Inspirium HMI-Studio: Authoring System for Creating HMI Scenarios

● Shunichi Hirabayashi

Performance enhancement, multifunctionalization and diversification of devices that use imaging LSIs are in progress and the scale of software installed is increasing. Hence, a major issue for embedded device manufacturers now is how to develop software while keeping the development costs down. Above all, in devices with many screen pages a large part of the software consists of the user interface (UI), and reducing software development costs especially for this UI portion is a significant challenge. In conventional software development for a UI, first the UI specifications are established, and then the UI is implemented and the design is built; the operability is verified only after it has been installed in an actual device. Accordingly, reworking tends to occur frequently in the following production processes. To reduce this amount of reworking it is necessary to brush up the UI specifications in the UI specification establishment phase. Meanwhile, software development tools for UI implementation are commercially available in large numbers but no UI development tool capable of brushing up UI specifications in the UI specification establishment phase has been prepared. This paper describes the excellent features of Inspirium HMI-Studio, which covers the entire UI development process from the UI specification establishment phase to the following production processes and reduces the total UI development costs.

1. Introduction

Embedded devices that use imaging LSIs are becoming increasingly high-performance, multifunctional and diversified, and this has led to the scale of software increasing at an annual rate of 30% or more. Hence, a major issue now is how to develop software while keeping the development costs down. Above all, in devices with many screen pages such as multifunction printers and in-vehicle devices, the number of user interface (UI) screens per model may amount to between a few hundred and a few thousand, showing that the UI portion accounts for an increasingly large proportion of the software. For example, of the software development costs, the UI software has proven to account for as much as 60% for one customer. As a result of interviews with many customers, the UI software development cost in the development of embedded UI devices in Japan has been estimated to be about 60 billion yen annually in total.

It has been found that, with reference to the specifications of UI screens established in an upstream process, certain reworking is generated in a downstream process for about half of all screens, and this causes longer UI development periods and increased development costs. Accordingly, how the UI specifications can be brushed up in the UI specification establishment phase is an important point in reducing this reworking. Meanwhile, software development tools for UI implementation are commercially available in large numbers but no UI development tool capable of brushing up UI specifications in the UI specification establishment phase has been prepared.

In addition, the scale of UI software development has been expanding and the development processes have been increasingly specialized and subdivided on the sites of embedded UI development. For that reason, without a tool that takes into account issues that span the entire development process, any improvement made is only effective for the individual subdivided processes and not sufficient for the entire

development process.

Inspirium HMI-Studio (hereafter "HMI-Studio") presented in this paper encompasses the entire UI development process ranging from the UI specification process to the software design, implementation and testing processes, minimizes the reworking generated and reduces the UI development periods and costs. This paper first presents issues in UI development and approaches to their resolution. Then, it outlines the functions of HMI-Studio and describes its cost-reduction effect, followed by future challenges.

2. Issues in UI development

As with most other types of software development, the development of UIs for embedded devices can be roughly classified into the specification, design, implementation and testing processes. The development of software for embedded devices has fundamentally been maintained by professional expertise. However, a greater volume of development has given rise to the various issues shown below in the respective processes for achieving reduced development costs and shorter time-to-delivery.

1) Issue in specification process

Under restrictions including limitations to the size of the screen and limited hardware resources available, UI screens in embedded devices need to have sufficient visibility, operability and functionality to ensure they are user-friendly. Another aspect is that performance enhancement, multifunctionalization and diversification of embedded devices are progressing, and this has led to a greater number of UI screens prepared. Accordingly, the conventional technique of describing UI screen specifications and reviewing the design is becoming inadequate for evaluating the visibility, operability and functionality of such screens.

While some portions are actually prototyped for evaluation on the actual sites of development, in view of the limited periods and limited development resources, all portions cannot be prototyped and the UI screen specifications are evaluated on paper for the most part. This means that the UI screen specifications are reviewed based on an imagining of the UI screens on paper and the UI specifications often need to be reworked when the actual UI screens are operated after implementation in the product in a downstream process. The issue in the specification process is how to increase the accuracy of the specification review to prevent reworking in a downstream process.

2) Issue in design process

Generally in the design process, functional software call logic is designed based on UI specifications established in the specification process. However, phenomena not in agreement with the UI specifications often occur, such as calling the functional logic in a wrong way and being unable to call the logic in the first place. This is because of the inability to sufficiently examine specifications including the feasibility of functions in the specification process, leading to inconsistency with the functional software. For that reason, the design of functional software call became complicated in the design process, and this sometimes meant the UI specifications needed to be reworked and the functional software modified. The issue in the design process is how to eliminate the need for matching between the UI screens and functional software so as to prevent extra software development.

