Device Debugging 1.1 Release Review

Pawel Piech

DSDP Device Debugging – Introduction

- Major project milestones
  - DD proposed as part of DSDP – May 2005; Created – June 2, 2005
  - New Eclipse Debug Platform Model/API's – Eclipse 3.2
  - New Memory View with customizable rendering – Eclipse 3.2
  - Debugger Services Framework introduced – July 2006
  - EclipseCon presentation – March 2007
  - ESC presentation – April 2007
  - 0.9 Release on Europa train – June 2007
  - EclipseCon tutorial and presentation – March 2008
  - 1.0 Release on Ganymede train – June 2008
- Continuing to expand community
  - DSF commercial adoption by Wind River
  - DD memory rendering used in several CDT-based products
  - Ericsson building GDB implementation under DSF
  - ARM using IP-XACT editor for internal development.

DD 1.1 Features

- 1.1 Plan available at http://www.eclipse.org/projects/project-plan.php?projectid=dsdp.dd
- Debugger Services Framework (DSF)
  - A framework integrated using Flexible Hierarchy API from Debug Platform
  - Designed as an extensible services architecture for building commercial embedded debuggers.
- Traditional Memory View Rendering
  - An implementation of the standard hex-based memory view look-and-feel found in most embedded debuggers.
- GDB debugger DSF implementation
  - Exemplary DSF implementation
  - Alternative to CDT's current GDB integration
- IP-XACT Editor (preview only)
  - XML editor compliant with the SPIRIT consortium’s IP-XACT standard (version 1.4) for defining SoC properties.
- DSF and IP-XACT documentation

Non-code aspects

- New and Noteworthy contains full user feature overview
  - http://www.eclipse.org/dsdp/dd/development/relnotes/dd_news-1.1.html
- User documentation and tutorials
  - Full documentation for IP-XACT component
  - Documentation for other components planned in future release.
- ISV documentation and tutorials
  - http://dsdp.eclipse.org/help/latest/
  - Includes Javadoc, DSF white paper, and DSF tutorials
- Externalization
  - Appropriate strings are externalized, but no localization will be done
- Publications and Conference talks as part of DSDP
- DSF tutorial presented at EclipseCon 08

API: 1.1 Status

- Previous releases of DD contained stable and provisional API
- DSF contains stable and some provisional APIs
  - Changes to stable APIs were backward compatible and are marked with a “@since 1.1” tag.
- Memory Rendering contains only few stable APIs
  - No Memory Rendering APIs were modified in 1.1 release.
- IP-XACT editor is still not on 1.0 version and contains provisional APIs only.

Architectural Issues

- DSF is an extensible framework intended to be extended by commercial device software development tools vendors for their proprietary debug engines and target platforms. More integrations with different debugger back ends are needed to fully validate this framework.
- The IP-XACT editor (provided still as a preview only for the 1.0 release) is also exclusively a Tool component.
- A stress and performance test framework is needed for the DSF UI components.
- Overlaps with other projects
  - This project functionally overlaps with the CDT debugger. This is by design, as the DD project is attempting to build the next-generation debug framework for CDT. Many of the CDT debug participants are also DD project participants.

Tool Usability

- The GDB debugger is a fully functional debugger. Although, it still lacks a few minor features available with the current CDI-based GDB debugger.
- IP-XACT editor
  - Utilizes the IP-XACT schema (separately downloaded) for rules checking.
  - Provides the user will a large library of SoC components for building a chip design.
  - Provides a wizard for creating new components.
- The Traditional Memory Rendering is a fully-functional memory view rendering that can be included in any Eclipse debugger that utilizes the Eclipse Memory View and framework.

End-of-life

- Nothing is end of life right now.

Bugzilla

- Statistics as of 14-November-2008

		Status						
		NEW	ASSIGNED	REOPENED	RESOLVED	VERIFIED	CLOSED	Total
Severity	blocker	.	.	.	.	1	1	2
	critical	1	.	1	4	1	7	14
	major	6	.	.	18	7	14	45
	normal	136	7	3	135	80	227	588
	minor	23	1	.	7	17	10	58
	trivial	1	.	.	1	.	2	4
	enhancement	90	1	.	43	18	48	200
	Total	257	9	4	208	124	309	**911**

- Release Exit Criteria
  - 0 Critical Bugs for Current release.
  - Critical bugs allowed for future releases.

