Establishment of Software Development Environment for Multimedia Products

Yasumasa MASUDA
Yoshinori SAKABE
Shozo SUZUKI
Tetsuhiro NOGUCHI

Abstract

For the purpose of efficient software development and higher quality of in-vehicle multimedia products (such as car navigation system), FUJITSU TEN has developed “SWIFT2”, a software development environment, that runs on a personal computer (hereinafter referred to as PC). In SWIFT2, the OS was shifted from μITRON to Windows-Automotive.

Due to the development of SWIFT2, we were able to follow the development know-how on the PC that was accumulated through the SWIFT (software development environment for car audio devices) and utilize the various functions installed in Windows-Automotive. This became possible to achieve efficient software development.

In addition, by using the advantage that the development is possible only by a PC, SWIFT2 has promoted global designing (offshore development) and enhanced skills of our younger software engineers.

1 Introduction

In the previous article (No.31), we introduced our establishment of "SWIFT (Software Integration Framework to the field)", a software development environment for car audio devices. SWIFT is a development tool targeted at software PF (platform) for car audio devices, and its operating system (OS) was limited to μ ITRON.

In this article, we introduce our "SWIFT2", a software development environment for car multimedia products (such as car navigation system). "SWIFT2" adopts the Microsoft's Windows-Automotive (application products of Windows-CE) as target OS, and enables us to develop applications on Windows-Automotive only by a PC.

With the development of "SWIFT2", we realized efficient application development on Windows-Automotive and quality improvement. This article also explains how we achieved accompanying effects using the advantage that the development is possible only by a PC.

2 Transition to Windows-Automotive

The transition to Windows has been discussed since 2005, nevertheless, we have not yet led to actual introduction because of the challenges of real-time operating and boot time, also the restriction of resources (memory capacity: ROM, RAM). However, the functions to be mounted on in-vehicle devices, including mobile phone linkage and internet connection, etc. have continued to expand, and nearly the same functionality as those to be mounted on the PC have become required. In accordance with this changing situation, man-hours for software development are also steadily increasing. We concluded that the transition to Windows-Automotive, which includes a wide variety of middleware and sophisticated HMI (Human Machine Interface) development environment as a standard specification, is required to meet the future functional extension, and thus we decided to introduce Windows-Automotive.

3 Method to Adopt Windows-Automotive

If we adopt Windows-Automotive, we need to develop the BSP (Board Support Package) at first. BSP is a driver software that adapts to the hardware to be targeted and the software that performs initializing process of the device to run the OS. By locating BSP at the lower layer of the OS, we achieved elimination of hardware dependence from applications on the higher layer and Windows-Automotive OS (See **Fig.1**). Since BSP is compliant with the hardware to be targeted, the port of BSP to the PC became the primary task to develop SWIFT2, a simulation environment on a PC.

