Development of Display Audio (DA) Software for Emerging Countries

Shoh SHINOGAMI　Kiyokazu IWASHITA　Tetsuya FUKUMIZU

Abstract

Vehicle sales in emerging countries have been expanding, especially in the ASEAN. In accordance with popularization of smartphones, the volume zone of the multimedia market toward emerging countries has significantly shifted from Audio to Display Audio (DA) in the markets that are expected to grow in the future.

To make our products competitive in this market environment, we conducted a market research. As a result of the research, we have developed the software in accordance with the following concepts in order to optimize expansion of functions and lower cost.

・Slim function (Low cost system)
　To adopt a System-on-Chip (SoC) on which the power core function, echo cancellation noise reduction (ECNR) function, and video output function are integrated in a single chip, and realize the functions on a low cost system.
・Smart Connect (Connecting function)
　To add connecting function to the low cost system.
・Simple Design (Intuitive human machine interface (HMI))
　To develop a HMI that can be intuitively operated by the generation familiar with smartphone operation.

　This paper explains each concept in detail.

1. Introduction

Vehicle sales in emerging countries have been expanding especially in the ASEAN. In accordance with the popularization of smartphones, the volume zone of the multimedia market for emerging countries has significantly shifted from audio to display audio (DA) in the markets that are expected to grow in the future. At present, we have introduced the following two types of DAs in the market: the DA that includes a connecting function but its price is not low (high DA) and the DA that does not include a connecting function but its price is low (low DA).

This market needs to be examined to ensure that the products are more competitive. Based on the result of such investigations, we need to expand their features by implementing the connecting function and maintain low prices. Thus, we developed a product with the following concepts in mind.

・Slim function (low-cost system)
　Employing the System-on-Chip (SoC), where the power supply core, echo cancellation noise reduction (ECNR), and video output functions are integrated into a single chip and implemented in a low-cost system.
・Smart connect (connecting function)
　Add a connecting function to a low-cost system.
・Simple design (intuitive human machine interface (HMI))
　Develop an HMI that provides intuitive operations for people who are familiar with smartphone operations.

　Table 1 describes the direction of development and the functions that are subject to the development based on the concepts of Slim × Smart × Simple.

Table 1 Direction of Development and Functions Subject to Slim × Smart × Simple-based Development

Item	Development item
Slim function	Porting High DA functions to low-cost system ・AM/FM　　　　　　・USB Music/Video ・Bluetooth® Audio/HF ・iPod/iPhone　　　　・Back Camera ・CarPlay　　　　　　・Android Auto
Smart connect	Implementing a connecting function in low-cost system ・SDL　　　　　　　・Vehicle information upload
Simple design	Creating an intuitive HMI ・Simple GUI　　　　・Sound UI

The following sections describe the developments that have been already realized in accordance with each concept.

2. Concept: Slim Function Low-cost System

The DA for emerging countries needs to be developed as a product with a connecting function within a low-cost range. Therefore, we decided to develop the DA for the emerging countries in the next development stage.

①Limit the quantity of development
　→The rate of appropriable development assets in high DA should be 90% or more.
②Limit the component costs (Fig. 1)
　→Reduce the component costs by reducing the DDR3 memory components by half.
　→Reduce the component costs by selecting a SoC where the power supply core, ECNR, and video image output are integrated into a single chip.
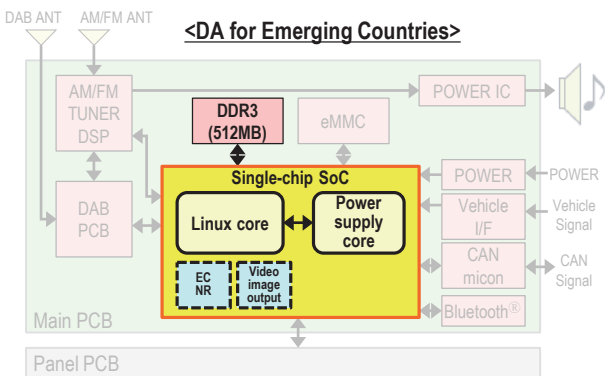

