Eclipse Members' Meeting

September 22, 2005
Chicago

Agenda

- Introduction and Welcome
- Executive Director Report – Mike Milinkovich
- New Strategic Developers
    - Iona – Carl Trieloff
    - Nokia –Heikki Koivu
- Project Update – Bjorn Freeman-Benson
- Break
- Member Spotlight – Macromedia – Phil Costa
- Project Spotlight – Web Tools Project – Tim Wagner
- Add-in Providers Requirements Session
- Lunch noon-1:00pm
- Add-in Provider Requirement Break-Out Sessions
- Add-in Provider Themes and Priorities Voting
- Break –
- Harvard Business School Research Project: 'Competing on a common platform' - Siobhan O'Mahoney
- Marketing Update – Ian Skerrett, Eclipse Marketing Director
- EclipseCon 2006 - Bjorn Freeman Benson, Eclipse Organizing Chair

Executive Director Report

Mike Milinkovich

Welcome to New Members

- BuildForge, Inc
- Cognos
- IDG Japan (Associate)
- I-Logix.
- IONA (Strategic)
- Ivis Technologies
- Jiva Medical
- Nokia (Strategic)
- Palamida
- Spike Source

Eclipse by the Numbers

- 109 Members
  - 15 Strategic Members
  - 79 Add-In Provider Members;
  - 15 Associate Members

- 8 Top level projects (plus 1 proposed)
  - 50 sub-projects (plus 4 proposed)

- 470+ committers; 30+ organizations

New Eclipse Foundation Staff

New Staff

- Wayne Beaton, Eclipse Evangelist

- Matt Ward, Web and IT Guy

- Ralph Mueller, European Eco-system

Current Staff

- Mike Milinkovich - Executive Director

- Skip McGaughey - Director, Eclipse Ecosystem

- Sharon Wolfe - Office Manager

- Ian Skerrett - Director, Marketing

- Denis Roy - Manager, IT (webmaster)

- Bjorn Freeman-Benson - Technical Director

- Janet Campbell - Manager, Intellectual Property

Eclipse Financials

- Budget for 2005 (excluding EclipseCon)
  - Projected Revenue: $3.1M
  - Project Expenses: $2.7M

What Have We Been Doing in Q3?

- Quality at Eclipse   (the Eclipse Way)

- IP Due Diligence Process

- Planning for EclipseCon 2006

- Project Phoenix

- Document processes to enable growth
    - Membership process
    - Committer process
    - Press release guidelines

- RCP Strategy

Looking Forward for Eclipse

Vision Statement

In 2008, Eclipse will be viewed world-wide as the leading development platform for building, integrating and deploying software applications on hardware ranging from embedded devices to desktops and servers. Rich Client Platform (RCP) will be the strategic platform for developing, deploying and managing software applications and products.

Eclipse's key differentiators are: vibrant and growing community, superior technology, free and open source, open process and governance model, and no vendor lock-in.

Strategic Goals

- To support the evolution of the Eclipse Technology as the technically pre-eminent open source development platform.
- To grow the Eclipse Project with innovative projects and technology, while maintaining quality. Create an environment where Committers and contributors value their participation in and association with the Eclipse Project.
- To promote the needs of the Eclipse ecosystem. Create an environment where the commercial members of the Eclipse ecosystem can profitably leverage the Eclipse Technology.
- To foster the widespread adoption of the Eclipse Technology.
- Have a large and diverse Membership.
- Be an industry leader.
- To ensure the long-term financial viability of the Foundation.

Enabling a Worldwide Ecosystem

- Eclipse Around the World
  - Leverage local members to promote Eclipse in different geographies
    - Ex. Eclipse Japan & Eclipse China working groups
  - Establish an Eclipse Foundation presence in Europe
  - Expand into South America

- Protect the integrity of the intellectual property

- Promote a commercial successful Eclipse ecosystem
  - Establish marketing programs to support commercial Eclipse products
  - Recruit and promote education, support and service providers

