

GEDAE: A TOOL FOR IMPLEMENTING SOFTWARE RADIO ON HETEROGENEOUS SYSTEMS

James Steed (jim@gedae.com), William Lundgren (bill@gedae.com),
Kerry Barnes (kerry@gedae.com)

Gedae, Inc., 18000 Horizon Way, Suite 200, Mount Laurel, NJ 08054
Phone: 856-231-4458, Fax: 836-231-1403

SDR FORUM TOPIC: 9.2 SDR development tools: Processor concurrence, latency

In order to achieve the throughput and latency requirements of many software radio (SWR) applications, multiple processors must be used. Gedae is an **integrated design environment** for deployed systems and advanced demonstrators based on boards of digital signal processors (DSP) (e.g., AltiVec, PowerPC, TigerSHARC) or distributed networks (e.g., Linux clusters). Its rich **block diagram language** streamlines and simplifies the task of porting applications to parallel systems. The block diagram provides a highly compartmentalized depiction of the algorithm, suitable for partitioning. This block diagram created by the developer specifies only the functionality of the graph, without regard to the target system. Under the direction and control of the user, Gedae is able to use its knowledge of the target (e.g., its processor layout, transfer methods, and optimized routines) to transform the graph into an efficient implementation of the application on the target processors.

Several language features make Gedae particularly powerful for SWR applications. For example, data **streams** are easily specified in Gedae, and the language allows developers to mark **segments** of streams. These user-specified markers on the beginning and end of stream segments can produce side effects that affect graph behavior. An excellent example of such a change in graph behavior is a mode change in a SWR application. Gedae also has a full suite of analysis tools for observing and debugging execution on the host and DSPs,

such as the **Trace Table** where all execution, transfers, and mode changes are displayed.

Increasingly, field programmable gate arrays (**FPGAs**) are being used alongside DSPs as a method for meeting these data flow requirements. These FPGAs are often used to implement front-end signal processing that must be processed at a high throughput. With the increased focus on targets such as FPGAs, the Gedae block diagram language has recently been extended to also enable porting to firmware. Unlike the AltiVec, PowerPC, and TigerSHARC these new targets generally do not allow cross-compilation of C-code. To support other languages, Gedae has been augmented with a single sample graphical meta-language based on the theory of register transfer languages called **Gedae-RTL**. This language is capable of exporting VHDL code for FPGAs as well as Ansi-C code optimized for a DSP. Much like Gedae's core language, the Gedae-RTL graph specifies only the functionality of the graph without regard to the target or its programming language. Through Gedae's knowledge of the target processor, the graph is transformed to generate correct results on the target and for optimized performance on the target. Then target code is exported to implement the application. Components implemented in Gedae-RTL interact seamlessly with core Gedae components, allowing an entire heterogeneous system to be specified in the Gedae programming environment.