Fig. 1 System Configuration Diagram of DA for Emerging Countries

As a reference, **Fig. 2** shows the system configuration diagram of high DA.
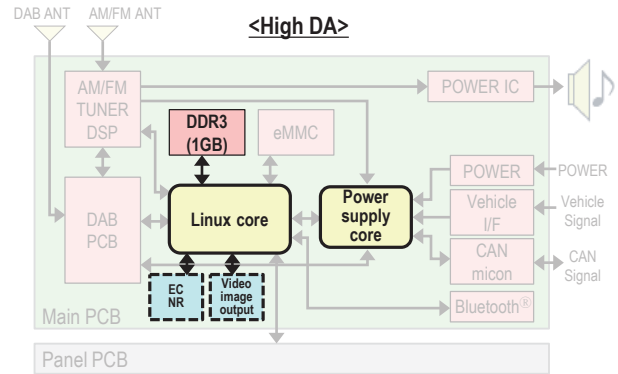

Fig. 2 System Configuration Diagram of High DA

For ①, we selected some functions (e.g., Miracast) that should be removed from high DA. Then, we designed the optimum layout of the development items listed in **Table 1** using the Linux core and power supply core to achieve an appropriable development asset rate of 92%.

For ②, this is the most difficult part in the development of DA for emerging countries. We achieved a cost rate of ▲12% through several improvements.

Some cases in our activities are described in the following sections (2.1 and 2.2).

2.1 Activities for reducing DDR3 memory components by half

Processes with no effect on the functions were optimized and the unnecessary codes were removed; however, the memory size then exceeded 250 MB, which constituted a problem.

To resolve this problem, we performed the following activities to verify the absence of inconsistencies.

・Activity for changing the touch-panel method (electrostatic ⇒ pressure sensitive type)
　→We tuned the pressure-sensitive touch panel (acceleration curve, list speed, click motion threshold, etc.) to keep its operability close to that of the electrostatic touch-panel; thus, we successfully reduced the memory by 11 MB.

・Activity for removing the Open Graphics Library
→We created an intuitive HMI while maintaining a UI with almost the same operability level as that of high DA; thus, we successfully reduced the memory by 242 MB. The details are described in Chapter 4.

As described above, we successfully reduced the memory by 253 MB in total, thereby resolving the problems and achieving reductions in the component costs.

2.2 Activities for developing a single built-in chip

When the processing status of the CPU, including a single built-in-chip, was measured under high load, the load of the Linux core was found to be 62% (MAX); thus, the CPU was not overloaded but the load of the power supply core exceeded 100% (with no IDEL) (Fig. 3).
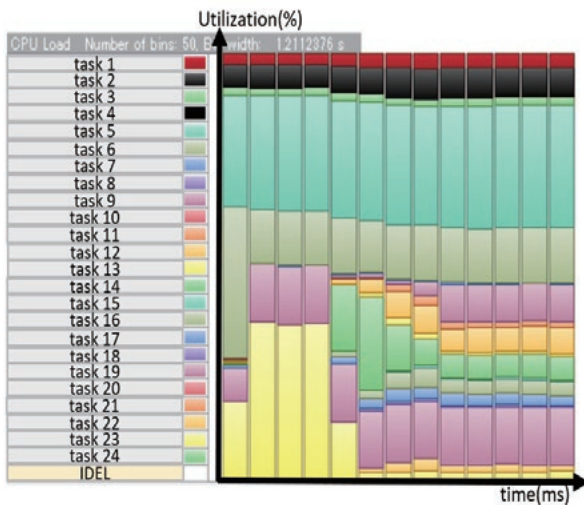

Fig. 3   Measurement Result of CPU Load without Countermeasures

If this situation continued, the functions (camera display performance, audio output requirement, etc.) requested by the customers could not be achieved. Therefore, we set a goal wherein the CPU load does not exceed 80% (MAX) (improvement ▲20%). As countermeasures, we performed the following improvements: memory layout change, task priority change, the removal of unnecessary timers, and others. Consequently, we successfully confirmed that IDEL was secured and the CPU load did not exceed 80% (MAX)