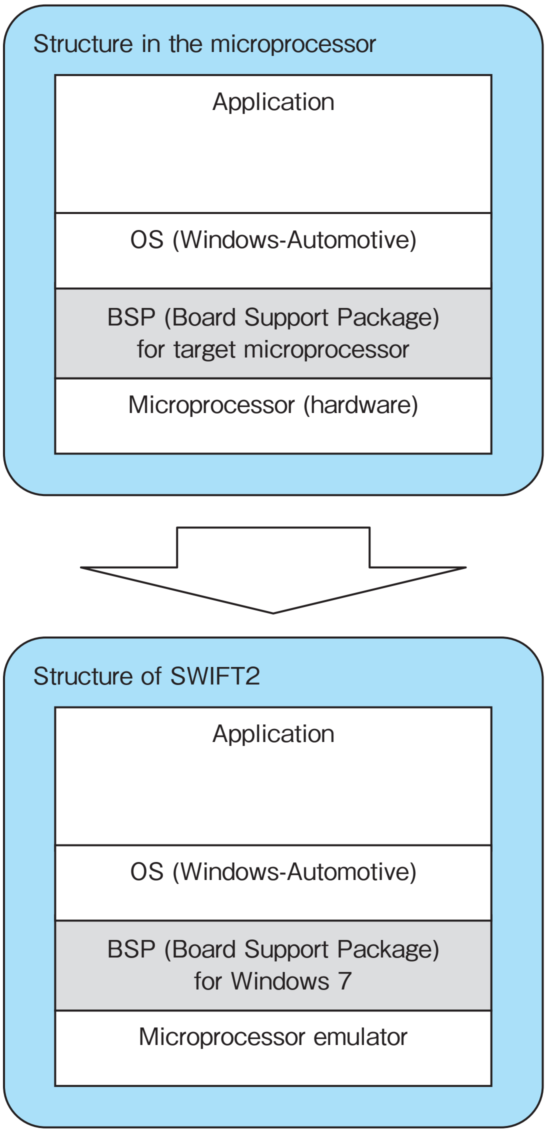

Fig.1 PC Image of Windows-Automotive

However, in establishing the software development environment on the PC, the development environment of Windows CE5.0 Platform Builder developed by Microsoft is fulfilling, and the porting task has been completed only by the port of BSP within a relatively short period. This is also one of the benefits of using Windows-Automotive.

4 Software Structure of SWIFT2

The software structure of SWIFT2 follows that of SWIFT. The main feature is the implementation of virtual devices. In the actual device, the target software is mounted on the microprocessor and various devices (hardware) including DSP, tuner, CD, DVD, etc. are mounted on the periphery of the microprocessor. The software runs with controlling those devices.

If the software is emulated on a PC, it won't run without peripheral devices. Therefore, we had implemented the software to simulate devices as virtual device. In developing the SWIFT2, we have made it possible to divert 100% of virtual device resources developed by SWIFT. As a result, we were able to port Windows-Automotive to the PC and realize the same operation as actual device on the PC.

Table 1 shows main virtual devices.

Table 1 Main Virtual Devices

Virtual device name
LCD device
TFT device
VF device
CD device
DVD device
DSP device
Power device
Switch device
Touch device
Tuner device

A virtual device needs to be created per device. When the actual device is upgraded, the virtual device needs to be modified. We have been developing virtual devices since the development of SWIFT (FY2007), and we have about 120 sorts of virtual devices at present (FY2011). We plan to increase the number to about 150 within FY2011.

5 New Functions of SWIFT2

In SWIFT2, the following functions that are specific to in-vehicle devices have been mounted in addition to the functions already mounted on SWIFT.

(1) Power fluctuation simulation

(2) Integrated virtual device

(1) Power fluctuation simulation is a function that is peculiar to in-vehicle devices. Although two types of power, one is the backup power directly connected to the automotive battery (BU power) and another is the accessory power working with the key (ACC power), are connected to in-vehicle devices, various fluctuations occur in these two powers when the engine starts. For example, although the power of in-vehicle device is usually approximately 12V (volts), the voltage decreases to the degree of 6V to 9V for a few milliseconds when the engine starts.

As evaluation method for this power fluctuation, the evaluation using actual power had been the mainstream in the past. However, in the power fluctuation simulation that we have adopted this time, we can reproduce the power fluctuation on a PC.

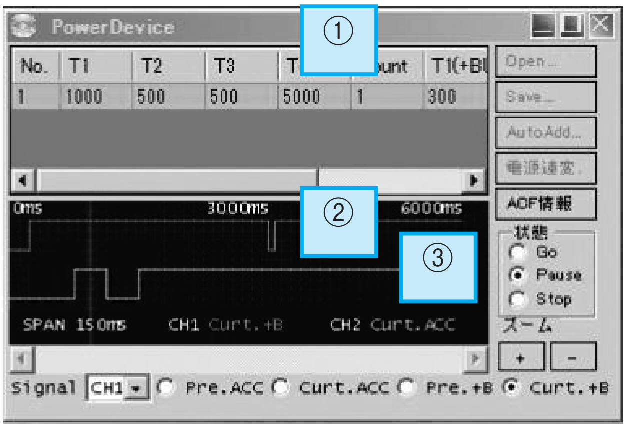

Fig.2 Simulation of Power Fluctuation

Fig.2 is the screen image of virtual device in power fluctuation simulation. There are following functions on ① to ③ shown in **Fig.2**.

①**Set the timing that power fluctuation starts and the power fluctuating time.**

②**Visually display the timing that BU power fluctuates.**

③**Visually display the timing that ACC power fluctuates.**

In addition, considering the accuracy of power fluctuation, we can set the time on the millisecond time scale, and we were able to achieve a great effect on ensuring the quality of power fluctuation.

(2) Integrated virtual device is a virtual device that has a feature that can switch multiple virtual devices dynamically. This is mounted to facilitate the simultaneous development of derived model that occurs frequently among automotive products.

