

## A Study of In-Vehicle Infotainment System Based on HTML5 Using Tizen IVI

Jae-Gon Yoo<sup>1</sup>, Kyung-Sik Jeon<sup>2</sup>, Yong-Ha Lee<sup>3</sup>, Sangphil Kim<sup>4</sup> and Jong-Bae Kim<sup>5\*</sup>

<sup>1,4,5\*</sup> Graduate School of Software, Soongsil University, Seoul, 156-743, Korea

<sup>2,3</sup> Department of IT Policy and Management, Soongsil University, Seoul, 156-743, Korea

<sup>1</sup> worhs0410@gmail.com, <sup>2</sup> ksjeon@inovacnc.co.kr, <sup>3</sup> dufma@korea.kr,  
<sup>4</sup> sangphilkim@ssu.ac.kr, <sup>5\*</sup> kjb123@ssu.ac.kr

### Abstract

Cars are the most essential and important means of transportation to modern people. However, the service paradigm of cars, which were just a simple means of transportation in the past, is being changed in combination with recent IT technology. In-Vehicle Infotainment system (IVI), with this flow, is not only playing an existing navigation role. However, unlike mobile OS market which Android and IOS occupy the most, the In-vehicle infotainment market is still in competition. Technical competition of related companies gets deepened to preoccupy this new market, and IT companies are also participating in competition one after another. Typically, GENIVI to establish Linux based open-source IVI platform, Android Auto of Google, Car Play of Apple, Window Embedded, Car2 of QNX, and Tizen IVI of Samsung are those companies. Therefore, the 3rd party companies have inconvenience having to configure their own services so that they will match with each platform. On the other hand, the services based on HTML 5 standard can reactively respond to hardware like screen size, unlike existing native services, and have advantage that they can promptly correct, distribute existing service so that they will meet the demand of consumers in rapidly changing market. Therefore, this study is going to design and implement In-Vehicle Infotainment system based on open-source and HTML 5 standard. So this study is going to provide the service which is independent in any in-vehicle OS platforms and can promptly deal with the demand of consumers through the design and implement of In-Vehicle Infotainment Application based on HTML 5 and open-source.

**Keywords:** In-Vehicle Infotainment, HTML5, Car, Tizen IVI

### 1. Introduction

In the past automobile was just a means of transportation. However, by recent fusion with IT making electronic equipment of automobile is in fast progress [1], thus the paradigm of automobile service is being changed. The combination of entertainment field like information field required for simple driving and road guide and audio, video account for a field of automobile IT, being called In Vehicle Infotainment, IVI in automobile. For example, it shows various development possibilities by combination with mobile, network, and with this growth prospects not only the field associated with automobile but also IT field participate in competition. This can have examples of cooperation among automobile camp and IT camp like Samsung-BMW, LG-Benz at world's biggest home appliances fair 'CES 2015' and the automobiles equipped with OS of Google and Apple.

---

<sup>5\*</sup> Corresponding author. Tel. : +82-10-9027-3148.  
Emailaddress: kjb123@ssu.ac.kr(Jong-Bae Kim).

However, IVI is one of still-to-be-developed fields, though automobile companies and global IT companies are in cooperation. As a reason for it, the diversity of OS could be mentioned. Unlike mobile OS market where Android of Google and IOS of Apple occupy the most, every automobile company in OS field does independent development or jumps into standard competition in order to occupy the market with severe competition. Like this the appearance of various OS has a merit to widen the margin of consumers' choice. But, the burden upon the expense and period required for development by development due to each different platform occurs to 3rd party companies. This is one of factors which prevent the building of an ecosystem in IVI market. To solve this environment composition, which can serve an independent service to platform, is required. Therefore, this study is going to design and realize IVI application which strengthened the connectivity with Smart Phone, Smart Watch having different OS, by using a web standard HTML 5 [3] that has a merit to drive in a multi-platform compared with native application development. This study is going to lead IVI ecosystem development through this, by providing the service independent to other platforms and which can positively deal with market change.

