

DSDP – TM Conference Call, 27-June-2005

Meeting Notes

Attendees:

Wind River

Rudi Frauenschuh – Wind River management sponsor for DSDP
Michael Scharf – Eclipse Architectural Council (EAC) member for DSDP
Martin Oberhuber – WR Technical lead for DSDP-Target Manager
Uwe Stieber – Developer and Maintainer of the Target Manager in WR Workbench

IBM

Dave Dykstal – Remote System Explorer (RSE)

QNX

Alain Magloire – QNX has own framework for Target connections, interested in DSDP
David Inglis, Mikhail Khodjaiants

Montavista

Pierre-Alexandre Masse – wrote the Bugzilla 65471 RSF framework

Accelerated Technologies (aka Mentor Graphics)

Mark Bozeman – Manager for Nucleus Edge Debugger

TI

Imtaz Ali – Manages TI Toronto Team
Andy Waterson – Manages Code Composer Studio
Chris Recoskie – Code Composer Essentials and Studio products

Intel

Peter Lachner – Manages XScale component tools project; currently compiler tools, nothing Eclipse based yet, interest in what's going on with DSDP-TM

Others

Scott Lewis – Component, Project Lead for Eclipse Communication Framework (ECF)
Mark Mason – Metaware
Marco Lopez – Interested Listener

Original Agenda

- Any new participants, "who is who"
- Recent News
 - Use Cases Revisited (v1.1)
 - ECF Findings
- Online demo of current systems – max. 10 minutes each
 - TI Code Composer Studio (Chris R.)
 - Wind River Workbench (Martin O.)
 - IBM RSE (David D.)
- Initial ideas on TM Design (as time permits)
- Next steps
- Resources

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.

Use Cases Revisited (v1.1):

Martin O: Incorporated feedback from the last phone call into the Use Cases Document. The new version will be sent out in the next days. The most important changes were:

- Added Use Case “Remote Build” – Not to be implemented by TM framework, but external clients may use TM services to accomplish remote build. Since this use case might introduce special requirements it was added to the list for documentation.
- Added Use Case “Target Discovery” – Motivated by the Discovery mechanism in ECF, “target discovery” allows to automatically find targets connected to local serial ports or the local LAN as well as target registries present in the local LAN.

Eclipse Communication Framework (ECF):

Martin O: Though ECF seems to be geared towards online collaboration between multiple (human) peers, discussions and meetings between Wind River and ECF (Scott Lewis) showed that ECF can also handle point-to-point communications over standard protocols like FTP where the remote system does not need to run ECF software. ECF provides some basic abstractions that seem to be helpful for Target Management, and Scott has volunteered to write initial ECF containers for FTP and Telnet as a showcase. We want to go forward and use ECF as our primary communications provider. Eclipse.org advocates and tries to foster such collaboration between various Eclipse projects.

Scott L. reinforces his offer to help in DSDP, extend and improve ECF where necessary, and invites members of the call to contribute to ECF as well.

Online Demo – TI (Code Composer):

Chris R. first shows the Code Composer Setup Utility. It allows to define a system that consists of multiple boards (either real or simulated). Boards can be predefined or custom defined by selecting CPU types, JTAG routers and other devices. The result is an XML file describing the JTAG scan chain of a board, that can be stored back into the database of custom components, and serves as the input for the debugger (currently in binary form).

The Parallel Debug Manager (PDM) takes a board configuration file as input. It allows to connect/disconnect individual devices on the system, define groups for components of the system, load software onto individual components or whole groups in one step, do synchronous run control for multiple components, and invoke a debugger for individual components. OS Awareness modules allow access to Tasks and Processes on the target system.

Currently, TI’s system is written in C++ using MS Visual Studio. They are currently working on a more generalized database for target components that will use XML more thoroughly. TI wants to have a single tool from target definition up to live target

connections (currently multiple tools). They have not yet started on Eclipse; they are somewhat waiting on what the DSDP-TM project will provide for them.