Standards

- IP-XACT 1.4
  - For details, see http://spiritconsortium.org
- ECSI (European Electronic Chips & Systems Design Initiative)
  - DSF was presented at the ECSI workshop on debug standard proliferation.
  - Discussions about potential overlap and synergy with other standards such as Sprint and TCF
- TCF (Target Communication Framework)
  - TCF is a proposed standard debug protocol in Power.org
  - DSDP/TM project is developing a TCF debugger integration using DSF

UI Usability

- Externalization and Accessibility guidelines followed
  - Keyboard accessibility of all items verified
  - Menu items for special keys
  - Messages marked up properly for screen readers
- All UI-visible Strings are externalized
- Externalization partially through Eclipse NLS mechanism
- No localization will be done.

Schedule

- Project plan posted 17-June-2008
- Release milestones
  - 1.1 M1 – July 17th, 2008
  - 1.1 M2 – August 29th, 2008 (completed on September 5th)
  - 1.1 M3 – October 10th, 2008 (completed on October 17th)
  - 1.1 M4 – November 7th, 2008
  - 1.1 RC1 – November 14th, 2008
  - 1.1 RC2 – November 21th, 2008
  - 1.1 – November 28th, 2008

Process

- Open, Transparent Planning and Execution:
  - Features and Technical Working Groups maintained on Bugzilla, with “Overview” index entries on the Wiki
  - Made all communications public on the Mailing List, Regular phone conferences open to the public. All notes on Wiki.
- Several open meetings to discuss requirements, use cases, and development issues.
- Infrastructure: Automated nightly builds

Committers and Contributors

- 8 committers from 3 organizations (WindRiver, ARM, Ericsson).
- Direct contributions from Nokia (1 engineer), Wind River (1 engineer)
- Mailing list participation from Nokia, Texas Instruments, IBM, ST Microelectronics
- Emphasis on completing the GDB debugger integration.
- Weekly DSF committer calls

Community

- Developer and Plug-in Provider Communities
  - Still the best supported community.
  - Emphasis with DD 1.0 is building a high quality framework for commercial adoption
  - EclipseCon '08 tutorial aimed at helping framework adopters.
- User Community
  - Fully functional GDB debugger integration now available. It is used by Ericsson, ST Mircoelectronics, and Nokia, and it is planned to be included in the Wascana (http://wascana.sourceforge.net/) distribution.
  - DSF is used in Wind River's IDE and is being considered by Nokia, Freescale, and other CDT members.
- Talks at EclipseCon, EclipseSummit Europe 2007
- Press activity as part of DSDP
  - See DSDP press coverage report

IP Issues

As per the Eclipse IP Policy, the project verifies that:

- ... the about files and use licenses are in place as per the Guidelines
- ... all contributions (code, documentation, images, etc) have been committed by individuals who are Members of the Foundation and are abiding by the Eclipse IP Policy (training through Committer HOWTO)
- ... all significant contributions have been reviewed by the Foundation's legal staff – even if written by committers prior to joining Eclipse
- ... third-party libraries, have been documented in the release and reviewed by the Foundation's legal staff
- ... all contribution questionnaires have been completed
- ... the "provider" field of each plug-in is set to "Eclipse.org"
- ... the "copyright" field of each plug-in is set to the copyright owner
- See the IP Log at http://www.eclipse.org/dsdp/dd/development/dd-log.csv

Future Plans

- Proposed service release 1.1.1 in March 09
- Move DSF, GDB, and Memory components to CDT project for CDT 6.0 release. Reasons for moving are:
  - Give the DD technologies access to a much bigger user community.
  - Make it easier for CDT committers to contribute to DD technologies.
  - Allow CDT to begin replacing some older technologies which have architectural problems.
- Move IP-XACT Editor component to a new DSDP Incubator project