## **2. Related Studies**

### **2.1. Marketing Tendency of Connected Car**

Recently, the interest in a connected car is being increased along with interest in IoT. We can see this from the fact that the position of automobile, as a demand place for IT technology and a living space, got strengthened in 'CES 2015,' the biggest world home appliances fair. 2 out of 5 keynoters are CEOs of automobile industry (Benz and Ford), and the number of automobile companies like BMW, Volkswagens, and Toyota and the companies who display their automobile technology is in 420 plus, making automobile come to the fore in this exhibition. This study checked the possibility of change in automobile not just as a simple transportation but also a living space with automotive electronics and development of mobile network [4].

### **2.2. IVI with a Smart-Phone**

IVI-related companies suggest a vision with each different platform design. However, we should consider if IVI can be more expanded through smart phone or by composing ecosystem like smart phone. And, we cannot be sure about success due to insufficient market size or developer-induced policy, though ecosystem is also built in IVI market like smart phone market. Also, the replacement cycle of vehicles and mobile shows a big difference. Though smart phone is replaced in an average of 2 years, vehicles are replaced with 5~10 year cycle. This could cause problems not only in simple replacement of hardware but also in stability of system. Although there is a way to update IVI through USB or wireless communication network by distributing firmware via internet, there could be limitation in maintenance aspect when hardware or interface is changed. This way can be used only for limited usage like small scale patch or simple map update. It is possible to maintain its newest service by other ways like replacement of hardware module, but, costs for re-development of system can happen. Thus, related industry judges that they can maintain the system within replacement cycle of vehicles through link of connected service with a smart phone.

### 2.3. Smart-Phone and IVI Interlock Solutions

Link solution of IVI with vehicles can largely be divided into ‘individual App link’, ‘App combined platform’, and ‘Multi-platform’. First, in case of ‘individual App link’, it is to use smart phone App and vehicle-use App separately. Though it uses common API and appears to do same movement through Bluetooth communication, it could cause burden in aspect of development, maintenance as it is developed and used separately. Second, ‘App combined platform’ can be classified as cloud type and mirroring type. Cloud type can be used by combining the contents between a smart phone App and an IVI App through cloud API link. Therefore, difficulty in maintenance can be reduced. However, additional problems like communication status, communication costs can happen as it strongly depends on data communication. Mirroring type is a method to project smart phone App screen on IVI screen. Though it is available to apply it on various Apps, it has demerit that UX should depend on a smart phone. Third, ‘Multi-platform’ can be applied to multi-platform by using web standard HTML5. It is possible to work even in off-line status unavailable for communication, and working in App type and can be operated in the same way in various environments as a multi-platform [5].

**Table 1. Link Solution Type of IVI [5]**

Division		Details	Remarks
Individual	App Link	App API link Separately develop for each App Burden of maintenance cost	Use USB, Bluetooth Android AOAP UseApple iAP
Combined	Cloud base	Link Cloud API Provide combined contents Easy in maintenance Strongly depends on data communication	Aha Radio: 3 beside Honda Airbiquity: 2 beside Nissan Tweddle: 1 beside Toyota
	Mirroring base	Mobile phone screen mirroring Possible for applying various Apps Depends on UX phone (OEM-led X)	Started mirror link certification CarPlay MirrorLink : Toyota CarPlay: 11 beside Hyundai
Multi	HTML5	Apply multi-platform Possible to apply Download App and smart phone link (Though App link, multi-platform)	Possible to introduce also in mirror link

### 2.4 Collaboration of Automakers and IT Companies

Many complete vehicle companies are under way in IVI platform standardization work, but it is expected to take time for it to be changed in revolutionary way like smart phone market. Due to the reason like this complete vehicle companies are operating cooperation with IT companies [6].

