2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

GS1 Connected Car: An Integrated Vehicle Information Platform and Its Ecosystem for Connected Car Services based on GS1 Standards

Jiyong Han[1], Hyunseob Kim[1], Sehyeon Heo[1], Nakyung Lee[1], Daeyoun Kang[1], Byungsoo Oh[1]
KyungTaek Kim[2], Wondeuk Yoon[1], Jaewook Byun[1], and Daeyoung Kim[1]

***Abstract*— In recent years, the connected automotive industry has grown explosively. Various connected car services are emerging, such as remote vehicle diagnostics, driver's health monitoring, infotainment, and vehicle safety management. As a result, the number and type of vehicle data are increasing tremendously day by day. However, existing connected automotive solutions have a limitation in that each company manages its own closed data silos. This restricts connected car services from using data sources in various domains. Hence, we propose the GS1 Connected Car, an integrated vehicle information platform, and its ecosystem. We suggest GS1-based automotive data standards for not only in-vehicle data but also all the automotive-related data generated during the lifecycle of vehicles. We provide standardized data collection to EPCIS, the discovery of global automotive services using ONS, IoT Mash-up service between an in-car dashboard platform and IoT devices, video infotainment called GS1 video, and automotive lifecycle management application. We have implemented our platform in a real car by developing an Android-based vehicle dashboard, including service discovery, Mash-up services, and GS1 Video. Also, a mobile application for lifecycle management and Amazon skills for collecting driver's information are developed. Our demonstration and case study show the feasibility of the proposed platform, widening the scope of future connected car services.**

I. INTRODUCTION

The connected car industry has seen explosive growth recently. Connected cars refer to vehicles with connectivity to networks such as the Internet, and this feature enables various services for the better driving experience, for instance, remote vehicle diagnostics, driver's health monitoring, infotainment, and vehicle safety management. Mckinsey reported that the automotive revenue pool would grow and diversify with new services, creating a USD 1.5 trillion market in 2030 [1]. Also, it is estimated that the future autonomous car will record anything and everything, producing 4,000 gigabytes of data per day [2].

In the industry, many automotive and IT companies have proposed own connected car solutions competitively. However, solutions so far did not consider much of vehicles' lifecycle data and the data standardization. Global automotive companies such as BMW, Volvo, Toyota, and Hyundai have been developing own connected car services on their in-car dashboard platforms. Also, global IT companies such as Amazon, IBM, Google, and Apple have been creating Vehicle to Cloud (V2C) services and car infotainment which support smartphone connection by collaborating with automotive companies [3] [4] [5].

However, these services are limited in that each company operates different data silos and does not share data semantics. This restricts service providers from combining various data sources, not only in-vehicle data but also other diverse data such as driver's information, smart-city open data, and roadside unit data. In addition, it is difficult to organize massive car data, share data with different domains, build and discover global connected automotive services using aggregated data on existing platforms.

Recent research [6] proposed a common vehicle information model for vehicle data marketplaces. It allows brand-independent and generic data sets to be collected and utilized together. This model is advantageous to vehicle data collection and sales; however, they did not take the whole connected car ecosystem which includes global service discovery, connection with Internet of Things (IoT) devices, and infotainment, into consideration.

Although much work has been done to date, a more comprehensive information platform that contemplates connected car ecosystem needs to be further studied. Thus, we have defined four major challenges as below.

- There should be automotive lifecycle open data standards to collect and share data between different domains.
- Connected car users should be able to discover and access global automotive application services.
- Connected cars should be able to support connection to IoT devices.
- A variety of infotainment needs to be provided that considers the user experience and global business connections.

To aim this goal, we propose the GS1 Connected Car, GS1 (Global Standard #1) standards-based integrated vehicle information platform and its ecosystem which provides standardized data collection, global service discovery, IoT Mash-up services, business-connected infotainment, and car lifecycle management. GS1 is the international standardization organization [1] which provides EPCglobal standards for objects. The objects could be products, services, physical locations, people, documents, and others. First, we define

[1]Authors are with Auto-ID Labs and School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. (email: {jyhanzz, hskim411, crux042, nnlee, ikasty, bsoh, wdyoon, bjw0829, kimd}@kaist.ac.kr)
[2]KyungTaek Kim with Hyundai Autron, Seongnam-si, Gyeonggi-do, South Korea. (email: KyungTaek.Kim@hyundai-autron.com)
The authors would like to thank Hyundai Autron, Ericsson, KMIT, and ENTfactory for their supports.