(Fig. 4); thus, we resolved the unsettled problems in customer requests and achieved cost reduction in the components.
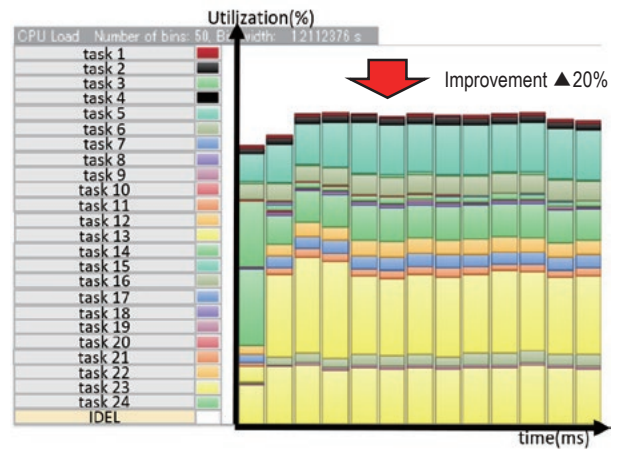

Fig. 4   Measurement Result of CPU Load with Countermeasures

3. Concept: Smart Connect Connecting Function

Considering the connecting functions listed in **Table 1**, the following describes the realization of the Smart Device Link (SDL) and vehicle information upload functions, which are characteristic functions in the low-cost system.

3.1 Activities for SDL function

SDL is an open source platform that can be used to connect applications like smartphones with in-vehicle devices to operate and use them in cooperation.

SDL has a unique characteristic wherein even when the operating systems of the smartphone and the in-vehicle device are different, the user can take advantage of navigation, music, and Internet in the vehicle if it supports SDL.

Fig. 5 demonstrates the basic procedure until the conventional SDL applications starts operating. Conventionally, there has been a problem of "high cost" because if the solution provided by Xevo Inc. (SDL core and a companion application called CAPP[1]) is used, loyalties are charged.

---

* [1] CAPP : Companion Application (launcher application)

(1) Download a companion application onto the smartphone
(2) Download an SDL-specific application onto the smartphone
(3) Connect the smartphone with the in-vehicle device and launch the companion application
(4) Check the white list and only display the icons of the permitted SDL applications on the in-vehicle device
(5) Start the application by pressing the application icon on the in-vehicle device

(2) Download the SDL application

(3) Companion application Launch

Xevo

Smartphone

(1) Companion application Download

Center

Bluetooth®/USB

In-vehicle device

(5) Specify the template for each application

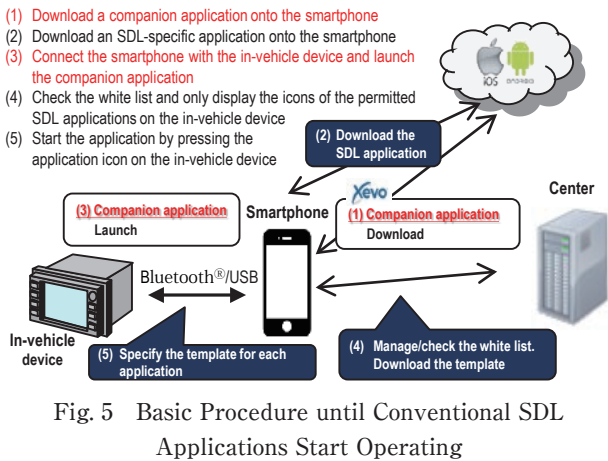**(4) Manage/check the white list. Download the template**