**Table 2. Collaboration of Automakers and IT Companies [6]**

Automakers – IT Company Collaboration	Note
KIA – MS	UVO
KIA – UVIVELOX	Car Widget
HYUNDAI – MS	BlueLink
HYUNDAI&KIA – VODAFON	TM
HYUNDAI – NHN	TM
HUNDAI&KIA – INTEL	Infotainment
SK Telecom – RENAULT&SAMSUMG	MIV
FORD – MS	Sync
BMW – INTEL	Infotainment
BMW – AT&T	BMW Assist
TOYOTA – MS	Cloud Computing TM Multi-Platform
TOYOTA – INTEL	Multi System
TOYOTA – LG	Navigation
Tesla - NVIDIA	TM
AUDI – TELNET	UMTS/HSDPA Module
AUDI – NVIDIA	Tegra
BENZ - FACEBOOK	Facebook for TM

### 3. IVI (In-Vehicle Infotainment) Platform

There are much various platforms in In-Vehicle Infotainment market platform market, unlike two strong systems of IOS and Android OS in mobile platform market. This market evolves day by day with growth of Smart Phone. There are GENIVI, a non-profit organization who intends to make an open-source based platform of infotainment system an industrial standard, Android Auto of Google, CARPlay of Apple, CAR 2 of QNX, MirrorLink™ of CCC and TIZEN of Samsung.

#### 3.1. GENIVI

GENIVI was established in March 2009 in order to make open source-based vehicles multimedia standard SW platform, and IT companies including complete vehicle companies participate in it. They launched as a specialist group in each area and are deducting requirements fit to each market and operating standardization work. As they aim at open platform and confirm API Spec and check if there is an open source fit to this by testing, and if it is appropriate, they adopt this open source as GENEVI Compliance or operate self-development. GENEVI platform is based on an open source in which 80% of Kernel and basic platform is provided by Linux Foundation, and accounts for about 15% after correcting, supplementing so that other existing open source module fit to IVI. And as a specified function to vehicles, the module developed by GENEVI Alliance accounts for about 5%. In May 2016 about 130 companies in each field as members of GENIVI participate in it [7].