3) Issue in implementation process

As embedded devices have recently been rolled out on a global scale, corporations are tending to provide region- or country-specific specifications more and more. For example, some devices are offered in as many as 25 countries. That is where product line expansion becomes an issue. Generally, a core model is developed first, based on which country-specific specifications are customized for product line expansion. In many cases, the functions provided are different according to the countries or regions in which the products are offered and more than one core model is prepared and used as the basis for product line expansion. Any customization naturally requires designers to implement software separately, and this complicates management and increases the development costs. In addition, the designers must separately implement differences in the UI screen layout for language localization and differences in the UI screens due to differences in the functions provided. In particular, the tasks of managing the source code differences for product line expansion, comparing products with the specifications, and making modifications due to specification changes have made manual coding difficult. The issue in the implementation process is how to prevent extra software development when there are differences between models.

Issues in the embedded UI development process are shown in **Figure 1 (a)**.

3. Approaches to resolution of issues

In order to reduce development costs and have a shorter time-to-delivery, we have taken the following approaches for HMI-Studio to resolve the issues that designers faced in the respective processes: early verification by UI specifications based on model-driven architecture (MDA), significant reduction of reworking by early function matching, and automation of manual work by automatic code generation made possible by the previous steps [**Figure 1 (b)**].

1) Resolution of issue in specification process

To allow designers to verify simulations (early verification) of the UI screen specifications after they have been described, they must first be able to handle the described specifications as information that can be used for simulation by MDA (hereafter "executable specifications"). In addition, specifications must be smoothly describable by anybody even with MDA. For that purpose, we have used modeled UI components and adopted formal specification description for describing UI screen specifications. As a result, designers can easily describe the UI screen specifications by selecting items from a list of candidates and they can thoroughly verify a piece of software's operability in the specification process by carrying out an operability simulation.

2) Resolution of issue in design process

To prevent inconsistency between the UI screen specifications and functional software in the design process, we have built a system that prepares functional software as a specification component which can be handled in the specification description to allow formal specification description in an upstream process. The aim of this measure is to make it possible to describe specifications that match the functional software already in view of the restrictions of the functional software when describing the UI screen specifications. By providing the specification components with combination information, we have made it impossible to describe wrong combinations or wrong UI screen specifications when specification components are used for formal specification description. In this way, the need to design the functional software call logic, or carry out unnecessary matching between the UI screens and functional software, has been eliminated by using specification components to describe the UI behavioral specifications in the specification process (**Figure 2**).

3) Resolution of issue in implementation process

One benefit of the ability to handle the UI screen specifications as executable specifications is that components can be switched in consideration

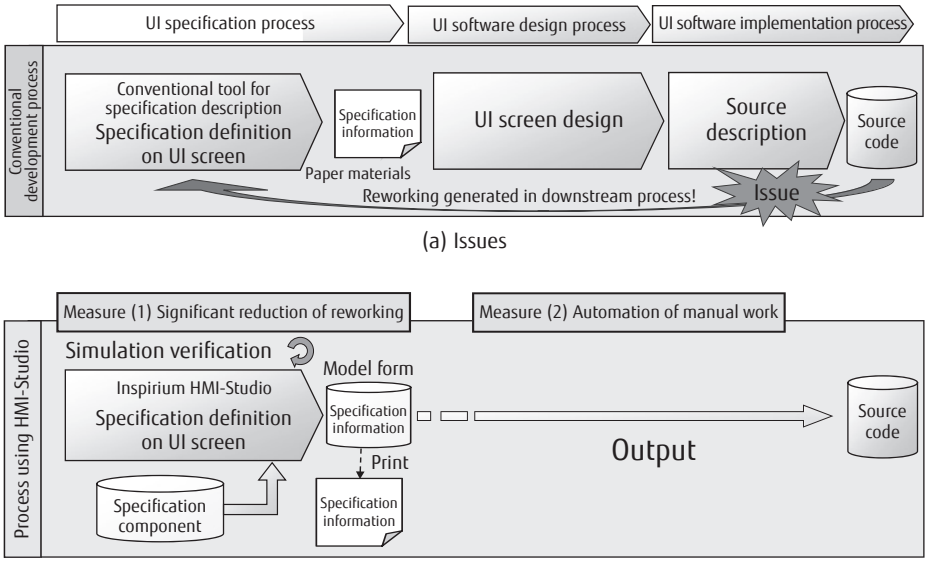

Figure 1
Issues and measures in embedded UI development process.