Online Demo – Wind River (WR Workbench):

Martin O. Shows the WR Workbench Target Manager. Various connection types (expandable through plug-ins) allow to connect various systems via Hardware Debuggers or Debug Agents. Target Connections are stored in a network-enabled Registry that allows sharing them in a team. Connection Types can define programs that need to be launched in order to facilitate the connection.

While connection types define the static information required to connect a target, dynamic information is also shown in the Target Manager Tree once a connection is made. Dynamic information is mostly execution contexts (processes, tasks) running on the target. The TM allows to reset targets, download software modules, and attach the debugger to contexts. Attachment is stored in a Launch Configuration; other Launch Configurations are available for running processes or Kernel Tasks. The Launch Configurations have combo-like controls in order to select or modify target connections from the Launch even if the Target Manager View is not visible.

Q: Does WR Workbench only support homogeneous systems?

Martin O: No, the Hardware Debugger allows defining a board file which contains multiple heterogeneous CPUs an/or Target OS plug-ins. Wind River currently only supports working with CPUs on the scan chain (no other devices yet).

Online Demo – IBM (Remote Systems Explorer, RSE):

RSE maintains a set of connections to systems of different type (e.g. Linux, AIX, Windows, ...). The resources on these systems are presented in an eclipse tree view that gives access to remote file systems and remote shells. Other types of subsystem can be plugged in.

Filters can be defined to show only those files that are of interest. Working on the remote file system is possible just as on the local one. Predefined and user-defined actions can be invoked on remote files. Refresh of remote file system information must be done manually every so often if the structure of the remote file system is changing.

When a shell is opened on a remote system, the first shell (“primary”) is also used for user-defined and compile actions. Multiple shells can be open at the same time. Information about user-defined filters and actions are stored as “Profiles” in a special Eclipse Project (“RemoteSystemDefinitions”) that can be shared through normal Eclipse Team support (CM tool integration). This works very well for the current user base (shipping since 3+ years).

The stock RSE uses a proprietary transport (called “datastore” – built on TCP) that requires an agent program to run on the remote system. Small footprint of the agent program has been an important design goal. Some IBM customers have written subsystem implementations that use FTP as file transport. Since they could re-use most of the remote file system framework, supporting FTP did not require too much coding.

RSE is used by IBM's remote debugger (not shown during this session), which uses RSE to download only partial source files for local display during a remote debug session. Remote Build has also been implemented already for iSeries tools (take project, export to remote iSeries host, invoke proper build style).

RSE is currently going through internal revision at IBM for making it open source. This internal revision process is expected to take about 6 more weeks. If accepted, the RSE team (about 3 developers) will be available for maintaining the Open Source RSE in the context of DSDP-TM.

Other Presentations:

Mentor Graphics wanted to present something but dropped off the call, we will give another slot for presentations in the next call.

Next Steps:

Martin O. will send out updated Use Cases Document as well as initial design proposal, incorporating lessons learned from today's presentations. The next phone conference will be in 3 weeks time, on 18-Jul-2005 at 9am PST.

Q: Is the scope of DSDP-TM too wide? Will the design proposal limit the scope?

Martin O.: Our main goal is to design a framework that should be as open and flexible for extensions as possible, thus we do not want to limit the scope too much now. Actual implementation will be limited to rather basic transport services in the Open Source version. Companies shall be able to use the framework in the way they need to. The Design Proposal will contain

- "Basic Decisions" like separating between static and dynamic model
- "Eclipse Extensions" listing what parts of Eclipse will be extended by the framework;
- "Extension Points" listing the capabilities where actual services can be plugged in.

Resources:

Rudi F.: Received no information regarding resources so far. Rudi will take on other duties in the future and asks for notes on resources to be sent to Martin O. instead.

Dave D. might be able to contribute 3 persons when RSE is accepted Open Source.

TI is willing to help out with defining interfaces for their area of expertise (Board Definitions) and reinforces basic willingness to contribute but cannot say anything more concrete yet.

Next Meeting:

Monday, 18-Jul-2005 at 9am PST.

Martin O. will be on vacation before the next meeting.