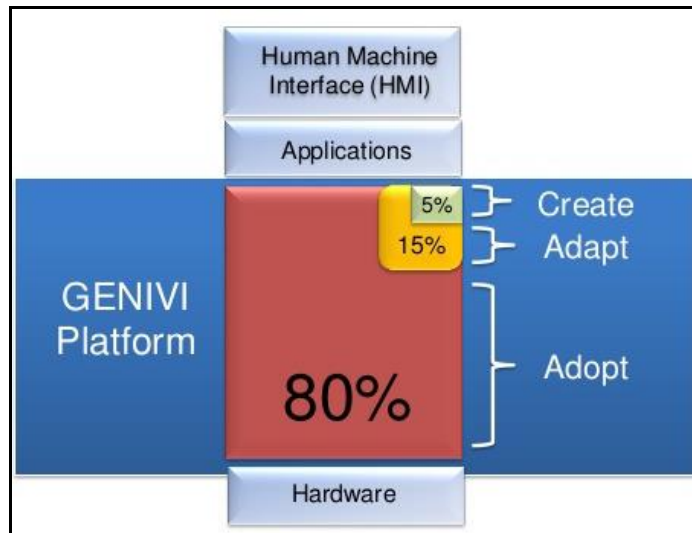


Figure 1. GENIVI Platform and Opensource

### 3.1. MirrorLink

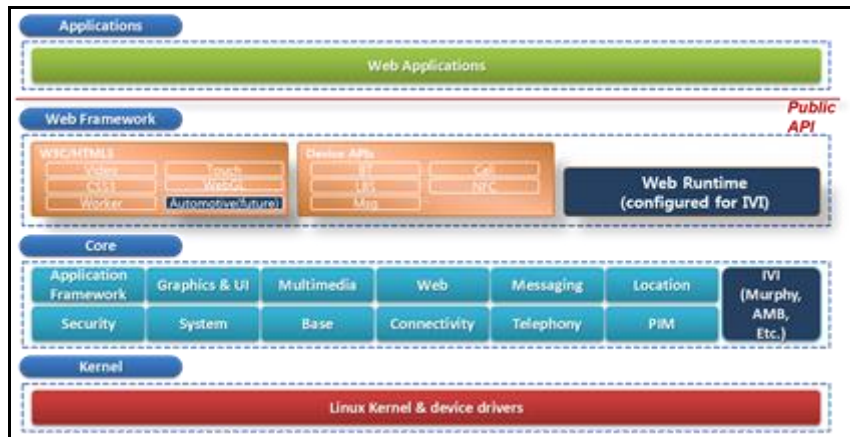
App is executed on a smart phone with standardized size suggested by CCC(Car Connectivity Consortium) and the contents follow the type that is outputted to a speaker and screen inside vehicle. Mirror link started from CE4A Terminal Mode Expert Group and was announced as a name MirrorLink™ as CCC was organized in 2011 after being announced as Terminal Mode 1.0. Smart phone plays a role of a server in MirrorLink™ and vehicle-use head unit a role of client of MirrorLink™. MirrorLink™ is used after a smart phone, a head unit, application receives CCC certification, and the certification of a smart phone and a head unit is determined according to how they will conform to the size of MirrorLink™ [9].



Figure 2. Test View of MirrorLink

### 3.3. Tizen IVI

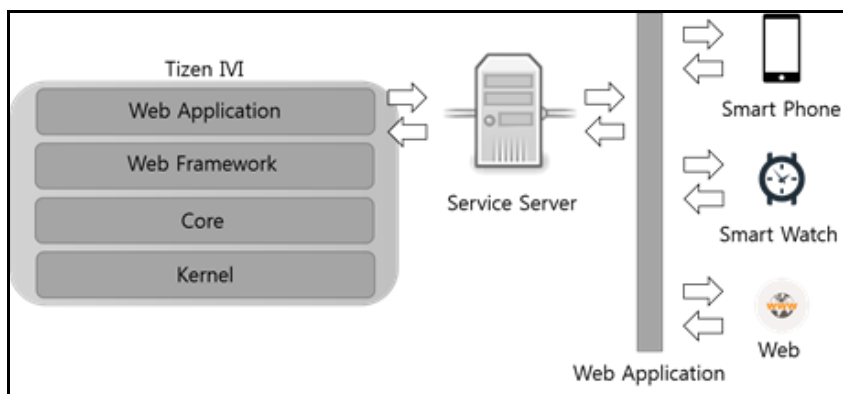
This study operates with TIZEN which actively supports W3C standard web technology like HTML5, and uses TIZEN IVI among them. Tizen Architecture is composed of Kernel, Core Framework, Web Framework, Native Framework, and Application Layer. However, unlike this Tizen IVI Architecture uses only a Web Framework, without providing Native Framework. The Figure 3 below is a Tizen IVI architecture [10].



**Figure 3. TizenIVI (In-Vehicle Infotainment) Profile Architecture**

#### 4. In-Vehicle Infotainment Application

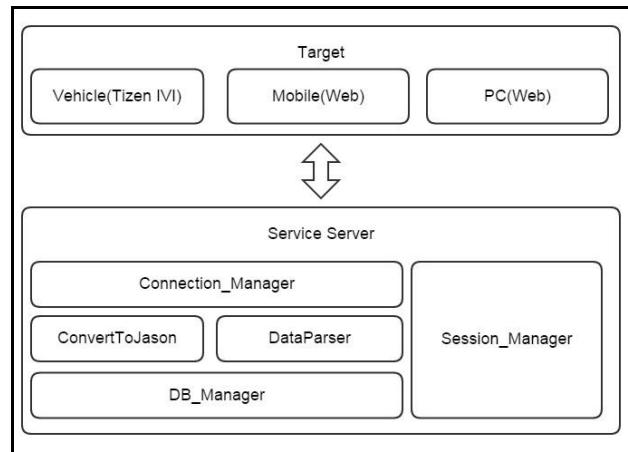
This study is going to produce an application which provides each other platform with service, by using web standard including HTML5. Figure 4 below falls on applications service block diagram. It composes automobile’s system based on TIZEN IVI, and is linked to a service server through a network. It does not realize the service by using actual automobile, but use a simulation provided by Tizen SDK. Service server collects user’s data through automobile, Smart Phone, Smart Watch, and web page, and provides this through these media again after processing the information collected.