[1]https://www.gs1.org/

the standardized vehicle data schemata of not only in-vehicle data but also all the car-related data generated during cars' lifetime, and we offer systematic data collection in Electronic Product Code Information Services (EPCIS) [7]. Second, our platform provides global connected car service discovery to search car-specific services based on users' environment setting such as countries and languages, leveraging Object Name Service (ONS) [8] infrastructure. Third, our IoT Mash-up service [9] enables various IoT devices to be merged with automotive dashboard platforms by retrieving device bundles from ONS automatically. Fourth, GS1 Video [10], an infotainment for the rear seats, provides viewers with information and services on video content. And finally, we offer a car lifecycle management application, GS1 Car Manager, which can collect and query car-related data in EPCIS.

We have demonstrated that the GS1 Connected Car could overcome challenges mentioned above by applying our platform in a real car, Grandeur HG (Azera) from Hyundai Motor Group. We have developed an Android-based automotive dashboard that supports standardized vehicle data collection and user interfaces for data display and service discovery. IoT Mash-up service was implemented to show the work with LG Smart Watch. We have also developed GS1 Video and GS1 Car Manager as well.

The rest of the paper is organized as follows: First we introduce the background in section II and propose the GS1 Connected Car ecosystem in section III. In section IV, we show our implementation on a real car and demonstrate it. In section V, we discuss four case studies, and finally, section VI concludes the paper.

II. Background

GS1, the international standardization organization, is working on the data standardization of various business fields, especially of retail and logistics. Recently, it is actively expanding its area to other businesses such as public transportation (e.g., railway visibility), healthcare, smart-farm, smart factory, and financial service. One of the standards they provide is Electronic Product Code (EPC) (e.g., RFID) for identifying objects, locations, people, etc. We exploited the EPCglobal standard to handle massive and various car data including in-vehicle data and other car-related data generated during cars' lifecycle. An EPC repository called EPCIS [7] was employed to store and share vehicle data between other domains in the connected car ecosystem. Also, we used ONS to globally discover connected car services of manufacturers, and the third party service providers.

1) GS1 Identification Keys: GS1 keys [11] are used to identify objects, services, physical locations, and others. The globally unique GS1 key is assigned to each object and links the actual object to the relevant information. Table I shows the list of GS1 identification keys. For example, we can use the Serialized Global Trade Item Number (SGTIN) with Vehicle Identification Number (VIN) for identifying a vehicle. This enables to trace the location or the history of the vehicle with given SGTIN. Automobile parts can be expressed in CPID. The location of repair shops, factories can also be indicated by Global Location Number (GLN).

2) EPCIS and Core Business Vocabulary: EPCIS is an event repository that collects and shares information about physical movement and status of objects. It returns the properties of objects based on *What, When, Where and Why* through standardized interfaces, capture, and query, which are used for data storing and accessing. EPCIS uses standardized formats and methods for data expression and access; therefore, it has strength regarding event data visibility and its sharing across enterprises. In our platform, we collect real-time car-related data and store them in EPCIS to share data with service providers.

The GS1 Core Business Vocabulary (CBV) is a standard [12] which describes the status of objects. It ensures a common understanding of data semantics between businesses, which increases interoperability. We have newly built user vocabularies for the connected car and its ecosystem. This standardizes vehicle event semantics so that service providers can access the various lifetime data of cars and also interpret events in the same semantic.

3) Object Name Service: ONS [8] is a service that retrieves object-related services and data using the GS1 identification key which is the id of the object. ONS is based on the Domain Name System (DNS) infrastructure to leverage existing standards. Therefore, record formats and resolving process follow the DNS standard. The ONS client queries the ONS infrastructure using the transformed FDQN (Fully Qualified Domain Name) from the object's GS1 identification key. The ONS infrastructure hierarchically searches ONS servers that know the list of data and services for objects. The ONS server returns Naming Authority Pointer (NAPTR) records which supports complex data encoding, such as a Uniform Resource Identifier (URI) and object descriptions.