The line-up addition based on destinations (Japan, North America, Europe, etc.) and the line-up addition based on grades (Lo, Mid, Hi) occur commonly among automotive products. By the addition of line-up, there will be various devices to be mounted, for example, XM (satellite radio) and HD (digital radio) functions for North America, and RDS (FM data broadcasting) for Europe. Furthermore, presence or absence of functions such as CD, DVD, USB, etc. changes depending on the grades.

In order to enable these virtual devices to be switched freely and to establish common software development environment, we developed the integrated virtual device.

Fig.3 Integrated Virtual Device

Fig.3 is the screen image of integrated virtual device. By registering these virtual devices in dedicated configuration file, the software development environment for each line-up can be established automatically.

Fig.4 is the screen image of SWIFT2. The efficient software development environment has become reality by locating various types of debug monitors, status of each virtual device, and internal state of software, etc. for easier viewing. We believe that the environment close to the actual device has been completed by importing HMI drawing and product design in the tool.

6 Utilization of Windows-Based Software Engineers

The effect of transition to Windows-Automotive also led to the securing of software development engineers. Mainly, there are following two aspects.

(1) Enhancement of young engineers' skills
(2) Global human resources

(1) Compared to the traditional embedded software, we can master the software development on Windows within a short period of time. Due to the transition to Windows-Automotive, the skills of our younger engineers were enhanced earlier. Although the embedded software had been developed by electronics manufacturers such as FUJITSU TEN and certain software houses, due to the transition to Windows-Automotive, the use of human resources familiar with open environment has become possible.

Fig.4 Image of “SWIFT2”

(2) With regard to global human resources, as we mentioned when we had developed SWIFT, it has been possible to develop the software only with a PC anywhere in the world due to the software development environment that requires no actual device (hardware). FUJITSU TEN has promoted offshore development in the Philippines and China. Due to the development of SWIFT2, we were able to implement outsourcing of software development to the overseas software houses smoothly, and contribute significantly to the offshore development.

7 Example of Utilization in Offshore Development

We are working on the offshore development mainly in the development of derived models. SWIFT that was introduced in the previous article and this SWIFT2 are now the essential tools for the offshore development.

We would like to introduce our new effort to implement offshore development more efficiently; the "software factory". The aims of this effort are to review the state of software development, to build a system to develop the software efficiently just like in the production line, and to reuse the software assets.

In the past development of derived models, if someone other than the designer of base model tried to conduct the development, it was difficult to ensure the quality due to the lack of understanding of base model's development content. Therefore, the base model designer (domestic software engineer) had actually developed the derived models, and we couldn't reduce the development cost. In the effort of "software factory" to overcome this problem, we will build a new development process for derived models based on the idea of XDDP (eXtreme Derivative Development Process), a process model specific to the derivative development, and promote offshore development (**Fig.5**).

[The idea of XDDP]

(1) Clarify the INPUT and the OUTPUT for each software development process, and visualize the difference information of change requirements.

(2) By having a review by the base model designer who understands the whole concept, the lack of current designer's understanding is covered.

Fig.5 shows the development process of current "software factory". As its feature, the domestic work and the overseas outsourcing work are divided clearly, and there is a structure that enables overseas affiliates to focus on the software development (the task of the actual programming).