Fig. 5   Basic Procedure until Conventional SDL Applications Start Operating

To resolve this problem, we aimed to realize an original SDL solution where a CAPP function is implemented in an in-vehicle device, instead of the solution from Xevo Inc. **(Fig. 6)**.
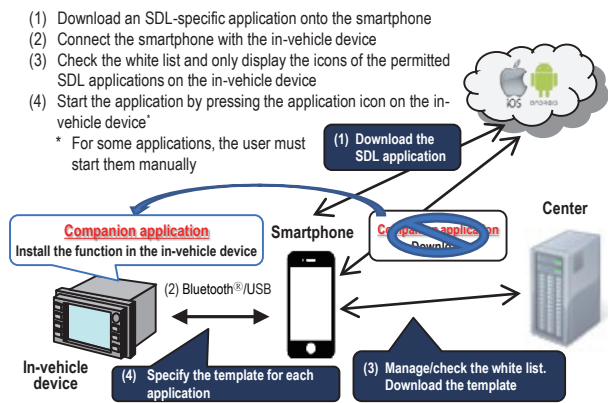
(1) Download an SDL-specific application onto the smartphone
(2) Connect the smartphone with the in-vehicle device
(3) Check the white list and only display the icons of the permitted SDL applications on the in-vehicle device
(4) Start the application by pressing the application icon on the in-vehicle device*
    * For some applications, the user must start them manually

(1) Download the SDL application

Companion application Install the function in the in-vehicle device

Smartphone

Companion application Download

Center

(2) Bluetooth®/USB

In-vehicle device

(4) Specify the template for each application

(3) Manage/check the white list. Download the template

Fig. 6   New Basic Procedure until SDL Applications Start Operating

Fig. 7 shows the configuration of the SDL system developed in this study.

First, a smartphone is connected with an in-vehicle device via USB or Bluetooth®. The system is configured as follows. A communication function module using USB/ Bluetooth® is implemented in the in-vehicle device, and the distribution files registered in the center can be downloaded to the in-vehicle device by connecting it to the center via the smartphone.

In addition, in this configuration, the in-vehicle device identifies the SDL application information by using a distribution file. Then, it communicates with the SDL application body in the smartphone via USB/

Bluetooth® and works in cooperation.

In-vehicle device (DENSO TEN)   **Smartphone**   **Center**

USB or Bluetooth®

SDL Core

Implement the CAPP function in the in-vehicle device

CAPP application

Distribution file

Communication module

Distribution file

White list Policy table Template
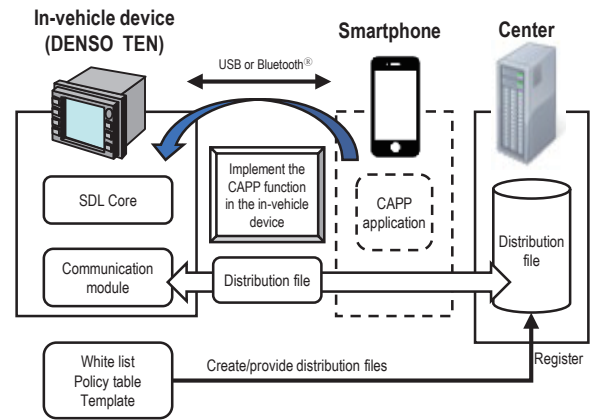
Create/provide distribution files

Register

Fig. 7   SDL System Configuration

In the configuration described above, if the CAPP function is implemented in the in-vehicle device, the following technical issues arise.

①Establishing a method to link a distribution file between the center and the in-vehicle device.

②Establishing a cooperation method that can be applied when the SDL application is started by the in-vehicle device.

For issue ①, the in-vehicle device has to directly access the center. Therefore, we gave importance to the data structure and secure data transmission and defined the following requirements.

・Distribution file data structure.

・Communication protocol between the in-vehicle device and the center.

・Authentication and data encryption/decryption methods between the in-vehicle device and the center.

For issue ②, we employed a UI with which SDL application can be selected via the in-vehicle device even when the target SDL application information in the smartphone is unknown. Accordingly, we defined the following requirements.

・Display the list of SDL applications in the in-vehicle device.

・Cooperative screen transition specifications for SDL applications in the in-vehicle device.

・A method to control the launching of an SDL application for each combination of the connection

route (USB or Bluetooth®) and smartphone OS (iOS or Android).

Finally, we repeatedly conducted the feasibility study according to the requirements specified for each issue. Then, as an achievement, we documented these requirements as specifications.

Fig. 8 displays a summary of the accomplishments.

