NOTE

Development of HTML5-based In-vehicle HMI

Teru SAWADA
Kokusyou RIN
Yoshitaka HIRASHIMA

1 Introduction

Recently, mobile devices represented by smartphones have been advanced remarkably, and terminals corresponding to a new function and a new service have been placed on the market one after another. On the other hand, regarding in-vehicle devices, in accordance with the prolonged cycle of buying a new vehicle, obsolescence of the functions on an in-vehicle system is actualized, and its divergence from evolution of the mobile devices is regarded as a problem. Therefore, the automotive industry has equipped in-vehicle systems with a communication function, and has been preventing the obsolescence of the functions on the in-vehicle system through the provision of the latest information and services.

Nowadays, a movement to apply the technology and services for mobile devices to the in-vehicle systems becomes active, and the in-vehicle system development using the technology for developing mobile applications has been promoted by respective companies in the industry.

This paper introduces some development examples of the in-vehicle application by use of HTML5 (Hyper Text Markup Language Ver.5) that is attracting a great deal of attention of the automotive industry as well as the mobile industry by its high versatility and open standard specifications, and describes advantages and issues of the in-vehicle application.

2 Evolution and Issues of Telematics System

The communication-linked service for automobiles (Telematics service) which has started inside and outside Japan since the late 1990s equips the in-vehicle systems with a communication module, and provides various services such as provision of the latest information and a SVT (Stolen Vehicle Tracking) system (**Fig. 1**: Dedicated system). In recent years, various approaches such as use of a smartphone-integrated mobile application (**Fig. 1**: Smartphone integrated system), and direct distribution of an application to the in-vehicle system by installing the general-purpose OS used on the mobile devices into the in-vehicle system (**Fig. 1**: General-purpose system) are ongoing.

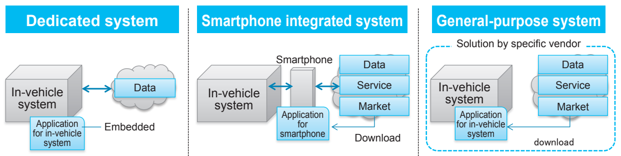

Fig.1 Configuration of telematics system

On the other hand, the current systems face the respective issues shown below.

- **Dedicated system: Individual service development depending on each system specification is required.**
- **Smartphone integrated system: Assessment of conformity with enormous number of mobile devices is needed every time a service is added or updated.**
- **General-purpose system: It is controlled by the business model of a specific vendor providing operating systems or application markets**

HTML5 is the global and open standard of which W3C (World Wide Web Consortium) promotes standardization, and draws attention of the automotive industry as a technology for solving various issues possessed by the current telematics system.

3 APIs and Development Examples of In-vehicle Application

To develop the HTML5-based in-vehicle application, we have applied the following three APIs (Application Programming Interface) to the application.

- **WebGL: Technology for providing screen designs that dynamically change in accordance with an environment such as vehicle speed**
- **WebSocket: Technology for achieving an application in which an in-vehicle terminal is linked with a center**
- **WebRTC: Technology for achieving an application by use of peripherals such as a camera and a microphone**

3.1 WebGL

3.1.1 Technical overview

WebGL is a standard specification that enables drawing of 3D graphics within a web browser on the platform where OpenGL 2.0 or later which is a standard for drawing computer graphics is available. Moreover, the library (Three.js) that makes this API easy to use is also open to the public, and the environment where a three-dimensional application can be relatively easily developed by JavaScript is prepared.

3.1.2 Prototype application

In order to evaluate a driver's distraction (Driver Distraction: hereinafter referred to as "DD") in the case where the complex dynamic design is provided by WebGL, we have prepared a prototype UI (User Interface) in consideration of DD. As an external factor affecting DD, the vehicle speed is the most important parameter. The method for switching between two modes of "during driving" and "during stop" by detecting the vehicle speed is common in the current UI for in-vehicle system. This time, the UI design which varies dynamically depending on the vehicle speed has been implemented by use of WebGL (**Fig. 2**).