The GS1 Connected Car platform provides integrated services leveraging GS1 ID Keys, EPCIS, CBV, and ONS. For instance, a car and an engine oil product would have GTIN *A* and *B*. When the oil change cycle expires, our platform automatically queries the ONS with GTIN to find services such as finding compatible oils for your car, moving to the cheapest oil shop or analyzing the performance of your old oil *B*. After oil change, CBV-formatted *engine oil change*

TABLE I: The GS1 Identification Keys

ID Key	Used to Identify
(S)GTIN	(serialized) Products and services
(S)GLN	(serialized) Parties and locations
SSCC	Logistics units
GRAI	Returnable assets
GIAI	Assets
GSRN	Service provider and recipient relationships
GDTI	Documents
GINC	Consignments
GSIN	Shipments
GCN	Coupons
CPID	Components and parts
GMN	Product model

event data is recorded in EPCIS with the GTIN A and B.

III. GS1 Connected Car Platform and its Ecosystem

A. Overview

To overcome the connected car challenges, we propose a comprehensive GS1 Connected Car ecosystem which provides automotive lifecycle open data standards, global automotive application service discovery, fully automated Mash-up with IoT devices, and user-interactive video infotainment to the business connection.

Fig. 1 shows the conceptual view of the GS1 Connected Car ecosystem from collecting vehicle data to accessing car-related services. First, the main feature of the GS1 Connected Car platform is connectivity to the outside of the vehicle. A vehicle gateway that can communicate through various protocols is installed in the car. Our prototype implementation mainly used LTE, Bluetooth, and Wi-Fi. All data transactions with the outside go through the gateway.

Second, vehicle event data collection is also one of the most important parts of the connected car ecosystem. Various events and related data occur from manufacturing to disposal of the car. We defined the form of data representation in the GS1 format and proposed a systematic collection in the EPCIS repository.

Third, we leverage ONS to discover and access a variety of connected car services. It provides up-to-date global services to users based on their environment setting such as countries and languages. The car requests the available service list to ONS; then, the hierarchically discovered ONS server returns the service list to the car which contains OEM service and the third-party services. Verified service providers can register their services such as remote diagnostics, car history management, location-based services, and navigation.

Fourth, existing IoT devices could be merged into the connected car ecosystem through our Mash-up service. It provides a fully automated interconnection between the automotive dashboard and IoT devices such as health-care products and AI speakers. Mash-up service scans available IoT devices, automatically downloads and installs device drivers from the address obtained through ONS.

Finally, we offer user-interactive video infotainment to the business connection such as e-commerce and finding travel information. Users can easily access related information using embedded GS1 keys while watching TV programs or movies in the car.

The remaining subsections explain the architecture of GS1 Connected Car Platform and cover more details.