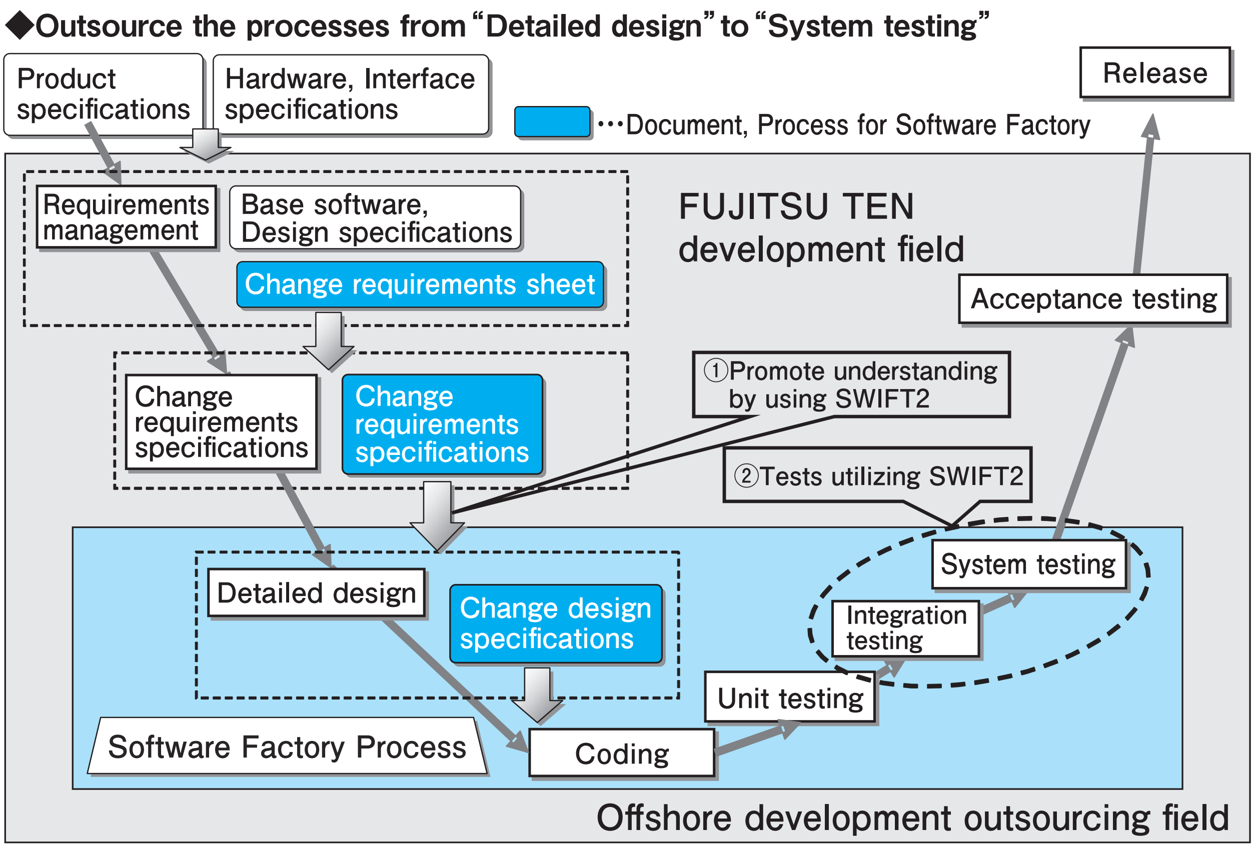

Fig.5 Development Process of Software Factory

In FY2010, overseas affiliated companies conducted the trial in the software development of aftermarket products based on this development process. This trial was conducted specializing in HMI application development, and it was applied to the final development process after repeated trials and errors.

In FY2011, we are starting to officially introduce this process into the derivative development by overseas affiliated companies.

In the future of offshore development, we expect the following two points by the use of SWIFT2.

①Enhancement of the change design specifications associated with the deeper understanding of base model's specifications

②Integration testing and system testing without actual devices.

For the next step, we will also introduce this process into the development outsourcing to the other oversea software houses, and aim to expand the offshore development.

8 Conclusion

Following the previous article (No.31), we once again introduced our development and utilization of software development environment "SWIFT".

The software development environment for automotive devices is making significant advancement each year. The following are the key three points.

- **Improvement of hardware performance**
- **High-capacity of the memory (ROM, RAM)**
- **Popularization of versatile OS (and middleware)**

In addition, as we mentioned in this article, the functions to be mounted on automotive devices are becoming less different from those mounted on the PC.

In the future, we are planning to continue efforts toward further efficiency and quality improvement, including development tool applications, such as SWIFT, in the development process, and we would like to create a structure that allows the tool's applicability and quality to be linked.

Profiles of Writers

Yasumasa MASUDA

Entered the company in 2004. Since then, has engaged in software development for audio products. Currently the Team Leader of Software Engineering Department 2, CI Software Division, Software Engineering Group.

Yoshinori SAKABE

Entered the company in 1993. Since then, has engaged in software development for audio products. Currently in Software Engineering Department 2, CI Software Division, Software Engineering Group.

Shozo SUZUKI

Entered the company in 1989. Since then, has engaged in software development for audio products. Currently in Software Factory Promotion Department, Software Engineering Group.

Tetsuhiro NOGUCHI

Entered the company in 1987. Since then, has engaged in software design for audio products. Currently the Department General Manager of Software Engineering Department 2, CI Software Division, Software Engineering Group.