. Widget development (whether the widget is written in EGL or JavaScript) should be left to a select group of developers with these specialized skills. IBM plans to improve its tooling for creating custom widgets. |
| • The tutorials provided on EGL Café cover many of the general activities of EGL but come up short when trying to teach extensions. | • IBM is planning additional tutorials for EGL Café in 2010. |
| • There is a lack of 3rd party tools or information regarding performance and runtime debugging. | • IBM's EGL Open strategy encourages 3rd party tool providers as demonstrated by the fact that they work closely with TSRI and other partners such as FBD Systems, Clearblade, etc. The EGL Eclipse plugin supports runtime debugging |
| • Limited code-completion support, underlying DOM/JS/Html elements are not easily referenced without prior understanding/knowledge | • The goal is to shield developers from DOM/JS/HTML. If there is something that needs to be exposed more natively in EGL IBM development teams are available to make these extensions. |
| • General EGL proficiency tends to require additional knowledge sets including JavaScript, html, java, dojo/Ajax principles. Extending or creating RUI widgets requires more advanced skills and knowledge than the average EGL application developer will be expected to have. | • For general application development, none of these skills are required. For advanced widget development in JavaScript, the developer does need JavaScript/HTML skills, but IBM expects EGL customers to centralize this work in one or two people or rely on other companies for specialized widgets. |

# EGL Strategic Positioning

**EGL will never replace Java, C# or C++. But, as IBM's Open EGL Initiative picks up momentum EGL stands to win acceptance as a leading open source DSL**

- **EGL is Platform Independent Domain-Specific Language (DSL) that works in perfect symbiosis with other languages such as Java, JavaScript and COBOL. To say that EGL is in conflict with Java, C#, C++ and COBOL is like saying C is in conflict with assembler. IBM advocates EGL as complement, not a replacement for the many platform specific languages that EGL leverages.**
- **As a platform Independent DSL, EGL's purpose is to simplify and unify the language that is used for expressing complex programming concepts that target multiple runtime environments. IBM promotes EGL as one simple highly expressive language that increases programmer productivity by seamlessly leveraging the infrastructure of native platforms and platform specific implementation languages and runtime environments.**
- **EGL's ability to leverage the runtime languages of the various platforms it supports makes creating complex business applications much more productive. It does this through seamless conversion of language and metadata into the runtimes of multiple target languages and platforms, thereby achieving versatility, robustness, interoperability and platform independence for software applications expressed in EGL.**
- **EGLs emergence as an open source DSL is not an IBM plot to supersede Java, it is the natural evolution of EGL's ability to map complex abstract concepts captured in language and metadata into various runtimes to facilitate interoperability between IBM proprietary and open source languages and platforms. There is, therefore, no conflict whatsoever between IBM advocacy of EGL as a target for modernization and IBM's Websphere and Java strategies. EGL is fully complementary to JEE and Java where it serves as a tool for open source software development in much the same way as UML does for modeling software designs and architecture.**
- **The purpose of IBM's open EGL initiative is to give the open source community access to the same infrastructure IBM has developed for its proprietary languages and platforms. EGL, by supporting seamless interoperability between IBM's proprietary and open community platforms is invaluable to IBM enterprise clients whose IT is built upon a blend of open source and proprietary technical infrastructures.**
- **IBM does advocate the use of EGL as an alternative when EGL can simplify of the complexities of the other languages, achieve more compact expression of complicated problems, improved interoperability, security, distribution, and increase program productivity, and EGL can be introduced cost-effectively.**
- **TSRI's unique ability to automatically reengineer and redesign applications as they are transformed and refactored into EGL at the lowest cost point and highest quality in industry is a powerful accelerant and incentive for taking TSRI clients towards a platform-neutral open-source DSL that is fully supportive and complementary of TSRI's other major open source and open architecture initiatives specifically: Model Driven Architecture (MDA) , Architecture Driven Modernization (ADM), Unified Modeling Language (UML).**

# Book Information




Information Systems Transformation: Architecture-Driven Modernization Case Studies

By William M. Ulrich and Philip H. Newcomb

Published by Morgan Kaufmann

- Acts as a one-stop shopping reference and complete guide for implementing various modernization models including core concepts, common scenarios, and a guide for getting started.
- Concepts are illustrated with real-life examples from various modernization projects, allowing you to immediately apply tested solutions and see results.
- Ten chapters containing in-depth modernization case studies, covering multiple platforms, industries and government agencies from four different countries.

### Grady Booch - Book Review

**Ulrich and Newcomb's book offers a comprehensive examination of the challenges of growing software-intensive systems. Every system has an architecture, but as the authors explain, it is only by continuous, intentional architectural transformation that one can attend to costs while simultaneously creating a mechanism wherein innovation may flourish.**

**I enjoyed the many case studies. Every circumstance is unique, but the authors have offered up best practices for systems modernization from their experience. Their focus on architecture as an artifact for governance is sound, but they go far beyond the technical issues and address many of the social and economic practices that help one evolve a good technical architecture.**

**From the moment one writes a line of code, it becomes legacy, and that legacy accumulates. Whether it becomes a cause of innovation inertia or a source of future value is a factor of how it is continuously modernized.**