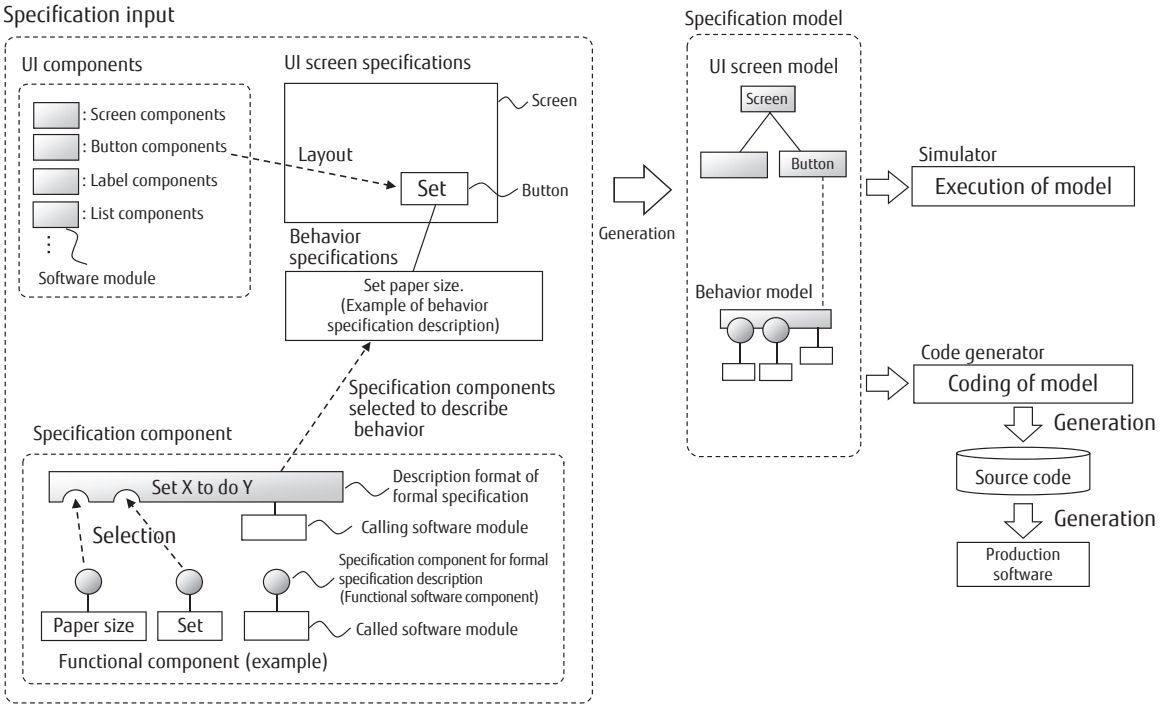

Figure 2
System of specification description using specification components.

of combinations. In the UI specification description process, we have made it possible to describe combination specifications by using specification components. By indicating the portions to be switched or replaced in combination specifications, we have supported software product lines made by the build-to-order method (BTO method, which involves building components based on an order sheet, as opposed to the design-to-order method, which refers to "a la carte" type development of doing work all over from software design). In addition, integration of the respective languages and designs for respective countries and generation of source code directly from the specifications including such integration for localization have been supported. This has allowed manual work caused by product line expansion to be automated (**Figure 3**).

4. Outline of functions of HMI-Studio

Roughly, HMI-Studio has two types of functions: functions for defining UI specifications so as to create UI specifications, and functions for defining UI system specifications so as to define UI system configurations. The former, which includes functions to give specification definitions for UI screen specifications and transition specifications, supports the MDA described above. It is capable of verifying simulations of the established specifications and automatically outputting source code. The latter, which is composed of functions to define the device configuration of multiple display and operation devices and input and output configuration, supports a simulation function capable of verifying visibility and operability with multiple display and operation devices, and this is unique to UIs for embedded devices. The functions of HMI-Studio are shown in **Table 1**.

To respond to organizations' needs for uniformity and a certain level of quality of designs and operability, which is common on operation sites, functions to help designers create specifications, such as the function for standard component creation and common template function that ensures design rules, are supported. In addition, to prevent burdens on operation, the output module has been made customizable according to the descriptions in the specifications in the conventional format used before the system is introduced.