Conventional CAPP function		Realized result of DA for emerging countries
(1) Communication function	1. Updating "App Info"	1. Instead, use "Policy Table," which is a standard SDL specification
	2. Updating "Head Unit App List"	2. Instead, use "WhiteList" in the in-vehicle device 3. Initially pre-installed (reprogrammed when updated)
	3. Installing and updating templates	4. Develop a tethering module for the in-vehicle device and connect it to a shared network via a smartphone by using Tethering Tethering Bluetooth®. Supporting the download of PolicyTable/Whitelist/Templates
(2) Country selection function		The country selection function is installed in the in-vehicle device where the country information set by the user is stored, and the information is sent to the server to get the relevant URL.
(3) App launcher function		This function allows the user to launch the application. (As an SDL standard, it displays messages to call attention to the user) (only for iOS)
(4) Trusted router function (only for Android)		This function provides multiple (10) SPP communication circuits via the Bluetooth® of the in-vehicle device. The AppManager module is provided in the in-vehicle device to enable app management. (Disable/Enable, Arrange the app order)

Fig. 8   Accomplishment Summary of CAPP-specific SDL Functions

3.2 Activities for vehicle information upload function

The vehicle information upload function is a service that provides vehicle information related to the fuel consumption, mileage, and other warnings to the user. This is accomplished by acquiring the CAN information of the vehicle via the in-vehicle device and uploading it to the center to indicate that the user needs to visit the car dealer's garage.

Conventionally, this involved a "high system cost" problem because a communication module was required in the system to upload data to the center in addition to the DA. Owing to this, a communication module was not mounted in the entry-level vehicles for emerging countries and the vehicle information upload function was not applied to them.

To resolve this problem, we aimed to realize a vehicle information collection service **(Fig. 9)** via the smartphone of the user instead of the communication module.
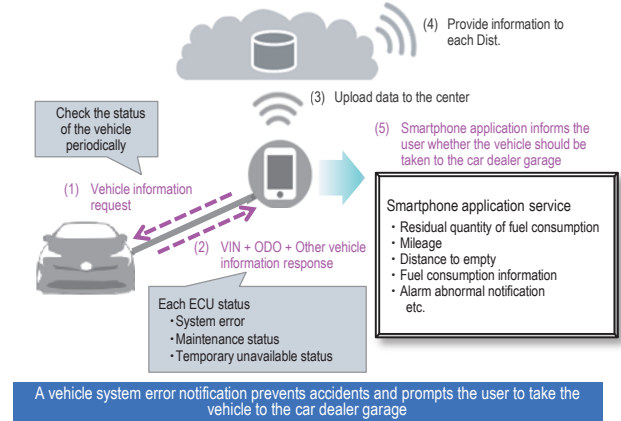

Fig. 9   Vehicle Information Collection Service in DA for Emerging Countries

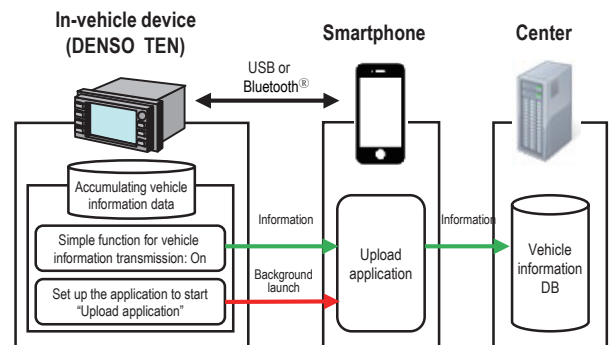**Fig. 10** shows the system configuration using a smartphone.

Fig. 10   Configuration of Vehicle Information Upload System

For the method using a smartphone, it was presupposed that the "smartphone is not always connected," as compared to the conventional "communication module always-on connection." Therefore, the following technical issues were encountered.

①Establishing a method for accumulating data in the in-vehicle device.

②Establishing a method to connect the in-vehicle device with a smartphone and a method for sending data.

For issue ①, it is impossible to promptly send vehicle information acquired from the vehicle.