**Figure 4. In-Vehicle Infotainment Application Service Configuration**

##### 4.1. System Configuration

Figure 5 below shows a diagram for a web-based IVI system. Vehicle (Tizen IVI), Mobile (Web) and PC (Web) are target devices. The service server consists of ‘Connection Manager’ which is responsible for connections between each device, ‘Session Manager’ which manages the identity of users, ‘Convert To Jason’ which converts the input information via ‘Connection Manager’ into ‘Jason Format’, ‘DB Manager’ which is responsible for database input and output of server and ‘Data Parser’ which converts the extracted information from DB into the desired information for each device.



**Figure 5. System Configuration**

#### 4.2. Demo Application

Two ways are considered to receive data from a vehicle which is one of the Target Devices. First, it is possible to extract information about several fragmentary items provided by a method of using OBD Simulator and OBD Scanner [11]. Secondly, there's a method of using Tizen IVI Simulator. Compared to the method of using OBD Scanner, this method makes it easier to get more diverse and detailed information.

Figure 6 below is an HW configuration for Vehicle Simulation. It consists of OBD Scanner for Bluetooth communication, ECUSim2000 as OBD Simulator and android device as an equipment for testing purposes.



**Figure 6. HW Configuration of Vehicle Simulation Using by OBD Scanner**

Figure 7 below shows a Vehicle Test which is using Auto Running of Tizen Web Simulator and this assumes a state in which the vehicle is traveling. By contrast with OBD Simulator Diverse, we can get more information such as Vehicle Speed, Engine Speed, Acceleration, Fuel, Steering Wheel Angle and *etc.* In this study, Demo Application is implemented by using the Tizen IVI Web Simulator.

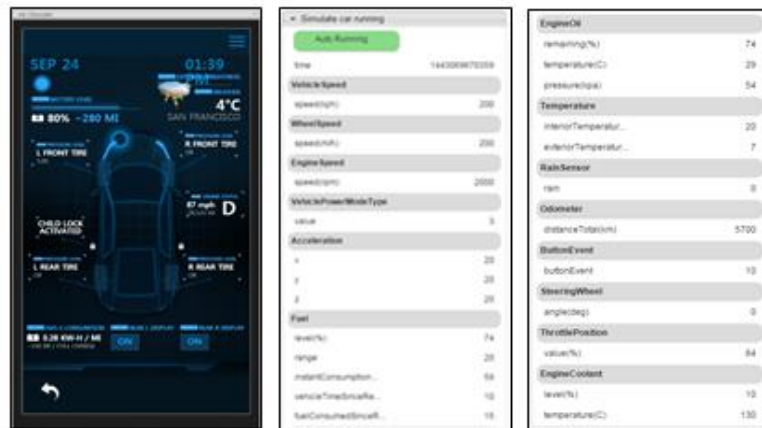


Figure 7. Test Running of Tizen IVI

Developers are highly accessible for Vehicle Information because Tizen IVI supports HTML5 which is web standard. For example, if you simply want to know about the speed of the vehicle at any timing point, you can use JavaScript code as seen in Figure 8.

```
navigator.vehicle.get("VehicleSpeed",onsuccess, onerror);

function onsuccess(value) {
    window.console.log(value.VehicleSpeed);
}

function onerror(value) {
    window.console.log(e.message);
}
```

Figure 8. Javascript Code of Tizen IVI to Get Vehicle Speed

If you want to know about the speed of the vehicle consistently, not just at a timing point like the above code, you can register as an Event Listener as seen in Figure 9 and then use it.

```
navigator.vehicle.addEventListener("VehicleSpeed", vehicleSpeedHandler, null);

function vehicleSpeedHandler(data) {
    window.console.log(data.VehicleSpeed + "kph");
}
```

Figure 9. Javascript Code of Tizen IVI to Add Event Listener

Via API we can also get a variety of information such as EngineSpeed, VehiclePowerMode, Acceleration, WheelBrake, LightState, Fuel and CruiseControlStatus.

