A Component-Based Process for Developing Automotive ECU Software*

Jin Sun Her, Si Won Choi, Du Wan Cheun, Jeong Seop Bae, and Soo Dong Kim

Department of Computer Science, Soongsil University
511 Sangdo-dong, Dongjak-Ku, Seoul, Korea 156-734
{jsher,swchoi,dwcheun,jsbae}@otlab.ssu.ac.kr, sdkim@ssu.ac.kr

Abstract. Software plays a vital role in operating modern automobiles, and it is a key element in providing innovative features such as Collision Prevention System. There are two essential issues to be resolved; managing software complexity, and reducing software cost and time-to-market. A key solution to these two issues is to maximize reusing components in building various Electronic Control Units (ECUs). Component-based development (CBD) is regarded as an effective reuse technology. However, current CBD methodologies do not effectively support developing reusable automotive components and ECUs. Hence, in this paper, we first define variability types and variation points for ECUs. Based on the variability types, we propose a component-based development process for developing ECUs. To assess the applicability of the proposed CBD process, we present the case study of developing an innovative automotive ECU for Automatic Parking System (APS).

1 Introduction

For modern automobiles, software is regarded as important as mechanical hardware elements since software monitors and controls various hardware devices and components. Moreover, software plays a key role in providing innovative features.

ECU of automobiles is an embedded microcontroller which provides automobile-intrinsic essential functionality such as controlling engine, operating air bags, and controlling traction stability. A ECU consists of several software components, and the components interact with various setpoint generators, sensors and actuators [1].

However, there are two essential issues to be resolved; managing software complexity, and reducing software cost and time-to-market [2]. Software of modern automobiles nowadays now handles around 80 controllers and hundreds of sensors and actuators connected on multiple bus systems. Hence, such software provides up to 2,500 functionalities, and its size can be up to 10 million lines of code.

A key solution to these two issues is to maximize reusing components in building various ECUs. Among the few reuse technologies, CBD is known to be effective for developing automotive software since CBD provides effective features for supporting

* This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. R01-2005-000-11215-0).

J. Münch and P. Abrahamsson (Eds.): PROFES 2007, LNCS 4589, pp. 358–373, 2007.

modularity, capturing commonality into components, customizing variability within components, assembling in *plug-n-play* fashion, and maintaining through replacement.

However, current CBD methodologies do not effectively support developing reusable automotive components and ECUs. Moreover, current research works including [1] do not handle variability among various ECUs and automobiles explicitly; rather they emphasize its significance.

Hence, in this paper, we first define types of variability which can occur in ECUs. Based on the variability types, we propose an effective component-based development process for developing ECUs. To show the applicability of the proposed CBD process, we present the case study of developing an innovative automotive ECU for Automatic Parking System (APS). Then we assess the approach in terms of process evaluation criteria.

2 Related Works

AUTomotive Open System Architecture (AUTOSAR) is a standard architecture for automotive software [3]. In this architecture, AUTOSAR provides a process for generating ECU software using reusable components. However AUTOSAR does not address concrete specification of software components and the method for mapping software components into ECU.

Schauffele's research provides the core process for the development of electronic systems and software as well as methods and tools [4]. The software components are implemented through general software life cycle. However, this research does not provide the process for developing reusable automotive software component which embeds variability. In addition, the core process does not include detailed instructions and activities for developing this software.

Hardung's work proposes a framework which supports automotive manufacturers to reuse software [1]. This framework consists of three processes; core asset development process, product development process, and management process. However, this work mainly focuses in classifying the types of software components and relating the process to the environment such as repository and tools. Therefore, specific instructions and guidelines are not sufficiently provided for developing automotive software.

3 Variability of ECU

In this section, we define types of variability which can occur in ECUs. ECU consists of ECU architecture and several software components as shown in Fig. 1. The ECU architecture is a generic structure for an ECU, and it consists of software components and their relationships. Star icon denotes the places where variability occurs.

Variability on Software Component: *Software component* provides the functionality of ECU and it contains attributes for persistent data. *Attribute variability* can occur in a software component and it denotes occurrences of variation points on the set of attributes needed by components. More specifically, variations on attributes can occur