NRL Workflow & Worklets

Gaurav S. Kc
Programming Systems Lab

Department of Computer Science, Columbia University
Overview

- Introduction to the MLS workflow system from Naval Research Labs.
- Introduction to PSL’s Worklets system
- Wish list for the NRL system, and what the Worklets system can provide.
- Level of progress
- Future directions
- Conclusion
The NRL system

- Multi-level security constraints prohibit “write down” of messages, meaning that information never flows from a higher security level to a lower one, and control only flows in the opposite direction.
- The NRL workflow designer ‘splits’ a workflow spanning multiple security levels into individual independent workflows, each completely bounded by a single security domain.
- Special “sync” tasks handle the transition of data and the flow of control between domains.
Design and Run

- Decoupled workflow designer and runtime
- 3 kinds of simple tasks - human, transactional, non-transactional.
- Network task - abstraction for an aggregation of tasks in the designer; at runtime, the aggregate has its own controller that is delegated the responsibility for completion of the network task.
The runtime components
Inside the scheduler

- Transition
- doTask
- fireNextTask
- Transition
Worklets, WorkletJunctions

Gskc: java psl.worklets.WVM
usage: java psl.worklets.WVM <wvmName>
WVM created
Creating the sockets transporter layer for the WVM
   SocketListener: 128.59.23.10:9101
Class server listening on Web port: 9102
   serving classes on http://128.59.23.10:9102/
Creating the RMI transporter layer for the WVM
Creating RMI Registry: 128.59.23.10:9100
Could not create the RMI Registry
Setting RMI codebase to: http://128.59.23.10:9102/
   RMI Listener: rmi://128.59.23.10:9100/WVM
   ready to accept worklets
Survivability is the result of training readiness, state-of-the-art equipment, sound leadership, and informed decision making in the face of hazards that we have not created, don't fully control, and can't walk away from.


Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures, and recover full services in a timely manner.

- [http://www.cert.org/nav/index_purple.html](http://www.cert.org/nav/index_purple.html)
Survivability ???

- Our [evolving] interpretation:
  - A distributed software system is survivable when failure of individual components does not cause the whole system to fail.
  - A trivial solution would be to provide “enough” backup components so that there would be no individual task that cannot be executed.
  - A more comprehensive approach will find out which tasks to be undo/redo, how to restore lost data, which replacement task to run, etc.
Survivable workflow, but how?

- SurvivorNG: applying dynamic adaptation to the whole workflow by using Worklets to figure out alternative routes among surviving nodes.
- If the next task can be executed, then where can this be done? How do you know if the next task can be executed at all or not?
- At a finer level of granularity, when partially completed tasks fail, which is the way to go? If re-doing, can we still get the results of the predecessor task, is it still alive?
What we added ...
Inside PLScheduler

- transition
- fireNextTask
- doTask
- transition
a demonstration ...
Future Directions

- SurvivorNG is a very young project, need at least:
  - A programmatic way of specifying alternative task processors, and their capabilities through an editor
  - Mutual awareness among task processors at runtime - this will lead on to dynamic service discovery capabilities of the system
- More short-term plans are:
  - Modeling parallel routes by cloning/merging Worklets
  - Recoverability from mid-process execution, when the Worklet’s current site goes down
  - Survivable loader/manager as a ‘distributed’ process
Newer applications for Worklets keep coming out!!

Extension of the NRL system with minimally invasive modifications.

We enable a situation where we can entrust the survivability of a workflow to an “James Bond”-like entity, the autonomous mobile agent - is this the Survivor-007 system?