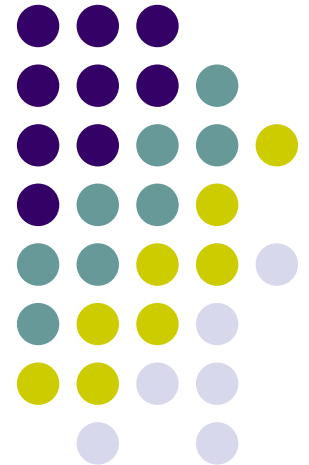
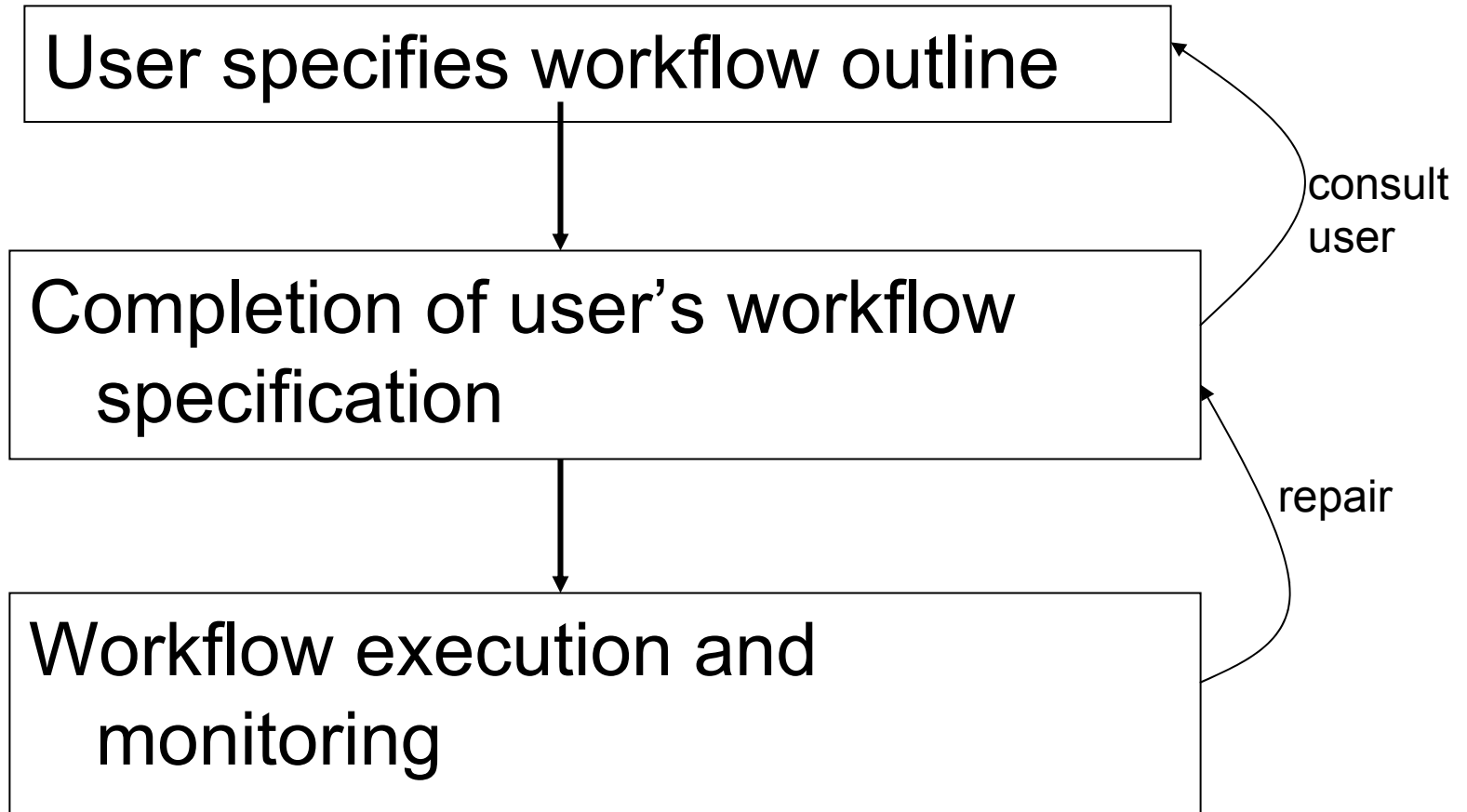


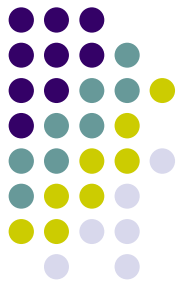
Planning services for workflow execution

Jim Blythe



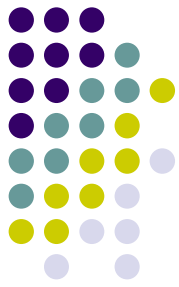
Phases in workflow management





1. User specifies workflow outline

- Use favourite GUI, possibly domain specific
- User wants complete control of key aspects of workflow
- Does not in general specify whole workflow in executable detail
 - 100's of tasks or more
 - Some resource details not interesting to user



Requirements

- Flexible language for user's constraints

“Use component S to achieve task T”

“Avoid using data from instrument I”

“Don't use more than 5 hours on resource M”

“Complete everything by 5pm”

- Should consider advice languages, e.g. [Myers 2000]
- Ideas from semantic grid group
- Some constraints can't be represented on single jobs, so must represent explicit workflows.

2. System completes the user's workflow



- Extends partial workflow to full concrete workflow satisfying constraints
- Uses GUI for feedback if constraints not satisfiable (or to give information)
- May require search (planning, scheduling, constraint satisfaction..)



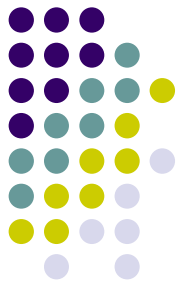
Issues/requirements

- Need to reason about the whole workflow to create an efficient executable version
 - Constrained tasks (via data or compute resources) may appear later in workflow
- User's constraints should be verifiable
- Need language for relaxing constraints
- Must interact with existing schedulers/job submission systems
- Variable levels of specification for immediate and future tasks

3. Execution and monitoring



- Environment may be highly dynamic during workflow execution
 - Monitor task progress during execution
 - Capability to repair workflow during execution: switch resources, jobs
- Constraints on workflows from earlier phases crucial to allow efficient, sensible repair



Requirements/issues

- Monitors that detect indications of problems or opportunities
 - Failing jobs, problems with resources that are needed for future jobs, opportunities to improve
- Requesting further user guidance when needed (variable autonomy)
- Predicting future problems?
- Conditional workflows

Workflow management services



- Distinguish ‘abstract’ workflows that can be instantiated many times from instances that can be executed
- Instances need persistent state. Specialized tools should be able to communicate with them,
 - e.g. GUI, matchmaker, monitors, refinement and repair tools,...
- Investigate appropriate service representations



Further issues

- Are existing workflow languages expressive enough for the constraints required? Or can they be extended?
- What QoS could a workflow service provide (based on those of the schedulers/matchmakers it uses)?
- Sharing workflows between different users or groups – public/private parts of workflows, resource resolution, collaboration.
- Many more, from other participants.