

DSDP – TM Conference Call, 29-August-2005

Meeting Notes

Attendees:

Wind River – Salzburg, Austria

Martin Oberhuber – WR Technical lead for DSDP-Target Manager

IBM

Dave Dykstal – Remote System Explorer (RSE)

QNX

Alain Magloire – QNX has own framework for Target connections, interested in DSDP

David Inglis -

Doug Schaefer – CDT Lead

Original Agenda

- Recent News
 - IBM RSE Status
 - Connectors by Intel
 - Start using eclipse.dsdp.tm newsgroup, no more crosspostings on eclipse.dsdp
- Discussions regarding TM Design
 - RSE, RSF, Connectors: differences, common things, what will we need?
- Next steps

Meeting Notes

Martin O. opens the call and asks attendees who had not joined the first TM call to introduce themselves (see attendee list above). There is no request for additional agenda items.

IBM RSE Status:

David D. – Open Source proposal will be discussed on the next exec level on Wednesday, everything moving on schedule so far; might be slowed down by IP and Legal in the end, but currently no critic obstacles in sight.

DSDP-TM Face-to-face meeting:

Martin O.: DSDP Face-to-face meeting scheduled for Oct 12-14 in Toronto, pending final OK for the rooms by Pete Nichols. Dave D and QNX folks are planning to be there.

Usage of Newsgroup and Mailinglist:

Martin O.: Want to use the dsdp.tm newsgroup for general announcements, questions and inquiries from the public; use the dsdp-tm-dev mailing list for internal detailed discussions, especially when there are attachments. Encourage to subscribe to the mailing list.

Other attendees agree that this way of using the newsgroup and mailing list worked well for them in other projects.

Connectors by Peter Lachner/Intel (TCA Proposal):

Dave D. and Martin O. see Peter's TCA Proposal on a very low level (covering the data transport for connections). Most probably, it will be possible to achieve the proposal's features by splitting

the RSE Subsystem into Transport and UI (see below). From the Connector Properties described in the TCA Proposal,

- The “Channel Class” would translate to the Subsystem Type (or abstract subsystem that the subsystem derives from or implements);
- The “Channel Protocol” doesn’t seem to be relevant, as long as the subsystem provides the services it has registered;

Other Properties like “Target Architecture”, “Share Channel”, “Multiple Instances” appear to be worth further discussion.

Design Discussions:

TM Extension Scenarios:

Assuming a working Target Management Framework, we see 3 Use Cases for 3rd parties to extend the system:

1. Additional Transport for an existing Service (Subsystem)
2. Add a Service (or Subsystem) – with its own new Transport, or via an existing Transport (though adding Properties for the connection if required)
3. Extension of an existing Service: e.g. for the Remote Files subsystem, one might want to add a view for showing file versions on remote file systems that support this; or implement other extensions specific to particular remote system types.

Some Services might become a framework by themselves.

Splitting RSE Subsystems into Transport and UI:

Currently, an RSE Subsystem consists internally of two parts:

1. The Functional Piece (API) for accessing the remote, and
2. An UI for representing (1).

By providing a separate extension point for replacing the transport of a subsystem with a new extension (that implements the functional API), we get a “service” or “transport” for a subsystem that appears to be exactly what Peter describes as the “connector” or “connection adaptor” in his proposal. Thus, a subsystem becomes a framework of its own:

- minor extensions (like changing the Transport) can be done via extension point.
- major modifications of a subsystem can be done by subclassing, and registering the extended subsystem.
- very large modifications (e.g. new functionality) can be registered by adding a new subsystem.

Still, backward compatibility with current RSE could be maintained, since subsystems are free to provide pluggable transport or not.

QNX: Shall the proposed Target Manager be tied to the Team Provider?

David D: RSE can share connection properties via Eclipse Team Support. Other RSE features are separate from Team, since requirements are different: RSE provides remote build, deploy, test. Team provides versioning and synchronization only.

Martin O: Target Management and Team Support might want to share connection and transport mechanisms, e.g. ssh (which is currently hidden in an “internal” package of team).

David D: Sharing a Transport like ssh is a problem of prerequisites. The best technical solution would be to have some ECF support in the Platform, and provide protocols like ssh as an ECF plug-in; but Eclipse wants to keep the Platform lightweight, so they may not want full ECF.

Perhaps a Core ECF for plugging protocols could go into the Platform some day.

Target Manager could bring its own “small ssh” implementation, but have it expandable by an ECF-provided “full ssh” implementation to be plugged in.

ServiceType definitions in plugin.xml:

Martin O: RSF defines a "ServiceType" extension point which RSE does not have, will we need it?

Dave D: Based on experience, they did not yet need a complete meta-description of services in the plugin.xml. So far, the RSE team defined extensible "Abstract Subsystems", e.g. the "Universal File Subsystem". By having other implementations of File Subsystems derive from the Universal File Subsystem, the Universal system implicitly defines the Subsystem Type, since all derivatives implement the interfaces defined by the Universal Filesystem.

How to do Debugger Integration?

Dave D: Already implemented a Debugger Launcher. RSE's Process Subsystem allows to select a remote process, which can be used to kick off the debugger (which has its own communication channel to the target). The Debugger's Source Locator is extended by RSE to allow showing remote files (by first copying them to local).

Will the TM Framework also support Launching?

Martin O: Planning a Framework for Launching, where the Launch is just a sequence of Launch Actions. Launch Actions can be contributed (e.g. connect-to-remote, download, reset, start-application). This Launching Framework can be part of TM, or Device Debugging. Some Launch Actions will be predefined, others can be contributed.

How does this fit into CDT?

Martin O: Remote stuff can be in its own package that extends CDT. Some extension points (e.g. Launch Configurations) already exist. Others might be needed in order to provide good integration and user experience, but still avoiding cross-dependencies: remote stuff should not depend on CDT, and CDT should not depend on remote stuff.

Next Steps:

Dave D will send out a stripped-down version of the RSE documentation, and the FTP file service example plugin, to foster better understanding of RSE architecture.

The next phone conference will be in 4 weeks time, on 26-Sep-2005 at 9am PST.

Next Meeting:

Monday, 26-Sep-2005 at 9am PST.