Therefore, we focused on surely sending minimum amount of data just after transition to the state where data transmission is enabled. Therefore, we defined the following requirements.

・Data accumulation timing in the in-vehicle device.
・Amount of accumulated data in the in-vehicle device and handling of it in the case where it exceeds the upper limit.
・Transmission timing of accumulated data and a condition for releasing data.

For issue ②, the device should simply connect with a smartphone and work with the dedicated smartphone application in cooperation. Therefore, we considered the launching of the application and secure data transmission to be important and defined the following requirements.

・A method to control the launching of a smartphone application for each combination of the connection route (USB or Bluetooth®) and the smartphone OS (iOS or Android).
・Communication protocol between the in-vehicle device and the smartphone application.
・Authentication and data encryption/decryption methods between the in-vehicle device, smartphone, and center.

Thus, we repeatedly conducted the feasibility study according to the requirements specified for each issue. Then, as an accomplishment, we documented these requirements as specifications.

Fig. 11 displays the summary of the accomplishments.

Target operation:
 "Enable uploading the vehicle information from the in-vehicle device to the smartphone"
Conclusion:
 Vehicle information can be uploaded irrespective of the Android and iOS terminals

○: Target operation is available　△: There is a restriction on the target operation　×: Not available

Target operation		Android terminal		iOS terminal	
		Bluetooth® connection	USB connection	Bluetooth® connection	USB connection
Application launch		○	○	○	○
Background launch		△	△	△	△
Vehicle information upload during background launch		○	○	○	○
Vehicle information upload	While CarPlay is in operation	-	-	×	○
	While AndroidAuto is in operation	○	△	-	-
	While SDL is in operation	○	△	○	○

Fig 11　Accomplishment Summary of Vehicle Information Upload

4. Concept: Simple Design Intuitive HMI

With the advancements in operation methods, operation elements have become diversified for the development of in-vehicle HMI. Therefore, there is a demand for developing an intuitive HMI to reduce the operation load of users.

We defined the design concepts of the intuitive HMI, as follows, and realized the HMI in products. The details are described in section 4.1 and 4.2.

①GUI (Graphical User Interface) that allows the user to call up an image of operation.
②Sound UI that allows the user to call up an image of function when hearing a feedback sound.

4.1 GUI that allows users to call up an image of operation

The following section describes the designs and evaluations implemented by us.

4.1.1 Designing base GUIs

・Shortening the operation time: Apply graphics that requires 2 s or less (ergonomics) for using a function.
・Using familiar operations: Apply vertical and horizontal slide operations as smartphone-like operations.

4.1.2 Impression evaluation

To survey the preference tendency of designs that are preferable in emerging countries, we created sample GUIs that provide the following four types of impressions by using the base GUIs.
・"Simple," which is light and generic.
・"Casual," which is light and characteristic.
・"Luxury," which is heavy and generic.
・"Aggressive," which is heavy and characteristic.

A total of 152 examinees from emerging countries (Thailand, Philippines, Indonesia, and Malaysia) were requested to answer a questionnaire about the GUIs described above.

As shown in the result (Fig. 12), "Aggressive" had the highest rating ("Luxury" and "Casual" were also close to it).
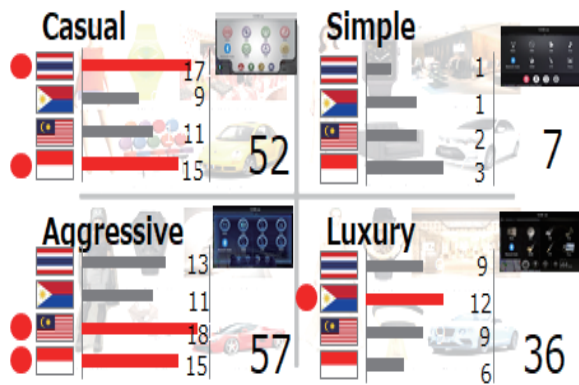

Fig. 12　Preference Questionnaire Result

Based on this result, we employed the GUI that gave an "Aggressive" impression (Fig. 13) as the default setting. We also implemented the "Luxury" and "Casual" GUIs that were close to the highest rating (Fig. 14 and Fig. 15) to include a skin change mechanism.