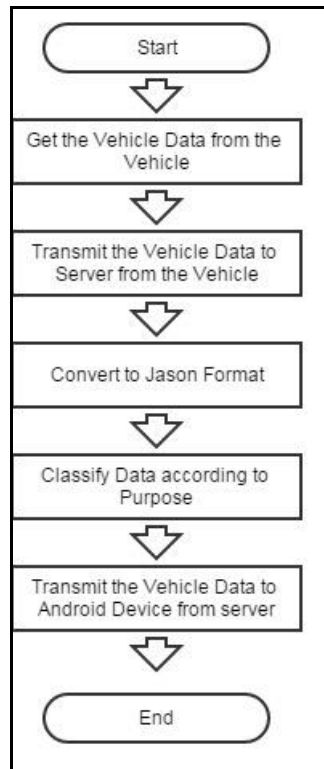




Figure 10. Vehicle Simulation Application

09:17:05.406	9116	9417	com.example.redis...	SendMsg	010D
09:17:05.406	9116	9417	com.example.redis...	BluetoothSoc...	writeNative
09:17:05.486	9116	9416	com.example.redis...	BluetoothSoc...	readNative
09:17:05.486	9116	9116	com.example.redis...	ReadMsg	41 0D 4E
09:17:05.486	9116	9116	com.example.redis...	Tmp_array[1]	0D 4F
09:17:05.486	9116	9116	com.example.redis...	SPEED	78 km/h
09:17:05.601	9116	9418	com.example.redis...	SaveInfo	1
09:17:05.611	9116	9417	com.example.redis...	SendMsg	010C
09:17:05.611	9116	9417	com.example.redis...	BluetoothSoc...	writeNative
09:17:05.646	9116	9418	com.example.redis...	SaveInfo	2
09:17:05.686	9116	9416	com.example.redis...	BluetoothSoc...	readNative
09:17:05.686	9116	9116	com.example.redis...	ReadMsg	41 0C 29 02
09:17:05.686	9116	9116	com.example.redis...	Tmp_array[1]	0C 29 02
09:17:05.691	9116	9116	com.example.redis...	RMP	2624 rpm
09:17:05.816	9116	9417	com.example.redis...	SendMsg	0105
09:17:05.816	9116	9417	com.example.redis...	BluetoothSoc...	writeNative
09:17:05.846	9116	9418	com.example.redis...	SaveInfo	1
09:17:05.886	9116	9416	com.example.redis...	BluetoothSoc...	readNative
09:17:05.896	9116	9116	com.example.redis...	ReadMsg	41 05 4C
09:17:05.901	9116	9418	com.example.redis...	SaveInfo	2
09:17:05.901	9116	9116	com.example.redis...	Tmp_array[1]	05 4C
09:17:05.906	9116	9116	com.example.redis...	COOLANT	36 °C
09:17:06.021	9116	9417	com.example.redis...	SendMsg	010D
09:17:06.021	9116	9417	com.example.redis...	BluetoothSoc...	writeNative
09:17:06.101	9116	9416	com.example.redis...	BluetoothSoc...	readNative
09:17:06.101	9116	9116	com.example.redis...	ReadMsg	41 0D 51
09:17:06.101	9116	9418	com.example.redis...	SaveInfo	1
09:17:06.116	9116	9116	com.example.redis...	Tmp_array[1]	0D 51
09:17:06.151	9116	9418	com.example.redis...	SaveInfo	2
09:17:06.181	9116	9116	com.example.redis...	Speeding!!!!	81
09:17:06.181	9116	9116	com.example.redis...	SPEED	81 km/h

Figure 11. Data Log



**Figure 12. Demo Application Process Flow Chart**

Figure10 above is an android application for testing purposes. It transmits the driving data of the virtual vehicle in Tizen IVI Web Simulator to the server and stores the converted format as Jason to server. In addition it requests required data by the android application on the server and extracts desired information from DB through Data Parser. Figure 11 above shows the output state for data received from the service server to the android application log. Figure 12 above is a Demo Application Process Flow Chart.

