Sigasi™
automated hardware refactoring

Why hardware designers should switch to Eclipse

2010-03-17

Introduction

Integrated Development Environments (IDEs) have long been the primary tool for software engineers. Like an airplane cockpit, an IDE is the control center from which the engineer accesses all of the data and tools that he needs. IDEs, and especially Eclipse, have proven to be extensible, open, high quality platforms.

However, until now, IDEs have not been popular in hardware development circles. This is partly because many of the available IDEs for hardware development have not lived up to the potential of IDEs that is typical in the software world. Instead, IDEs tend to be overly complex, closed, and they lock the customer in.

Today, though, Eclipse is finally gaining traction among EDA (electronic design automation) and FPGA companies. One such EDA company, Sigasi, has just released the first commercial VHDL plugin for Eclipse. Now, at last, hardware design teams can use Eclipse as a basis for their own customized IDEs, based on the commercial and open-source plugins that they need in their central cockpit for hardware design.

Cockpit view

The cockpit of a fighter jet is a powerful control center. Everything the pilot needs is right there, within the reach of his hands. The pilot can access dozens of subsystems, including navigation, flight management, guns, and missiles, among others, through hundreds of buttons and dozens of displays.

Head-Up Display of a commercial airplane while it is landing

During dogfights (one-on-one acrobatic fights), a fighter jet pilot needs to keep his attention focused on the enemy aircraft. He cannot afford to keep looking down at his dashboard in order to check his weaponry status. Instead, he uses what is known as a *Head-Up Display* (HUD). This display consists of a transparent screen through which the pilot looks at his opponent. At the same time, the HUD projects extra information like air speed and altitude on the screen. This way, the pilot can keep his *head up* and remain looking at the surrounding environment, while still keeping track of critical data.

HUDs have been shown not only to improve fighting capabilities, but they also improve the capability to fly in bad weather conditions. Today, HUDs are also used in commercial airliners, allowing landing and takeoff even when visibility is low, by *augmenting* reality with additional information. Even high-end automobiles and

Sigasi bvba | F. Rooseveltlaan 348/N | B-9000 Gent | Belgium | T +32 9 265 71 38 | F +32 70 42 57 58 | info@sigasi.com | KBC 737-0232932-78 | BTW BE 0895 543 788 | RPR Gent

motorcycle helmets are now being equipped with HUDs, which illustrates their broad applicability.

A HUD is not a substitute for pilot skills, but it enhances the pilot's capabilities to levels that are *unreachable* without this technological support.

Toolkit mess

Like an airplane pilot, any electronics engineer or software engineer needs all the technological support he can get.

Unlike a pilot, the typical hardware designer does not have all of the controls he needs at the tips of his fingers, nor does he have a head-up display. Information is scattered across different files: documentation, high-level models, RTL models, build scripts, simulation results, wave traces, synthesis log output, etc. Different files that deal with the same part of the project *might* contain consistent information – or they might not.

Inspecting all of these different files requires different programs, many of which barely interact.

Suppose an engineer is checking the log output of a synthesis run. If a warning shows up in the logs, he then has to manually navigate to the corresponding design file and line number. However, once he finds the location, does he remember the relevant details about the warning or does he again have to switch tools to look them up?

Similarly, a small bug can become extremely costly if is detected late: processing the bug report, compensating your customer, chasing the bug down, fixing it...all add time, which means money. Managers should seriously consider every single tool that can possibly help avoid this mess. They should also look for ways to increase the overall productivity of their engineering teams. Bad decisions, bad tools, and bad processes kill productivity.

IDE is the cockpit for developers

Software engineers, like hardware designers, continuously work with code, documentation, build scripts, and logs. What can they use as their cockpit?

For decades, software engineers have looked for and built platforms that improve the usability of their tools and the quality of their work. Since the 1990s, the generally accepted cockpit for software development has been the Integrated Development Environment (IDE). IDEs have come in different flavors and have been produced by different companies. Borland Developer Studio, Microsoft Visual Studio, IntelliJ IDEA, Netbeans, and Eclipse are some well-known examples.

The IDE is, in many ways, a cockpit for the software engineer. It is the single access point for interactive compilation, debugging, navigation, and editing of code. However, IDEs do not only serve as a cockpit, they also offer advanced HUDs. The IDE will flag syntactic and semantic errors immediately in the code, similar to spell a checker. It will also provide extra information, for example about data types, in mouse-over pop-ups.

Modern IDEs even have autopilot-like features. They will suggest automated fixes for common mistakes, like using an undeclared variable. IDEs also help in performing tedious and repetitive code modifications, while still leaving the helm – and the ultimate decisions – to the engineer.

Sigasi bvba | F. Rooseveltlaan 348/N | B-9000 Gent | Belgium | T +32 9 265 71 38 | F +32 70 42 57 58 | info@sigasi.com | KBC 737-0232932-78 | BTW BE 0895 543 788 | RPR Gent

The IDE's job as prominent tool is not finished until the software engineer is completely satisfied with his code. At that time, other tools, like the automatic test, build and integration systems kick in. These tools do not require the same level of interaction that the IDE provides and they are often executed as scripted batch jobs.

