

Eclipse for embedded targets



By Gene Sally

As readers of this column are well aware, Eclipse is a general-purpose application development platform incarnated as an IDE. While first introduced as a Java IDE in November 2001, the C Development Tools (CDT) team shortly followed that release with the first installment of the C/C++ development environment.

One segment of the market that quickly adopted Eclipse was the embedded tools market. Prior to the rapid adoption of Eclipse, all of the vendors in the embedded market maintained their own IDE distributions. However, the open nature of the Eclipse framework and surrounding plug-ins made it a natural choice for those who wanted to just concentrate on the value-added features for embedded developers, namely:

- Cross-compilation
- Remote debugging
- Target management
- Target configuration

Code development for an embedded target shares many of the same IDE features as any other coding project. From the perspective of the user, these are low-valued features. Having them work well is vitally important, but the user just expects the editor to contain all of the bells and whistles found in the hundreds of editors already in the market.

Likewise, the user of an IDE expects the project management functions just to work, and consequently does not view them as highly valuable features of the product. The large amounts of effort required to get these features working takes resources away from the goal of producing the specific requirements of embedded engineers. Furthermore, the engineering costs for making the things that should “just work” must be amortized over a smaller user base, driving up the costs and pricing of tool products.

Device coding projects

With the introduction of Eclipse and the CDT, companies in the embedded tools space found the solution to their problem of delivering a high quality IDE experience to their users while delivering the embedded developers specialized tools. Eclipse and CDT handled the C/C++ development toolset, granting the embedded tools vendor the bandwidth to focus on just the small slice of features required by their users.

As embedded tools vendors worked on their products, it became apparent that many were still solving problems common to all vendors in the market. For example, when TimeSys launched TimeStorm IDE three years ago, it contained plug-ins for cross-compiler support, remote execution/debugging, and target management. Other IDEs from companies in the market followed suit, each with their own plug-ins solving similar problems.

The open, plug-in-oriented nature of Eclipse made it possible for embedded vendors to work on creating plug-ins for these common requirements as well. To reach that goal, the Device Software Development Platform (DSDP) project was proposed and subsequently approved by the Eclipse board of directors June 8, 2005.

The DSDP project charter states: “The Eclipse Device Software Development Platform Project is an open source collaborative software development project dedicated to providing an extensible, standards-based platform to address a broad range of needs in the device software development space using the Eclipse platform.”

The project, currently led and staffed by engineers from Wind River Systems, is working toward creating solutions for target management and device debugging.

DSDP target management

Embedded engineering typically means developing code that will run on a target other than that of the development host. The target management project will focus on integrating that interaction into Eclipse by allowing the user to define remote targets and the preferred method of communication. The project aims to provide management and communication to a wide range of devices, from FPGAs attached to a hardware debugging device to a full-blown Linux distribution running *ssh* daemons.

Once a communication channel has been established with the device, the target management project will also cover how to manage the device by being able to query its state and starting and stopping processes, services, and cores. These features will afford the engineer complete control over the target from within the Eclipse environment.

This month, the target management group anticipates the open source release of the IBM Remote System Explorer (RSE) plug-in, which has been deployed to IBM customers the past four years. The RSE will allow users to inspect running processes, upload/download files and inspect a Linux target’s file system. The RSE’s

data model will serve as the foundation for additional systems supported by the tool in the future.

Subsequent milestones for the target management project in summer 2006 are to build a remote launch facility that leverages the infrastructure and data model exposed by the RSE, as well as that of the device debugging project.

DSDP device debugging

Even the best coders spend most of their time debugging. Vendors in the embedded space that support remote debugging within Eclipse do so by taking advantage of the existing local debugging code base, then adding code that initiates a remote connection during the phase where the underlying debugger (typically GDB) starts. Once GDB starts, it supplies the information used by Eclipse to render the views in the user interface, and the user is presented with the same interface as used for local debugging.

While this approach provides a good user experience, it suffers from some limitations. For example, a user cannot attach to a running process on the target, nor can the user have more than one debugging session running at a time. In addition, the user interface doesn't efficiently collect the necessary data to populate the views, meaning slower performance as the low-bandwidth connection to the target is taxed gathering data to drive the display.

To address these issues, the device debugging project is first reworking how the underlying debugging interfaces in Eclipse work to better accommodate embedded engineers. The device debugging team has developed a set of prototype interfaces that allows views to have their own update policies, allowing each view to have a different update policy to match the data displayed and the capabilities of the remote target. Due to the complexity of the proposed solution and the coordination efforts between the CDT and DSDP teams, expect to see these features starting to appear this summer.

Conclusion

Eclipse has gained momentum in the embedded market, and the DSDP project is now busy tackling the issues salient to embedded engineers and the vendors supplying them tools. This year promises great advancements in the Eclipse project for embedded engineers and will lay the foundation for continual improvements in the years to come.

Gene Sally is product manager for TimeSys LinuxLink. Gene has been working with Linux for eight years, five of those in an embedded environment, and has been using Eclipse since beta.

For more information, contact Gene at:

TimeSys Corporation

925 Liberty Avenue • Pittsburgh, PA 15222

Tel: 412-232-3250

E-mail: gene.sally@timesys.com

Website: www.timesys.com