## 5. Discussion and Conclusions

Existing service companies have spent much time and expenses in development in order to provide service from various platforms. However, an independent service can be provided to platform by using web standard like HTML5. And, the service based on web standard has a merit to strengthen the connectivity among devices as it can be provided from various devices which can use web standard. This study is going to verify this through realization that used Tizen which strongly supports web standard. And, this study is going to develop it as a data management framework which can be operated in various platforms and OS based on this study.

## References

- [1] Hyundai Research Institute, Center Movement of the Automobile Industry Core Competency. VIP-Report Vol.14-22, vol.573, (2014).
- [2] KOCCA (Korea Creative Content Agency), Contents Industry Trend of USA, vol. 3, (2015).
- [3] A. Taicalsaari and T. Mikkonen, "The Web as an Application Platform: The Saga Continues", Proceedings of the EUROMICRO Conference on Software Engineering and Advanced Applications, Oulu, Finland, (2011).
- [4] KISA (Korea Internet & Security Agency) Future Internet Team, Trend Analysis of CES 2015, Internet & Security Focus, vol. 2, (2015).
- [5] Electronic Science, <http://www.elec4.co.kr/article/articleView.asp?idx=4996>, (2013).

- [6] EPNC, 2012.06, Special Feature, "The Current State of Automotive Infotainment Technology", pp. 18-25.
- [7] [www.genivi.org/](http://www.genivi.org/)
- [8] Session given at Connected Cars on November 13, 2012 by John Lehmann, Embedded Automotive Solutions, Mentor Graphics, and Board Member of the GENIVI Alliance, "The Importance of IVI, GENIVI, and Open Source".
- [9] <http://www.mirrorlink.com/>
- [10] Tizen IVI, <http://www.tizen.org>
- [11] J. G. Yoo, "Driving Analysis for Improving Driving Habits", Proceedings of the International Conference of Future Information & Communication Engineering, Kota Kinabalu, Malaysia, (2015).

## Authors



**Jae-Gon Yoo**, received his Bachelor's Degree of Computer Science in Soongsil University, Seoul(2015). And he is studying his Master's Degree in Software Engineering in the Graduate School of Software, Soongsil University, Seoul. His current research interests include Open Source Software and Software Engineering.



**Kyung Sik Jeon**, received his Bachelor's Degree in Management from Seoul Cyber University in Korea, (2013). And he is studying his Master's Degree in Software Engineering in Soongsil University, Seoul. Now he is CEO of INOVA C&C Co., Ltd. since 2014. His research interests focus on Digital Forensics.



**Yong-Ha Lee**, received Master's Degree in Construction Engineering in the Sangju University. He is studying his Christ and Authoritative Degree Combined with Software Engineering at a Graduated Soongsil University which is in Seoul. He worked in IT in The Cultural Heritage Administration from 1997 to 2014, and work to Regulatory Reform Information Department in Office for Government Policy coordination.



**Sangphil Kim**, received his Bachelor's Degree of Business Administration in Yonsei University (2011), Korea. And he is studying his Master's Degree in Software Engineering in the Graduate School of Software, Soongsil University, Korea. His current research interests include Open Source Software and Management of Technology.



**Jong-Bae Kim**, received his Bachelor's Degree of Business Administration in University of Seoul, Seoul(1995) and Master's Degree (2002), Doctor's Degree in Computer Science in Soongsil University, Seoul(2006). Now he is a professor in the Graduate School of Software, Soongsil University, Seoul, Korea. His research interests focus on Software Engineering, and Open Source Software.