5. Cost effect

This section describes the cost-reduction effect in

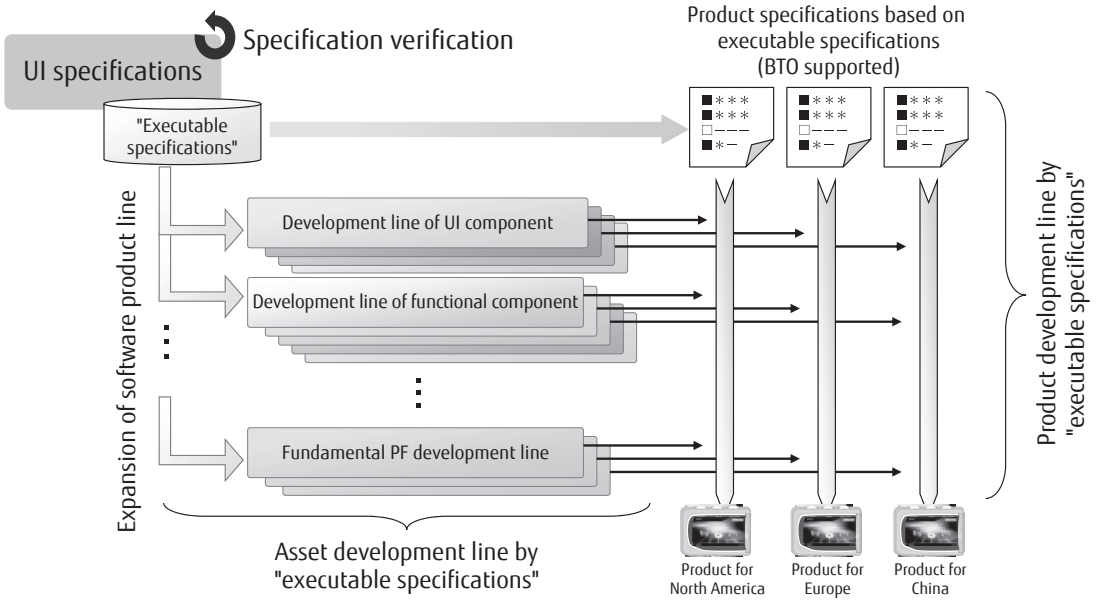

Figure 3
Support for BTO method.

Table 1
Functions of HMI-Studio.

Tool classification	Tool	Description
Definition of UI system specification	Design tool for UI system configuration • Output device configuration definition function • Input device configuration definition function	• Defines input and output devices that exist in the system and allows simulation verification by using the actual devices. • Supports widescreen displays, analog input devices, etc.
	Design tool for UI display section control • Function to define display section • Function to define display control	• Allows definition of the layout and sizes of sections for application display. • Defines controls including the priority and display method as display attributes for the display sections and allows simulation verification according to the mode.
UI specification definition	Design tool for transition specification	• Defines screen transitions including conditions and allows simulation verification.
	Design tool for screen specification	• Provides a means to design UI application screens by using the actual UI components. Allows simulation verification including the actual behavior such as focus movement. • Allows simulation verification by switching between different designs and words and phrases displayed.

UI development produced by introducing HMI-Studio. Regarding the effect relating to the "significant reduction of reworking," in a certain industry, UI screens with reworking generated in the conventional production verification process (downstream process) to render development wasted accounted for almost half of all UI screens. This is because the authority to determine the UI screen specifications was not held by companies that manufacture the devices using imaging LSIs, but by companies to which the devices are delivered. It has been estimated that, by verifying simulations according to the purpose in the specification and design processes to create an agreement in advance between the party in charge of establishing the specifications and party in charge of manufacturing, the amount of reworking generated can be halved, leading to a cost reduction of 25%.

For the effect relating to the "automation of

manual work," an estimate shows that automatically generating source code in the implementation process increases the production efficiency by 35% and, together with the significant reduction of reworking, a cost reduction of 51% can be achieved as a result (**Figure 4, Case 1**). In another case in which the parties in charge of establishing the UI screen specifications and manufacturing are in the same company, the UI screens with reworking generated accounts for 20% of all UI screens. It has been estimated that the amount of reworking generated can be reduced by half and the costs can be reduced by 10%. Based on this estimate, automation in the implementation process is thought to improve the production efficiency by 35% in the same way as above, which, together with the prevention of reworking, has resulted in an estimated cost reduction of 42% (**Figure 4, Case 2**).

