Borland and Eclipse

Strategic Developer Membership
Agenda

• Historic Context
• Current Borland products on Eclipse
• Future of Borland products
• Strategic Developer Commitment
• Graphical Modeling Framework project
Historic Context

- Borland & TogetherSoft formed part of the initial Board of Stewards in November 2001
- TogetherSoft’s Together® product was rewritten on Eclipse and released in late 2002
  - LiveSource™ UML™ Modeling for Java
  - Now includes CaliberRM integration
- StarTeam client written for Eclipse
  - Arguably, best in the industry
- Other products in development
Current Products on Eclipse

Together
- LiveSource UML
- Design Patterns
- Doc Generation
- Audits & Metrics

(Next generation in development)

StarTeam
- Version Control
- Change Requests
- Task Management

CaliberRM
- Requirements
- Traceability
- Doc Generation

Borland
Strategic Developer

• Motivation for Borland
  – Continued Support of
    • Customers
    • Industry
    • Standards
    • Open source
  – Borland SDO vision is more rapidly realized by leveraging Eclipse
  – Contributing to the Eclipse community is essential to the success of vision!
Contribution

• Borland brings expertise in:
  – Modeling Tools (Together products)
  – Java/J2EE Development Tools (JBuilder)
  – C++ Tooling (Borland C++ Builder)
  – Performance Tooling (OptimizeIt)
  – Requirements Management (CaliberRM)
  – Configuration Management (StarTeam)
  – Source Analysis Tooling (Together audits & metrics)

• Many areas of Eclipse represent potential for Borland contribution
  – EMF, CDT, WTP, JDT, TPTP, BIRT, GEF, etc.
Contribution

- Borland will commit 8+ full-time developers and additional staff on a part-time basis
  - *Names, experience, and contribution breakdown to be provided by 03/15/2005*

- Borland has proposed to lead a new Graphical Modeling Framework project...
Graphical Modeling Framework (GMF)

An Eclipse Technology Project Proposal
What is GMF?

• The Graphical Modeling Framework (GMF) Project provides the underlying components and framework for the generation of design surfaces within Eclipse from domain and diagram models.

  – Intended to bridge GEF & EMF, hence GMF
  – Goal: to deliver exemplary diagramming for Eclipse UML2 Project, while providing general diagram support for any domain model
GMF Overview

• Main Components
  – Diagramming Infrastructure
    • Eclipse framework for design surfaces
      – Editor, view, properties, navigator
    • Generic frameworks for constraint, query, validation, etc.
    • Diagram definition designer (visual option, of course)
  – Diagram Generator
    • Diagram metamodel, generator model, mapping view
    • Generation framework for diagram elements
      – Node, edge, connector, constraints, etc.
    • Constraint implementation generator
  – Exemplary Tools
    • Diagramming for **Eclipse UML2 Project**
    • ECore modeling surface
GMF Overview

EMF's ECore

Diagram Metamodel

Domain Model

Diagram Designer

Generation

Diagram Definition

Diagram GenModel

Instance of Diagram Metamodel, dependency on Domain Model (DSL)

GEF

GMF Runtime Plug-in(s)

Domain Model EMF.edit

EMF serialization

Diagram Plug-in(s)

Diagram Instance

Optional XSL Transformation

Diagram SVG
Questions?