ESML: Embedded System Modeling Language and Toolchain

Toolchain for Developing Embedded Mission Computing Applications

Summary of Features

ESML is a domain-specific modeling language to model and construct applications running on a Real-Time CORBA platform, like the Bold Stroke Framework. The language is supported by a set of integrated tools that fully support a model-based design process: component modeling, system architecting and modeling, design analysis, code synthesis and execution.

A generic modeling-based development process (illustrated on the left side figure below) includes

- domain-specific multiple-aspect *modeling* to create models of the system to be built,
- the use of *design analysis* tools to verify the design before code synthesis,
- *generation* that results in synthesized applications (which may incorporate custom code),
- monitored *execution* to run the application under controlled circumstances and to provide feedback to the designer.

The ESML Toolchain implements this process a specific category of component-based, distributed, real-time embedded systems: systems running on the real-time CORBA platform, with "soft" real-time requirements. The toolchain includes the following elements (as illustrated on the right side figure below):

- Component Modeling Tool (Rational Rose currently) that supports creation of domain-specific component models
- System Modeling Tool (ESML/GME visual model editor) that supports the construction of system architecture models describing component interactions, component allocations, fault management, task dependencies, etc.
- Design Analysis Tools (AIRES\(^2\) currently) that supports event and timing analysis (including schedulability)
- Generators and build tool that create the executable application.

The tools are integrated using an Open Tool Integration Framework that allows including other tools into the toolchain if needed.

The supported engineering design process starts with component design and modeling (in Rational Rose), and component functional code implementation (using a traditional IDE). Next, system level architecture models are created using the visual modeling tool (ESML/GME). These models can be analyzed with the help of the design analysis tool and feedback from the analysis can be used to back-annotate the models. From the system models executable “configuration” (or “glue”) code can be generated that gets integrated into the final executable application.

Typical applications of the technology include mission avionics systems, C4ISR systems, robotic vehicle control systems.

---

\(^1\) Original research and development was supported by DARPA/IXO MOBIES program through USAFRL.

\(^2\) Developed by University of Michigan.

Last updated: 5/5/2008