Driving at low speed　　　　Driving at high speed

Fig.2 Menu display linked to vehicle speed

3.1.3 Advantages

For design parts (drawing parts) for in-vehicle system, visual effects are often given by a dedicated drawing tool. However, by use of WebGL, drawing processing can be relatively easily provided. Therefore, as compared with the conventional embedded type of development, WebGL can contribute to the efficiency of design development. Also for the prototype UI design linked to the vehicle speed, the dynamic visual effects of the menu displayed can be achieved, only by using Three.js and JavaScript.

3.1.4 Issues

With the capability of a graphics processor (GPU) which is normally installed in the in-vehicle system, when gloss or viewpoint effects or the like of WebGL are applied, events such as flickers of the overlapped icons and operating delay in a flick gesture occur. Especially when rendering of the animation having a high drawing update rate is performed (60fps or more), the GPU load becomes extremely higher. Therefore, tuning in consideration of performance of the hardware to be used is important.

When WebGL is applied to the actual in-vehicle system, the drawing update rate needs to be fixed to reduce the GPU load, and a sufficient response needs to be ensured. In the case where a map of navigation system is drawn, the map may be drawn at the time of GPS update. Thus, a rendering frequency can be significantly reduced.

3.2 WebSocket

3.2.1 Technical overview

WebSocket is a technology that has a low server load and can establish a large number of sessions concurrently compared to the communication by XMLHttpRequest and Comet which are the conventional web communication technologies, and that allows a push notification from the server to the in-vehicle terminal.

3.2.2 Prototype application

By placing the UI design for in-vehicle system in the center with the HTML 5 format, the screen can be shared between the in-vehicle system and the center. Moreover, by using WebSocket corresponding to push communication, the screen of the in-vehicle system can be remotely controlled from the center (operator service). This time, by use of the web server for content delivery and the WebSocket server required for the linkage with intercommunication, the remote operator service has been realized. The image files and the screen configuration / transition data to be shared are placed on the web server (**Fig. 3**), and the in-vehicle system and the external terminal share the content by accessing the HTML content which is appropriate for each of them. In addition, coordinates of touch inputs by operations, information on the central coordinate of the map, screen number, and user ID, etc. are transmitted and received via the WebSocket server so that the mutual state transition is synchronized.

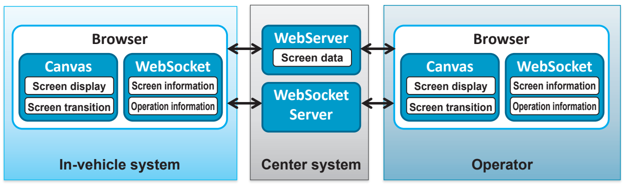

Fig.3 Screen sharing using WebSocket

3.2.3 Advantages

Currently, the system by which an operator remotely controls the screen of the in-vehicle terminal requires implementation of the mechanism by which the in-vehicle system side is highly linked with the operator side, and it has taken root as a paid service for luxury vehicles. However, by use of the functions of WebSocket, such a system can be basically realized both on the in-vehicle system side and on the operator side only by preparing a browser compatible with HTML5. Thus, the service can be easily provided also to entry-level class vehicles.

3.2.4 Issues

For the content having relatively large-capacity data such as sounds and videos, a data relay is performed by the server to transmit and receive the data. Due to the increased amount of communication data, a decrease in communication speed and a data delay occur, thus, a real-time property will be lost. Therefore, WebRTC (To be described below) that is capable of P2P (Peer to Peer) communication without going through a server should be applied. As described above, it is important to select an optimum API according to the capacity of data to be handled.

3.3 WebRTC

3.3.1 Technical overview

WebRTC is an API that enables real-time communication on browsers. Its specification includes the PeerConnection API that realizes the P2P communication, the DataChannel API that is capable of transmitting and receiving the arbitrary data by P2P, and the MediaStream API that is capable of obtaining the data stream and of getting access to cameras and microphones.

