Early Detection of Software Design Flaws with ERNEST

Growing requirements and the demand for more functionality are making embedded systems increasingly more complex, a trend that is evident in the modern automobile. Without validating the quality of such complex systems in the early stages of development, design flaws go undetected, leading to problems in later development phases. That has consequences, because the longer it takes to uncover flaws during the design process, the more it costs to rectify them. With this in mind, Fraunhofer ESK is actively researching ways to tackle this challenge with component-based software development methods.

To achieve this objective, the ERNEST platform relies on a proprietary simulation framework to check their software designs for errors in the timing behavior. Research has shown that only examining the functional aspects of a networked embedded system is not enough to detect flaws. Especially important to ensuring error-free functionality of the system is determining if non-functional requirements such as timing behavior are being adhered to. The increasing integration of various components within a networked embedded system exacerbates this situation however. Driving down development costs and maintaining more robust software thus requires the validation of non-functional requirements in the early stages.

With ERNEST, Fraunhofer ESK researchers are creating an open platform that can be used for the early-stage analysis of component-based software systems employed in the area of networked embedded systems, particularly in automotive applications. The goal is to validate non-functional requirements during the system modeling and design phases when possible.
framework and integrates it in the model driven process with the Eclipse development environment for a flexible use.

Integration and Adaptability

A key design element of the ERNEST platform is its broad utility. For this reason, Fraunhofer ESK researchers made sure that the analysis remains independent of the actual modeling language in which the system architecture is described.

This is accomplished through a proprietary meta model that makes it possible to represent the simulation-relevant information and extract it from a user model describing the system architecture. ERNEST provides a model transformation that converts the user model into a so-called analysis model that is structured on the meta model. ERNEST currently supports models from the EAST-ADL architecture description language. Other transformations are being planned for modeling languages such as UML/MARTE and AUTOSAR/Artop.

Simulation-Based Analysis

The heart of the ERNEST platform is a simulation-based analysis program that relies on a Fraunhofer ESK simulation framework developed under the SystemC description language. The framework aids in simulating the behavior of the modeled embedded system as closely as possible. Code for the ERNEST framework is also generated from the analysis model.

The system creates traces while the simulation is being executed. The logging of the communication behavior occurs at the ports of the analysis model components where the non-functional requirements were defined.

To analyze the resulting data, the information has to be fed back into the analysis model. To do this, the simulation program records the data in a binary trace. The trace information is automatically adopted by the analysis model. This makes it possible to allocate the simulation data to the corresponding elements of the analysis model. The simulation framework can be expanded to support additional trace formats.

Expandable Analysis Tool

The open framework approach and the use of the Eclipse development environment transforms ERNEST into a flexible analysis platform that can be expanded as needed. The ERNEST Eclipse expansion thus provides an extension point for incorporating external analyses for use by the analysis models. The Eclipse modeling framework furthermore offers various options for representing the analysis results. ERNEST currently contains a visualization model for displaying adherence to existing requirements of timing behavior.

Open Source

The base framework and the Eclipse integration are available on an open source basis. Therefore, ERNEST is free to use and enhance as needed. With the support of Fraunhofer ESK and through on-site assistance from ESK engineers if desired, customers are able to adapt the ERNEST platform to their own requirements. Additional functionality, such as support for special bus systems or links to modeling languages such as AUTOSAR, is also available from Fraunhofer ESK via expansion modules. In addition, ERNEST will be continually enhanced through other Fraunhofer ESK projects.

The ERNEST platform consists of the following features:

- ERNEST Analysis model and framework*
- ERNEST Code-Generator*
- ERNEST Simulation Framework*
- ERNEST Visualization*
- EAST-ADL-BbW-Case-Study
- EAST-ADL-Transformation to ERNEST
- EAST-ADL-Feedback from ERNEST
- Binary Trace Reader

It is available at: https://s.fhg.de/ERNEST-en

* Source-Code available on Github: https://github.com/FraunhoferESK/ernest-simulation-framework
https://github.com/FraunhoferESK/ernest-eclipse-integration