We Need Your Help to Make This Happen

Quality and Innovation

- Eclipse = Quality + Innovation

- The Eclipse Way documents the Eclipse development process

- Continue to recruit new technology and incubator projects

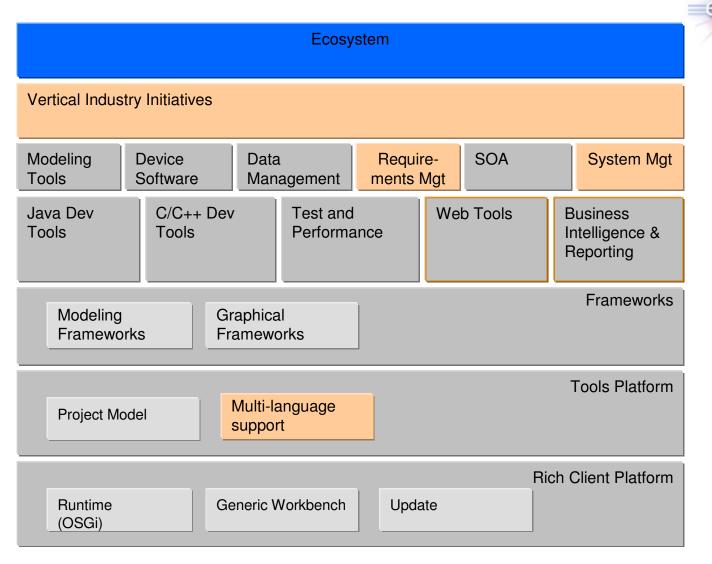- Need to deliver on the existing projects

To Build and Define an Open Development Platform

- Lifecycle coverage and tools infrastructure

- Rich Client Platform

Eclipse Architecture

Ecosystem

Vertical Industry Initiatives

Modeling Tools	Device Software	Data Management	Require-ments Mgt	SOA	System Mgt

Java Dev Tools	C/C++ Dev Tools	Test and Performance	Web Tools	Business Intelligence & Reporting

Frameworks

Modeling Frameworks	Graphical Frameworks

Tools Platform

Project Model	Multi-language support

Rich Client Platform

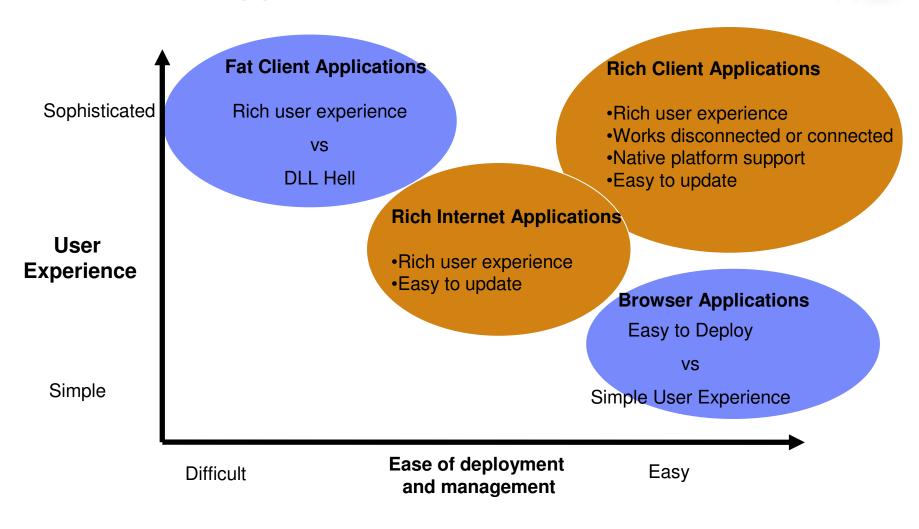| Runtime (OSGi)	Generic Workbench	Update