3.3.2 Prototype application

Out of components of WebRTC, we have prepared a prototype of the in-vehicle-device-integrated application that uses getUserMedia (a function for getting access to cameras and microphones by MediaStream API) which is the most advanced API in terms of implementation at present. The getUserMedia obtains a video of the locally connected camera from the browser, and the video of the camera of in-vehicle system is displayed on the browser by use of the HTML5-based application. By drawing this camera video on a specific layer using Canvas, the camera image processing which has been generally implemented in the embedded system so far becomes possible to be implemented on the browser by use of HTML5. Moreover, the obtained camera video is associated with the own-vehicle location (the location information is obtained by GeoLocation which is one of the APIs of HTML5) so as to easily realize the Augmented Reality navigation system. The AR navigation application using the above-mentioned technology (**Fig. 4**) and the configuration of the drawing layer (**Fig. 5**) are shown below.

Fig.4 Example of screen display of AR navigation system using WebRTC and Canvas

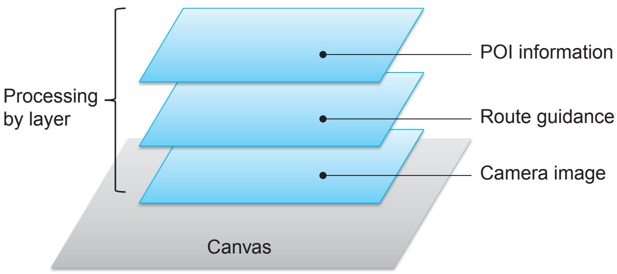

Fig.5 Configuration of AR navigation system

3.3.3 Advantages

Conventionally, in order to perform the camera image capturing and the image processing on the in-vehicle system, the dedicated hardware and the embedded application for image processing need to be implemented in the in-vehicle system.

On the other hand, WebRTC allows the camera image processing and the superimposed display with a variety of drawing parts at the browser level, and applications for AR navigation system and back guide monitors can be relatively easily realized. Moreover, as long as an interface connecting the in-vehicle system to the camera is implemented, different variations of services utilizing cameras can be distributed as the HTML5-based applications. Thus, WebRTC is useful for enhancing the added value of the camera of in-vehicle system.

3.3.4 Issues

Although the processing for superimposing navigation information or the like is executed on the camera video obtained by getUserMedia, when its throughput is increased, the frame rate of the video cannot be ensured. Therefore, JavaScript must be optimized and there is a need to adjust the rate to the frame rate appropriate for the application.

In the current browser standard, up to two cameras are accessible by getUserMedia. When a vehicle is equipped with cameras such as front cameras and side cameras other than rear cameras, the inaccessible situation occurs. Other issues such as a response to the camera with high resolution exceeding VGA still remain.

4 Conclusion

We have explained, out of various technologies held by HTML5, the advantages and the issues regarding WebGL, WebSocket, and WebRTC that are especially prospective and important for realization of the HTML5-based in-vehicle infotainment systems.

The specifications for HTML5 have not reached the final recommendation at the time of writing. However, in the entire W3C-centered industry, the standardization and the implementation of HTML 5 are steadily advancing, and the study of the specifications of HTML5 for in-vehicle systems has started.

As described in this paper, there are specific development requirements for the in-vehicle systems, and the ability of each in-vehicle system manufacturer to respond to the requirements is the big appeal to show its skill. We will continue to develop the prototype for HTML5-based in-vehicle application, and accumulate our know-how.

Profiles of Writers

Teru SAWADA
Department General Manager of Service Engineering Dept.
SS Engineering Group.

Kokusyou RIN
Service Engineering Dept.
SS Engineering Group.

Yoshitaka HIRASHIMA
Service Engineering Dept.
SS Engineering Group.