Implementation of ECU Configuration Framework based on AUTOSAR Methodology

Joo-Chul Lee*, Tae-Man Han
161 Gajeong-Dong, Yuseong-Gu, Daejeon, 305-350, KOREA
Tel: +82-42-860-1021
Fax: +82-42-861-5404
{rune, tmhan}@etri.re.kr

Topic Area: Systems & Software Engineering

1. Introduction

AUTOSAR (AUTomotive Open System ARchitecture) is a partnership of automotive manufacturers and suppliers working together to develop and establish a de-facto open industry standard for automotive E/E architectures. AUTOSAR defines software architecture, methodology, and application interfaces. The methodology describes ways to exchange formats or description templates to enable a seamless configuration process of the basic software stack and the integration of application software in ECUs and it includes even the methodology how to use this framework. The configuration process is divided into two major steps such as “System Configuration” and “ECU Configuration”. Software architecture design, hardware topology, network configuration, and system mapping are done in system configuration step. BSW (Basic SoftWare) module configuration and source code generation are done in ECU configuration step. As results of ECU configuration step, the source codes generated from ECU configuration step are merged with the BSW module sources, and they will be the complete sources for target system. This paper describes implementation of ECF (ECU Configuration Framework) for configuring parameters of BSW modules and generating source codes reflecting configuration results of BSW module parameters.

2. Related Study

2.1. AUTOSAR methodology [1]

AUTOSAR methodology is a kind of common technical approach for some steps of system development. AUTOSAR methodology uses SPEM (Software Process Engineering Meta-model) which is defined by the OMG (Object Management Group). SPEM is not described in this paper.

![Figure 1. Overview of AUTOSAR Methodology](image)

AUTOSAR methodology includes design steps from the system level configuration to the generation of an ECU executable. The result of each step is delivered to the input of next step in XML format.

Firstly system configuration input has to be defined. This step is not described in Figure 1. System configuration input includes software and hardware components and identified overall system constraints. AUTOSAR provides templates to ease formal description of these initial system designs.

In configure system step, software components are mapped to the ECUs with regard to resources and timing requirements. The output of this step is “system configuration description”. This description includes all system information (e.g. bus mapping, topology) and mapping of which software component is located on which ECU.

After getting system configuration description, tool extracts ECU specific description from it.

Configure ECU activity is the scope of this paper. In this step all necessary information for implementation such as task scheduling, necessary BSW (Basic SoftWare) modules, configuration of BSW, assignment of runnable entities to tasks, etc.

In the last step, generate executables, source codes based on ECU configuration description delivered from previous step are generated, and these source codes are compiled and liked together.
3. **ECF design**

ECF is composed of following subcomponents (Figure 2). AutoWorks is the name of tool we are developing, it follows AUTOSAR methodology.

![Figure 2 Overall ECF design](image)

ECF is implemented in the form of eclipse plugin. It provides interfaces for each functions (BSW module editor, BSW module generator) to other BSW modules. It composes UI by the help of AutoWorks IDE, and it also can access model through RM/EMF.

3.1. **BSW Model Management**

ECF provides eclipse extension points to the BSW modules. Each BSW module describes extension point, and therefore it can be looked up by user. Users can select BSW module from the list, and configure parameters and generate source codes by using each BSW module.

3.2. **BSW Module Editor**

ECF BSW module editor function lets the user edit configuration parameters. BSW module editor function abstracts common UI and functionalities by providing `ModuleConfigurationEditor` abstract class. BSW module programmer should inherit this class and implement module specific UI. This class also provides control class for each parameter and parameter handling class.

3.3. **BSW Module Generation**

ECF BSW module generation function provides `IModuleCodeGenerator` interface. BSW module programmer should implement this interface to add source codes generation functionality. ECF adopts velocity template engine to generate variable source codes [8]. BSW module programmer should make velocity template and dispose configured parameter values in velocity template, then velocity engine merges these values with template and generate source codes.

3.4. **BSW Module Verification**

This functionality provides verification interface to check integrity of parameter configuration results. Each BSW module should check integrity and report check result to ECF.

4. **ECF BSW module editor design**

Architecture of BSW module editor module is like Figure 3.
BSW module plugin provides UI for editing configuration parameters. BSW module editor is composed of many configuration pages. Each page represents a parameter container in AUTOSAR meta-model, therefore it contains many parameter editing controls in it.

ECF provides editor, page, parameter control, and BSW module programmer can implement editor components by inheriting them. BSW module editor interacts with ECF to support navigation among pages and error locating. It also interacts with RM/EMF plugin to access and handle model data representing configuration parameters. Module navigation view from AutoWorks IDE plugin helps BSW module editor to navigate among pages fast. It manages containers (editor pages) as tree structure and informs BSW module programmer of each container’s current position.

5. **ECF BSW source codes generator design**

Figure 4 shows ECF BSW source codes generator architecture. BSW source codes generator uses velocity template engine plugin [8]. User selects “ECU generator” to generate source codes reflecting configuration parameter values.

At the initial phase of source generation procedure, source code generator verifies integrity of parameter configuration results. If there is any problem code generation procedure will stop and user should fix problems. If there is no problem it starts to generate source codes. The source codes generator passes variable code templates and VelocityContenst data including configuration results. ECF initialize velocity engine, and generate source code by using these information. The generated variable source codes are placed output directory.

Source codes generator also passes invariable source codes for BSW modules. ECF just copies these source codes to output directory. Now user can get complete source code configured for the specific target system.

6. **Conclusion**

AUTOSAR requires a common technical approach for some steps of system development. This approach is called “AUTOSAR methodology”. These steps are divided into two main steps such as system configuration and ECU configuration. In ECU configuration steps many ECU configuration parameters based on system configuration
results are configured, and the final goals for this configuration phase is source codes configured for specific target system reflecting user’s configuration parameters.

ECF (ECU Configuration Framework) is composed of BSW module management, BSW module editor, BSW module generation, and BSW module verification function. Through these functions ECF provides BSW module selection, parameter configuration UIs, generation of source code reflecting parameter configuration results, and verification for checking integrity of configured parameter results.

After System Configuration and ECU Configuration, user can get complete source codes set configured for specific target system.

AUTOSAR methodology provides standardized procedures for this works.

7. **References**

[1] AUTOSAR technical overview 2.2.0, Nov. 2007
[4] AUTOSAR BSWMD Template 1.1.0, Aug. 06 2008