

Steve Newhouse comments on ESI

* Unique Identifier on submission

This has discussed several times in the past and has appeared and disappeared from the BES specification - currently out - as BES probably thought it could go into JSDL. The JSDL gurus at the F2F felt that this did not belong in the JSDL document itself (their argument - it has nothing to do with describing the job) but sat within the message body of the create activity (or equivalent) operation.

There was some discussion at the F2F about the failure semantics of the lower-level layers (beneath the webs ervice) that should be included here.

[Expansion of the WS-Addressing comment on ESI page 3 would be welcome.]

-- Decision (OK)

Optional with the characteristics that if it exists then it must be unique as specified in the ESI document. Temporal uniqueness? Over what time period does an implementation need to keep historical information about the activity to help determine uniqueness. The JSDL document JobName element should contain the unique ID. At most once semantics for the Unique ID.

=====

* State Model

The main difference (at first glance) is a distinct Hold state for each primary state - although BES records the proceeding state so the BES generic Hold is qualified with where it came from. Dave Snelling stated that the state model was still under discussion so no further comment will be made here. However, it appears to be clear that different state models may be associated with different activities - and in many cases may be simpler than either the BES or ESI document. Perhaps linking the state model to the JSDL application type (e.g. POSIX) may be a way forward? This would allow (say) a WebServiceApplication type in the JSDL document to have a very simple model (running/not running).

- HOLD means stop after you completed the current state then you move to a hold state.

- SUSPEND attempts to suspend the current state. The implementation may or may not support suspension. This should be advertised as a capability of the BES.

-- Decision (OK)

To add a HOLD state. Clarify the suspend state. And clarify the state transition table with a table. Change the CreateActivityFromJSDL to allow holds on any/all states. (Don't proceed past the specified states).

=====
* Activity Interface

Once the job has started in ESI it presents an interface that can be used to control the job. BES does not require or exclude the presence of such an interface, as activity specific control is done through the interface. Indeed, there is a probably a case to be made for 'any' interface to be described here - the job control interface described in the ESI document, and activity specific interfaces. There was some discussion in BES-WG of the POSIX JSDL job type being linked to a POSIX Activity Interface for control purposes. This generic capability would seem worth pursuing.

-- Decision (OK)

The Activity Interface is out of scope. A managed job interface needs to be done. Another group should handle this. The reason is we need want to push this to another group is because the BES is very generic and there can be several different types of activity interfaces. We encourage different activity interfaces to be developed based on the types of JSDL documents or activities. Separate document in the BES working group to define the Activity Interface. We need to make sure that we are consistent.

=====
* Information Model

Again, there has been much discussion with BES-WG and the OGSA-WG on the 'right' way to describe the BES container and the activities within it - picking up on CIM, GLUE and JSDL work. These segments of the ESI document should be compared to the work done in this area to date. If the inclusion of supported applications/libraries is important this should probably be modeled here.

=====
* Notification

I don't think there is any particular issue that it should be possible for a something to subscribe to state changes. Mandating WS-Notification mechanisms with the WS-RF rendering is obviously one route. As is WS-Eventing in other renderings. BES does not mandate one mechanism over another.

=====
* Notification on Submit

BES supports something approaching this by submitting into a suspended state at which point notification is set up. There is a gap here between BES and ESI functionality.

[Text Slide 1]