Eclipse

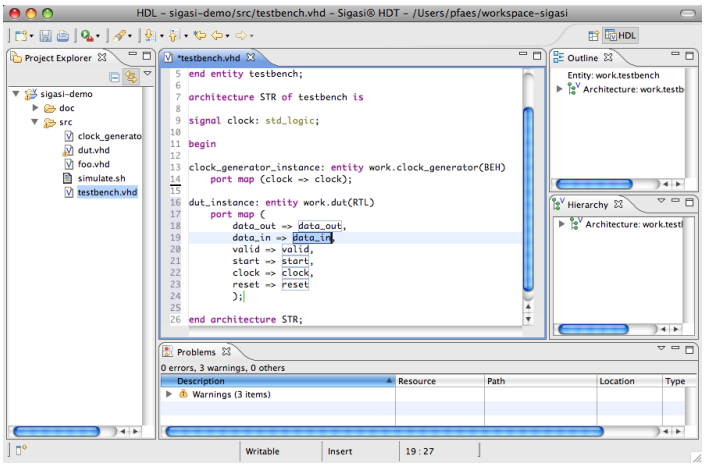
Eclipse is an open source IDE with an open and extensible plugin system. It is an open platform that nurtures a thriving community of companies and volunteers who use Eclipse as the foundation for their projects.

Whatever revision control system you use, whatever programming or scripting languages you use, whatever issue tracker you use, there is a very good chance that you will find support in one of over a thousand available plugins. Each team can configure their own flavor of Eclipse, tailored to their specific needs.

With the introduction of Eclipse, IDEs are no longer proprietary closed systems. Support is not limited to a monopolist vendor: hundreds of companies and contractors provide support and build custom plugins.

Eclipse for hardware development

IDEs are a self-evident choice for software development. Even in EDA-land, Eclipse is omnipresent for embedded software development.[1] Both major FPGA vendors, Xilinx and Altera, use the power of Eclipse as an open platform for the development of their embedded software development tools. So – why is it not used for hardware design?



Sigasi's VHDL plugin for Eclipse

In hardware design, there is no big culture that uses IDEs. Quite the contrary: some early IDE attempts were overly *complex* and *locked in* the customer. There was not enough control over the project build cycle (simulation and synthesis) and people were disappointed with the new user interface, which over-emphasized the use of a mouse over keyboard shortcuts.
These experiences have scared away some electronics engineers from IDEs in general, which is a shame.

For EDA vendors with a near monopoly position, closed systems have been ideal to lock-in their users. Hence, they were not motivated to plug their tools into an open, collaborative system. However, since EDA companies do not have a track record of building intuitive graphical user interfaces (GUI), they should seriously consider

1) The list of official Eclipse Associate and Solution Members includes such companies as Mentor Graphics, Altera, Xilinx, Tensilica, ARM, Ericsson, Freescale Semiconductors, Nec and Intel.

Sigasi bvba | F. Rooseveltlaan 348/N | B-9000 Gent | Belgium | T +32 9 265 71 38 | F +32 70 42 57 58 | info@sigasi.com | KBC 737-0232932-78 | BTW BE 0895 543 788 | RPR Gent

moving their tools to Eclipse. This would be a win-win situation for both the EDA tool users and the EDA vendors. The users get a well-thought-out user interface that allows for extensive customization, while, on the other hand, the EDA vendors are able to focus on their core competence: writing EDA software.

Another reason why so few hardware designers use Eclipse today is the scarcity of EDA plugins. Recently, only a few open source projects existed that made a fair attempt at providing a plugin for hardware description languages. However, now that Sigasi has released its Hardware Development Toolkit (HDT) as an industry ready Eclipse plugin, the tipping point has been reached.

More and more, the industries of EDA and electronics are discovering Eclipse and are slowly embracing it.

Where to from here?

As increasingly more EDA tools become available as Eclipse plugins, Eclipse truly becomes the central cockpit for hardware developers, complete with a nice head-up display. Instead of learning to use several different tools with different keyboard shortcuts and GUI conventions, engineers will only have to face a single, smooth learning curve. Eclipse is a future proof platform that allows you to tailor your design environment to your specific needs.

Now is the moment to take some time to reevaluate IDEs for hardware design. Go download Eclipse, explore the plugins, and experience the power of a state-of-the-art hardware development cockpit.

About Sigasi

Sigasi is an Electronic Design Automation (EDA) company that focuses on the development of agile hardware design tools. The company is headquartered in Ghent, Belgium. For further information, please visit http://www.sigasi.com.

About Eclipse

Eclipse is an open project, focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software. For further information, please visit http://www.eclipse.org.

Sigasi bvba | F. Rooseveltlaan 348/N | B-9000 Gent | Belgium | T +32 9 265 71 38 | F +32 70 42 57 58 | info@sigasi.com | KBC 737-0232932-78 | BTW BE 0895 543 788 | RPR Gent