In addition, we checked how the examinees perceived the employed GUI and confirmed that the GUI is a GUI in accordance with the impression of DA in emerging countries. Thus, we realized a GUI that enables users to call up an image of operation.

Fig. 13　Aggressive HMI Screen of DA for Emerging Countries

Fig. 14　Luxury HMI Screen of DA for Emerging Countries

Fig. 15　Casual HMI Screen of DA for Emerging Countries

4.2 Sound UI that allows users to call up an image of a function when hearing a feedback sound

The following section describes our designs and evaluations.

4.2.1 Designing base sound UIs

・Distinct sound wherein no error of judgment can be made: Generate a sound at less than 2 kHz to distinguish it from ADAS (2 kHz or more).

・Easily recognizable sound: Generate a sound closer to the tendency of multimedia sounds from smartphones (with the application installed) used by users.

Considering the points described above, we selected eight feedback sounds, whose distribution is shown in **Fig. 16**; the horizontal axis represents the sound factor (metallic) and the vertical axis represents the pitch (kHz).
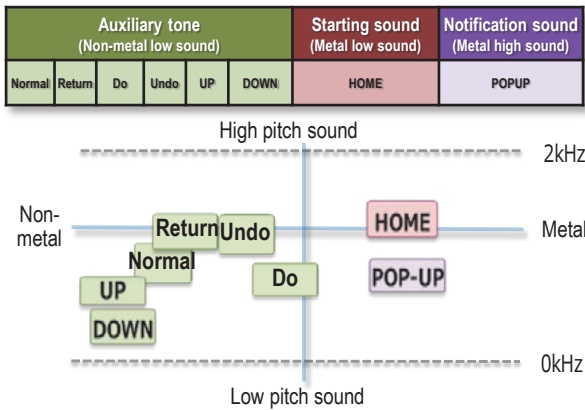

Fig. 16 Tendency of Feedback Sounds of DA for Emerging Countries

In addition, to generate a tone according to the skin change function, we tuned a sound by combining it with a chord.

As the impression of an "Aggressive" sound, a sound that can stimulate "power."

As the impression of a "Luxury" sound, a sound that can stimulate "broadness."

As the impression of a "Casual" sound, a sound that can stimulate "plainness."

4.2.2 Impression evaluation

We invited the staff members with perfect pitch and the examinees at one site and asked them to fill a questionnaire about the impression of operation sounds to check whether the sounds are suitable for the GUIs.

According to the results, two of the eight feedback sounds did not suit the impression of the operation. We tuned the combination of sounds and again conducted a questionnaire to check whether the sound is a feedback sound in accordance with the impression of the DA for emerging countries; thus, we successfully generated the sound UI using a feedback sound that stimulates the image of the function.

5. Conclusion

In the DA market, where the cost competition is progressing due to commoditization, we developed technologies for DA according to the concepts of Slim × Smart × Simple to differentiate from other competitors. We resolved various issues and successfully developed the connecting DA, low-cost product, in association with subcontractors and interested departments.

In the future, it is necessary to surely respond to the changes in the requested functions, which may vary with region, and to work toward flexibly expanding the product range to respond to a coming phase of changes in cockpit IVI.

We will continuously acknowledge user needs and develop products to contribute to a more comfortable mobility society.

Finally, we sincerely appreciate the cooperation of the people who helped in the development of this product.

・Miracast is a registered trademark of Wi-Fi Alliance.
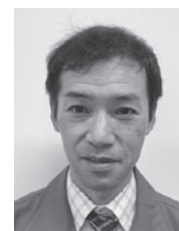・Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

Profiles of Writers

Shoh
SHINOGAMI

CI Business Group
Engineering Div.2
Engineering Dept.2

Kiyokazu
IWASHITA

CI Business Group
Engineering Div.2
Engineering Dept.2

Tetsuya
FUKUMIZU

CI Business Group
Engineering Div.2
Engineering Dept.2