Refurbishment of Automotive Electronic Components regarding Update Capability of Applications

Nils Böcher

Robert Bosch GmbH
Business Division Automotive Aftermarket, Product Area Electronic Services
Hildesheim, Germany
Email: Nils.boecher@de.bosch.com

***Abstract*— In the desktop multimedia area, modern update techniques are well known and firmly integrated. Fixed update management systems supply nearly automatically the latest software applications. In addition, the system based hardware drivers are also updated until the base system becomes obsolete and no longer supported. Additionally, functionality for data recovery, system and user settings are also given. In the automotive area, the application is based on platform package integrations and delivering states. Hereby, the application will mostly be frozen for the whole product life cycle after the end of line manufacturing process. For future Electrical/Electronic (E/E) architecture, with central functional integrations and domain centralized systems, there is a new challenge regarding the strategies over the product lifecycle. This includes update functionality in the field and after series production, for new functionalities (Car to Car/Environment (C2X) communication standards, automatic driving assistance, certificates) in consideration for latest security and safety requirements. Solutions must be developed in order to obtain a long life cycle regarding the obsolescence of the product.**

Keywords - variant management; reuse; life cycle.

I. INTRODUCTION

The refurbishment of assembly units is a common technique to be able to reuse an affected nonfunctional peripheral equipment for its target operation mode. This is mainly used in long term usage environments [1]. These methods can be sub-classified into two main areas. The first one is the individual repair, where only the functional affected root cause will be mend into a valid state. The second method is the remanufacturing. Hereby, the whole integrity of the assembly unit will be rechecked and reworked while setting the unit to a current state. This also includes software versions and constitutes software states. Today, the main focus of remanufacturing or refurbishment is still on hardware. The number of electronic modules in the cars on the roads is high and the trend to autonomous driving and Car to X communication will increase further the complexity. Customer service includes no longer only hardware exchange. Also, there is a need for software updates or system integration in car workshops. The global spare parts availability and their integration have a significant impact on success and customer satisfaction. In relation to the economic growth, in addition to the natural resource use, it is insufficient to overcome the even higher demands. This rapid growth involves increasing business risk for higher material costs, supply uncertainties and disruptions. With this background, it is necessary to improve the efficiency of the material, sustainable throughout the maintenance of the systems software components. This is especially true with regard to different life cycles for electronics and cars, long aftermarket supply obligation periods, longer warranty periods and increasing complexity of encryption and aspects related to software. Refurbishment activities have to focus more and more on the software applications.

In the future, the requirements of an application or platform development project will place new demands on the methods of a usual product creation process for a new piece of software. The software management with the accompanying support, updates and maintenance services has to be enhanced regarding update capability and long life support. As a rule, a software application for a model series - after release; it is a fixed package for serial production and additionally for spare parts in workshops. Hereby, no software updates in the entire life cycle are planned. Based on the launch management in an industrial production, the software will be specified regarding a reference architecture within the component design. After a customer software release, where additional calibration and review steps are performed, the software is packed and integrated into the product.

From Start of Production (SOP) to the stable series production, additional derivatives for certain software variants will usually be released, as shown in Figure 1. After the number of variants and quantity demands from the customer, the series production enters the stable phase. Until this section of high rate of production, usually no new software change, or engineering change to the product will be implemented.

The life cycle, which is between the End of Production (EOP) and End of Delivery (EOD) of the product, including the latest software variants, typically covers 15 years - due to post serial supply obligations. During this time, the product must be available based on a given forecast. This will reduce the impact on obsolescence due to the End of Delivery (EOD), based on a product life cycle sales forecast over time [1].

