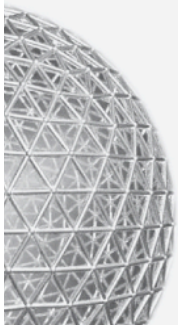

Buckminster

Helios Review

May, 2010

AGENDA

- Buckminster Objective
- Buckminster Capabilities
- Common Components used
- API Quality
- IP Clearance and License
- Committer Diversity
- Suggestions for Next Steps

BUCKMINSTER OBJECTIVE

Buckminster's objective is to leverage and extend the Eclipse platform to make mixed-component development as efficient as plug-in development. The basic approach is to introduce a project-agnostic way of describing a development project's component structure and dependencies, and to provide a mechanism for materializing source and binary artifacts for a project of any degree of complexity.

BUCKMINSTER CAPABILITIES

- **Complex dependency resolution**, providing recursive resolution of dependencies leveraging existing Eclipse "Team Providers," with the addition of new retrievers, for exemplary purposes, covering source and binary artifacts that are not version-controlled in a traditional sense as well as P2 repositories. Resolution uses the flexible Omni Version from P2 to describe versions and ranges. This allows comparison of current and prior dependency resolutions to support update impact analyses.
- **Uniform component dependency format**, using a component-type agnostic mechanism for describing components and their respective targets and dependency requirements. Most Eclipse projects, and many other component types, have some level of dependency information that can be leveraged. Extensions can be added to provide additional strategies for dependency pattern recognition.
- **Intelligent retrieval mechanisms**, using a component query mechanism the resolution and generated bill of material needed for a given configuration are separated from the actual materialization. This allows sharing of configurations with varying degree of variability between team members (e.g. from “all source needed for latest revision on main branch” to “frozen release configuration”).
- **Flexible project workspace binding**, allowing components materialized on disc to be bound to a workspace in different ways, including invoking “build action” before binding to a workspace and supporting "Proxy Projects" consisting of links to physical artifacts and auto-generated Eclipse project information. These capabilities are helpful when sharing code or other artifacts that are not eclipse projects.

BUCKMINSTER CAPABILITIES

- **Actions**, leveraging existing “build technologies” both within Eclipse (PDE-build) and external (ANT), Buckminster can drive building, and assembling components. Materialization is not just “copying of files” – a compiled artifact can be materialized from its source.
- **Headless mode**, a headless packaging of Buckminster gives the same capabilities to command line level tools and scripts as what is available in the Eclipse IDE. As an example Buckminster can drive headless PDE builds.

COMMON COMPONENTS USED

- Platform Resources
- Team CVS
- Team SVN
- Egit/JGit
- PDE build
- P2
- ECF
- EMF

Buckminster

Thirdparty bundles that we can make use of

- Subclipse from Tigris.org
- P4WSAD from Perforce.com
- SVN client libraries from Polarion
- ANT
- Hudson
- EcLemma
- Groovy

API QUALITY

- In active use
  - Buckminster is used by a number of companies such as Progress, Oracle, Chello, Cape Clear, Microsimage, eXXcellent solutions, Versant
  - The Buckminster project uses Buckminster of course, to build and publish Buckminster
  - Used by Eclipse EMF (several projects), TMF, STP, ECF, EPP.
- Community participation
  - Users and people who try things out can communicate on the buckminster-dev mailing-list or the buckminster newsgroup. The mailing-list is also mirrored as a newsgroup.
  - Users can communicate on the #eclipse-b3 ‘build channel’ on IRC
  - We participate in a number of other electronic forums such as pde-build, equinox-dev, cvs-dev, ecf-dev, egit-dev, and subversive newsgroup where we collaborate with the Eclipse community as a whole.
  - We attend build related workshops and all major conferences.
- Documentation
  - Documentation for Buckminster can be found at http://wiki.eclipse.org/index.php/Buckminster_Project

IP CLEARANCE AND LICENSE

- IPZilla Approvals
  - All code for Buckminster resides in the SVN repository at Eclipse.org, it is licensed under EPL, and has been IP approved.
  - See Auto generated IP Log at: http://www.eclipse.org/projects/ip_log.php?projectid=tools.buckminster

SCHEDULE

- Development adhered to Helios schedule
  - All milestone deliveries were met

COMITTER DIVERSITY

- Currently 9 committers
  - From 3 companies + individual
- Committers from multiple countries
  - US
  - Sweden
  - Czech Republic
  - Germany
  - France
- Working with / supporting projects
  - EMF (several)
  - Equinox P2
  - PDE Build
  - “Helios”
  - STP
  - ECF
  - EPP

SUGGESTIONS FOR NEXT STEPS

- Git and Buckminster
  - Increase quality in the integration with git as new features and fixes become available from Egit/Jgit.
- P2 and Buckminster
  - Continued effort to improve build processes involving p2
- PDE Build, Buckminster => b3
  - Continued effort to merge build technology into a unified build system under the EMFT.b3 project.
- Common Build Infrastructure and Buckminster
  - Continued improvements to Hudson/Buckminster integration
  - Make automated source–to–update site builds using Buckminster even easier

VISIT US AT

http://www.eclipse.org/buckminster

Buckminster