Potential New Projects

Lifecycle coverage and tools infrastructure

- Deliver on current projects

- Recruit new projects to complete the lifecycle

- Continue to expand into embedded and mobile development

- Add support for more languages; ex: scripting languages
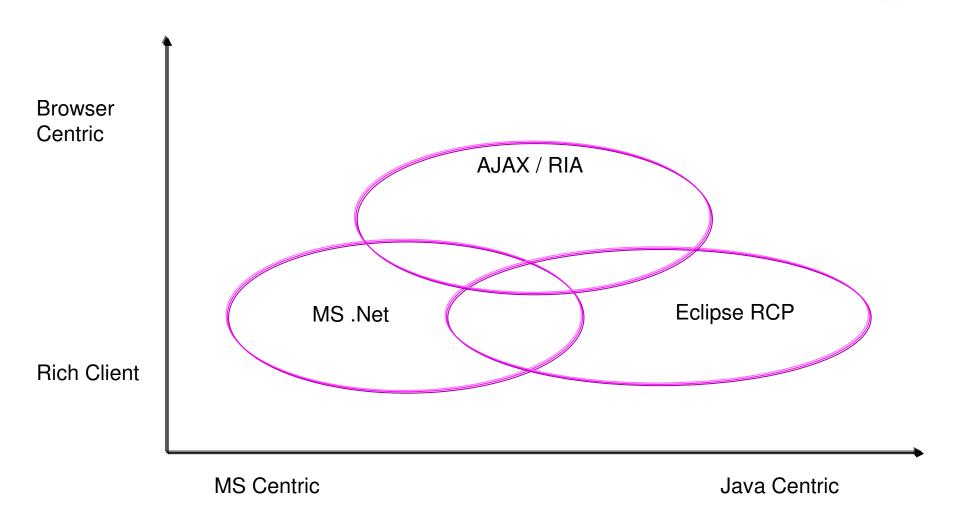
Rich Client Applications

Fat Client Applications

Rich user experience

vs

DLL Hell

Rich Client Applications

- Rich user experience
- Works disconnected or connected
- Native platform support
- Easy to update

Rich Internet Applications

- Rich user experience
- Easy to update

Browser Applications

Easy to Deploy

vs

Simple User Experience

Sophisticated

Simple

User Experience

Difficult

Ease of deployment and management

Easy

Opportunity Knocks

- MS Vista will cause ISVs to retool

- Linux is pushing into the desktop

- Web services / Web 2.0 is causing ISVs and Enterprises to rethink their client strategy
    - Simple browser interface vs rich client vs RIA

Technology Alternatives

Browser
Centric

AJAX / RIA

MS .Net

Eclipse RCP

Rich Client

MS Centric

Java Centric

Eclipse RCP

- Robust component model
  - Based on OSGi standard
  - Component model based on *plug-ins* that are versioned and can be shared across applications

- Intelligent Install and Update
  - Easy initial install
  - Dynamic updates from central server with no requirement for application restart

- Component libraries to accelerate building applications
  - Core set including a generic workbench, error handling, preference setting, menu management, context sensitive help, etc.
  - Easily extend with other frameworks: BIRT, Eclipse Communications Framework, Eclipse Trust Framework

- Portability
  - Applications built on RCP can run on Windows, Linux, Mac OSX, etc.
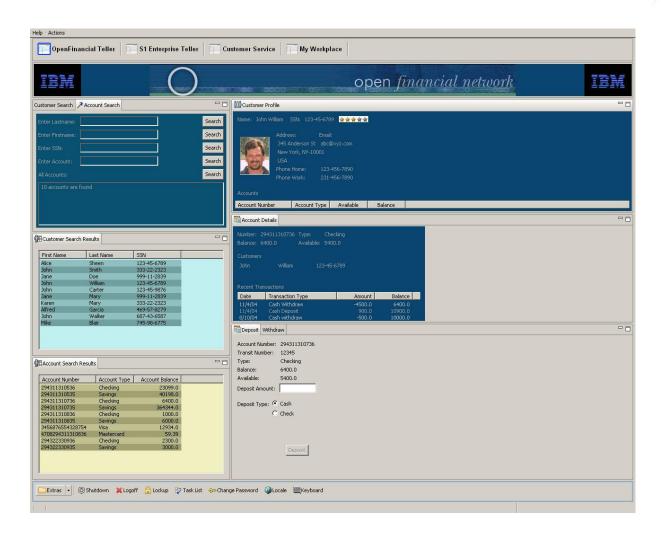 - Future support for constrained devices such as PDAs, smart phones

Eclipse RCP

- **Native Platform Support**
  - Applications take full advantage of native platform resources, including native UI widgets