 ISBN: 978-1-61208-532-6

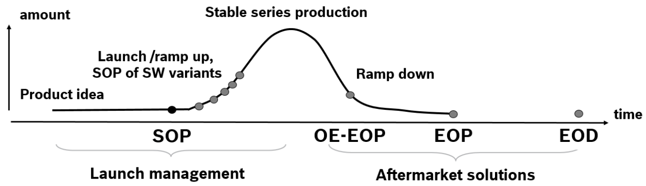

Figure 1. Product life cycle

Nowadays, a classic final stocking of finished products is cost intensive and difficult with respect to the huge model variation, which only differ by software variants within a model series. Hereby, a special software version can be used to have a bigger diversity of usable family version types. However, the product liability and requirements for protection against malicious manipulation and safety classifications in the automotive environment will affect several care strategies over the product lifecycle in the aftermarket. In case of tuning protection techniques, based on software signatures, a new target software without a valid signature is an irreversible incompatibility to the existing system - feasible despite the same hardware variant. Due to increasing security features, a software change is equivalent to a hardware change. A common security method is to fuse a complete block of defined code area in the memory area, by using a One Time Programmable (OTP) functionality of the program flash. In this area, mostly watchdog functionalities supervise the running application. This technique only makes it possible to create an updating option, by using a valid signed software. The signing technique is usually based on an asymmetric key cryptography. Additionally, new control unit cores use an internal Hardware Security Module (HSM). Besides further monitoring, runtime tuning detection regarding manipulated memory blocks and watchdog tasks, the HSM contains a program and data flash layout to store symmetric and asymmetric keys, but also data settings to handle the usage of certificates to protect public keys. This implements password handling for read/write protection, and also for open/close permissions to the debugger interface. A secure log of access entries will be written.

Future Electrical/Electronic (E/E) architectures handle the increasing complexity of the central vehicle functions, while building up a logical centralization and physical distributions. Hereby, the functionality can be partitioned into cross domain based and central vehicle functions on a multi-platform architecture [2]. The requirements for future applications and Base Software (BSW), will bring new challenges by increasing the number of features, which are activated and controlled by the software. Consumer electronic devices will be connected to the car networks, hence a growing complexity has to be expected for the in-car software and 3rd party software integration.

II. ADAPTIVE APPROACH TOWARDS INCREMENTAL UPDATE TECHNIQUES

Current software automotive standards, such as the Automotive Open System Architecture (AUTOSAR) initiative, decouple the application from the basic software package. In this case, the software is using a layer-oriented model, through the introduction of abstraction layers. These layers are separated into services areas. This abstraction opens up new possibilities for an adequate after-series supply strategy.

A suitable method for an adequate after-series supply strategy is the remanufacturing method. Hereby within the framework of the individual repair the software & spare parts can be integrated into the system. This is used to adapt function modules and software components, inside the BSW without altering/affecting the application. For this purpose, a basis must be created such as a non-volatile memory management scheme, such as in [3]. This can be, for example, a layer oriented partitioning so that updating a module in the BSW does not require a complete software container rebuild with flashing of the entire software. This also places new demands on the handling of memory allocation. This approach also requires new tasks with regard to the evaluation [4], and release procedures for hardware and software, in order to meet the latest requirements of operating approvals and aspects of vehicle safety, such as the Automotive Safety Integrity Level (ASIL) classification.

Another aspect is the individualization of the data records in future Electrical/Electronic (E/E) control units. For example this includes calibrated curves, variant coding, etc. This individualization could be used to create a basis for the prerequisite that already existing data from the field can be integrated into the supplier's production, to improve the aftermarket supply strategies. Therefore, an advanced update management is required, regarding runtime functionalities inside the control unit. This can be achieved by using version control, update history or update malfunction protection.

III. CONCLUSION

The approach, of a layer oriented base software design, provides a good opportunity to implement new requirements for automotive applications and future service solutions for update capability, in particular for increased complexity and network topology in the automotive sector, regarding future E/E architectures and its hardware requirements.

REFERENCES

[1] C. Jennings, D. Wu and J. Terpenny, "Forecasting Obsolescence Risk and Product Life Cycle With Machine Learning," in *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 9, pp. 1428-1439, Sept. 2016.

[2] M. Becker, D. Dasari, B. Nicolic, B. Akesson, V. Nélis and T. Nolte, "Contention-Free Execution of Automotive Applications on a Clustered Many-Core Platform," *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, Toulouse, 2016, pp. 14-24.

[3] J. Shin *et al.*, "A Non-Volatile Memory Management Scheme for Automotive Electronic Control Units," *2012 International Conference* on Connected Vehicles and Expo (ICCVE), Beijing, China, 2012, pp. 237-238.

[4] N. Englisch, F. Hänchen, F. Ullmann, A. Masrur and W. Hardt, "Application-Driven Evaluation of AUTOSAR Basic Software on Modern ECUs," 2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing, Porto, 2015, pp. 60-67.