

Agenda

- Day 2 (May 8, 2009)
- * Participants: Ilkay Altintas*, Chad Berkley, Shawn Bowers, Christopher Brooks, Daniel Crawl*, Matt Jones, Timothy McPhillips, Ben Leinfelder, Bertram Ludaescher, Mark Schildhauer, Aaron Schultz, David Welker, Sean Riddle, Jianwu Wang*
- * [9-9:45]: Status of current Kepler/CORE projects
 - o Brief updates on Kepler/CORE development activities
- * [9:45-10:30]: Overview of target Kepler/CORE year 3 deliverables (MATT)
- * [10:30-10:45]: Break
- * [10:45-5 (rest of day)]: Discuss each target deliverable (MATT)
 - o Requirements, features, scope of deliverable
 - o List of tasks and action items for deliverable
- * [12-1:30]: Lunch
- * [3:30-3:45]: Break

Development priorities

- High
 - Packaging of modules and actors, dependency management, repositories, licensing
 - Configuration systems
 - Data access and binding components
- Medium
 - Documentation of architecture
 - Refactor into core and modules
 - Distributed and remote execution framework, kepler job management
- Lower
 - Test system and tests
 - Effective debugging and error reporting
 - Separate GUI from execution engine fully
 - File/Run/Project management UI and framework
 - Authentication/Authorization framework
 - Support provenance archives for repeatability
 - Ability to execute actors/wfs in multiple execution environments
 - SDK (module?) for developing actors against release installs

Development priorities

- Packaging of modules and actors, dependency management, repositories, licensing
 - Packaging of actors and modules -- harmonize KAR and module formats
 - version management of {actors, workflows, modules, ...}
 - dependency management {between modules, jars, external systems, native libs}
 - repository (API + implementation)
 - licensing inclusion in packages

Development priorities

- Configuration systems
 - Consolidate multiple configuration formats (down to 1 or 2?)
 - Add ability for modules to inject configuration

Development priorities

- Data access and binding components
 - Common actor for DataSources (change Adapter to Delegate model)
 - Common API for accessing remote data services
 - Data management facilities in Kepler
 - ...

LT

- Kepler Leadership Deliverables:
- System development guidelines, policies, culture (43)
- "User"-level documentation (40)
 - -- Kepler/CORE develop infrastructure for modularizing & building documentation
- Release roadmap for core and standard modules (20)
- Sustainability

Others

- Documentation of architecture
 - more than UML diagrams, communication aid for developers
 - help new developers find information for where to hook into system, etc.
- Refactor into core and modules
 - create a coherent set of modules for kernel and standard extensions
 - refactoring
 - guidelines/metrics for when to define modules
 - modules inspired by architecture
- Distributed and remote execution framework, kepler wf job management
 - parameter sweep / parallel execution
 - support for running kepler wf engine and messaging