- **Work Offline**
  - Application work when not connected to the network

- **Robust set of developer tools**
  - Extensive Eclipse eco-system of developer tools
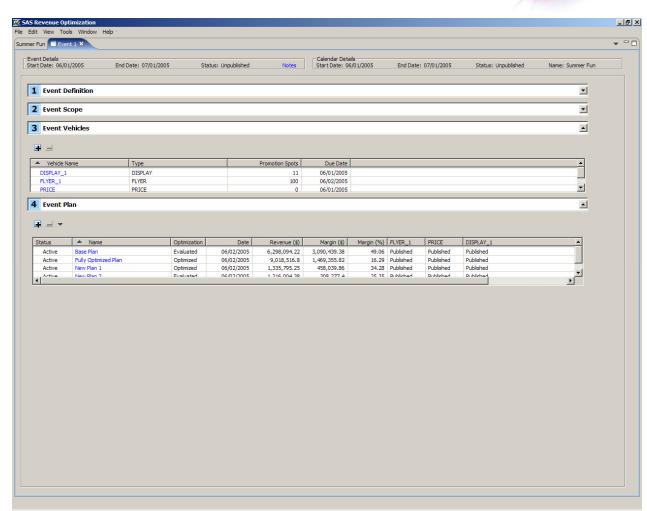
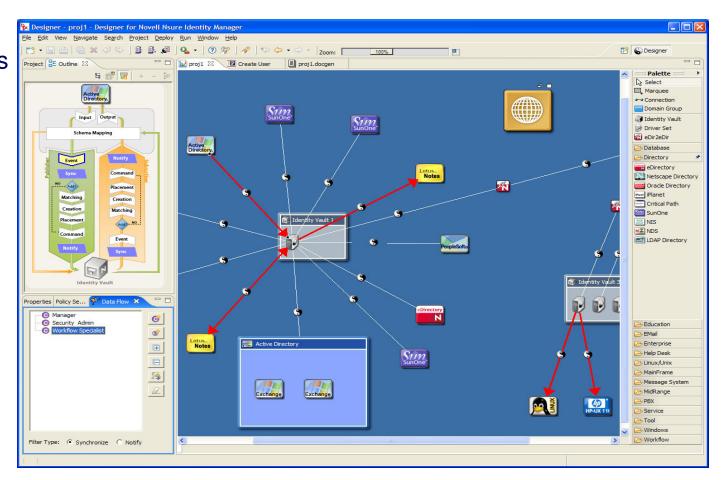# IBM Workplace Client Technology

SAS – Promotional Planning

- From SAS Revenue Optimization Suite

Novell Identity Manager Designer

- Graphical modeling tools for identity management

Why Eclipse RCP Rules

1. Web is not for everything

2. .Net Smart Client Replacement

3. Tools and Support

4. It's Java

5. Above all, its Free.
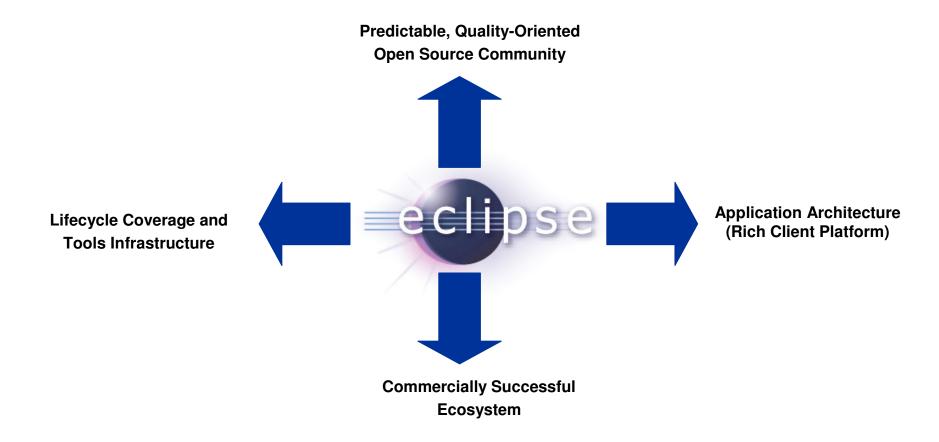
BLOG: http://www.lakesidelab.com/henry/105.asp

RCP Strategy

- Technology is here now!
  - Well, there's still some work to be done….

- Next 12 months focus on gaining momentum
  - Recruitment of ISVs
  - Developer Evangelism

- What can you do?
  - Tools for RCP – ex. RAD, testing tools
  - Adopt RCP for as your application platform
  - Education and services

The Future is Bright

Predictable, Quality-Oriented
Open Source Community

Lifecycle Coverage and
Tools Infrastructure

Application Architecture
(Rich Client Platform)

Commercially Successful
Ecosystem

Welcome to Iona and Nokia