Among fields in which the cost-reduction effect is expected to be high is UI development for devices with many screen pages such as vehicle human machine interfaces (HMIs) including on-board devices and multi-function printers (MFPs).

6. Future development

This section presents two approaches to future development of HMI-Studio. One is an approach to further automate the testing process. HMI-Studio internally holds information on UI screen specifications that has been input as executable models. These models can be used for creating scenarios based on the selected specification components, combinations of the boundaries of values that can be taken by the specification components and the specification information defined. These are used as the basis for conducting simulations. The expected program behavior is recorded and compared with the result of outputting the source code, thereby ensuring consistency between the specifications (models) and generated software. In this way, the cost of verification that involved a lot of manual labor in implementation can be significantly reduced.

The other is an approach to improve UI designs so as to have better customer UI quality. The key points are achieving high productivity by separating the designs from the UI specification process, and realizing 2D/3D UIs with enhanced designs. Basically, the model-view-controller (MVC) model is used as the basis of the

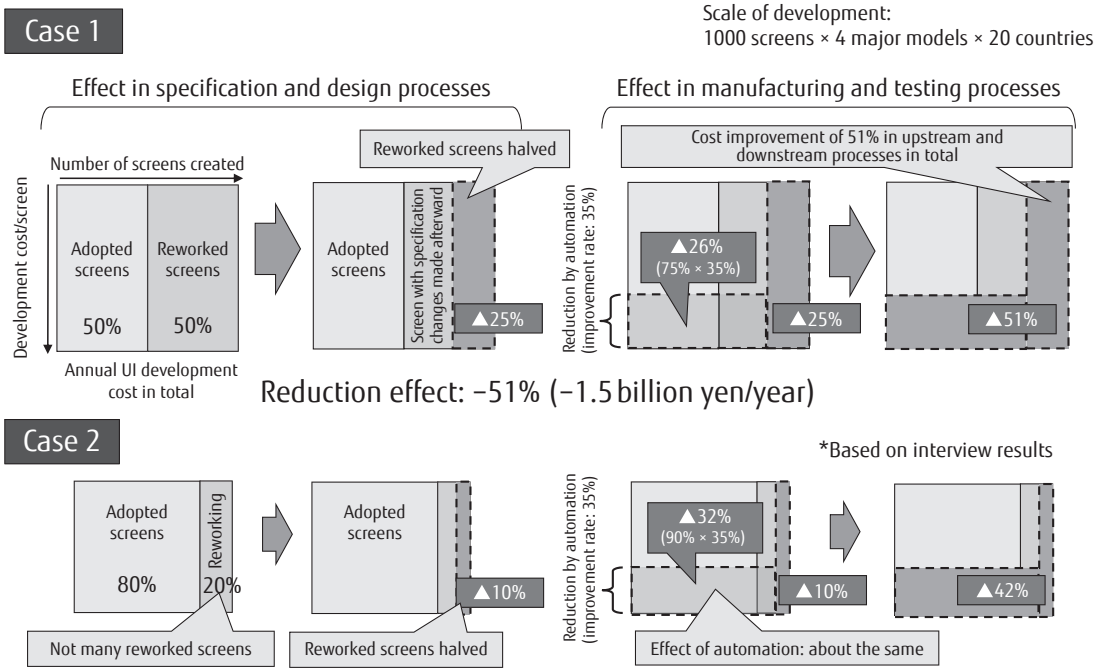

Figure 4
Effect of reduction in UI development cost.

system. Into the UI screen specifications created with HMI-Studio, sophisticated designs that support spatial designs and swipe operation created with tools such as CGI Studio presented in "3D Graphics Authoring System: CGI Studio" contained in this magazine can be automatically integrated. This allows operation simulation verification and UI development. Naturally, simulation verification and automatic source code generation are also possible in this case, offering high productivity.

7. Conclusion

This paper has described how Inspirium HMI-Studio can reduce the development costs of UIs of embedded devices that use imaging LSIs. UI development in many other fields involves similar problems and we believe that the technology developed with Inspirium HMI-Studio is applicable to UI development in those other fields. In view of such application, we intend to continue enhancing this technology in the future.

Shunichi Hirabayashi
Fujitsu Ltd.
Mr. Hirabayashi is currently engaged in development of Inspirium HMI-Studio.