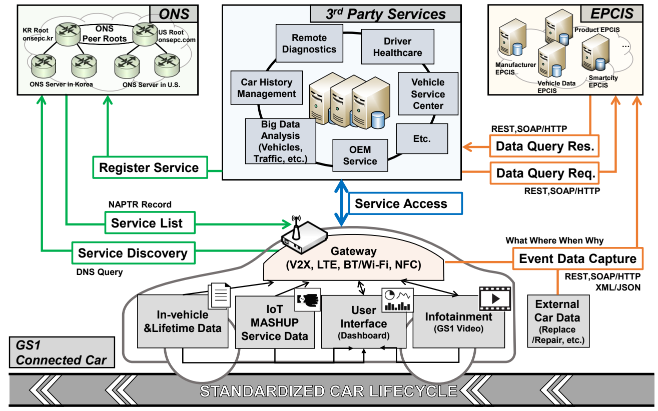


Fig. 1: The Conceptual View of the GS1 Connected Car Platform and Its Ecosystem

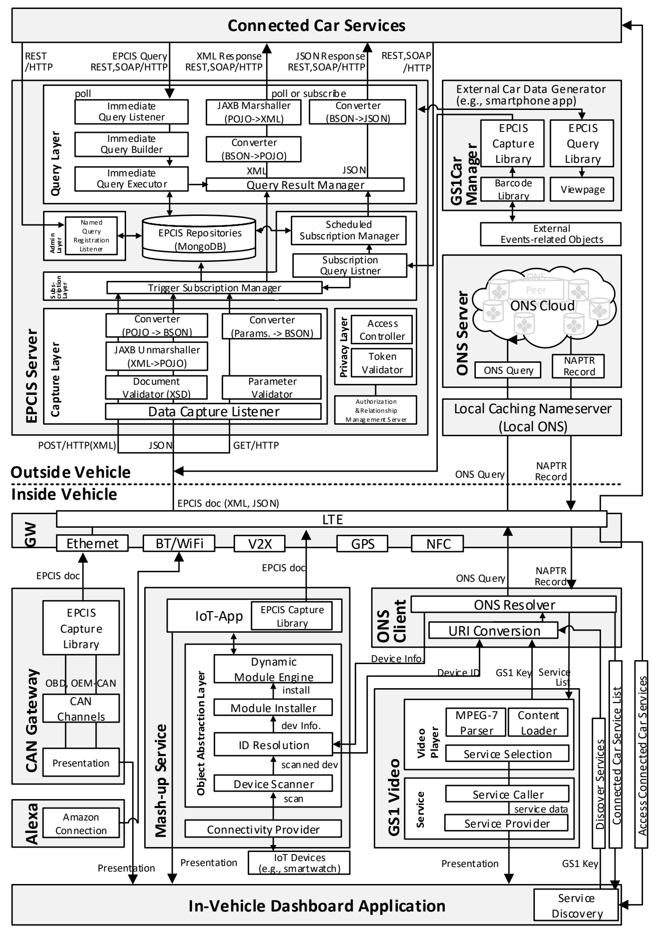


Fig. 2: The Architectural Overview of the GS1 Connected Car Platform

B. Architecture

Fig. 2 describes the architectural overview of the GS1 Connected Car Platform. We have two domains, *Inside Vehicle* and *Outside Vehicle*. First, based on our previous work [13] [14], we implemented an EPCIS server outside the vehicle. The server has backend repositories in the core and two main interfaces, data capture (i.e., collect) and data query interfaces. The data from vehicles are translated into EPCIS documents either XML or JSON format. Then, they go to EPCIS capture interface through the vehicle gateway, and finally stored in EPCIS repositories using standardized vocabularies. EPCIS supports various vehicle data from internal vehicle signal data such as OBD-II and OEM-CAN to external infrastructure data such as roadside unit, and also event data during a car's lifecycle such as part replacement, car repair, driver's identification, and healthcare information.

EPCIS-accessing connected car services can query data to EPCIS servers with SOAP/REST APIs. Note that the EPCIS server supports complex data queries with regex. For example, 'find all engine oil *Replacing* events of the last year when the price of engine oil was under $20'. Non EPCIS-accessing connected car services are also discoverable via ONS.

ONS cloud is located in *Outside Vehicle*. When a car requests a service discovery with its GTIN, the ONS client converts GTIN to an ONS query and hierarchically finds the ONS server which has the list of available services. Local caching name server, local ONS, is also used for caching ONS query results.

In the case of *Inside Vehicle*, the vehicle gateway manages all communications between the inside and the outside of the vehicle. CAN gateway collects in-vehicle data and translates into EPCIS document using the EPCIS capture library, then sends to the vehicle gateway. Mash-up service data is also translated and gathered through the gateway. Devices that require an Internet connection can also connect to the gateway.

The in-vehicle dashboard platform basically presents car-related information to users. It also provides the interfaces of service discovery and IoT Mash-up. Built-in Mash-up service enables various IoT devices to be used with the dashboard platform. It scans devices nearby, queries ONS using scanned device ID, and receives the device specification and the location of device drivers. Module installer installs the driver and activates the IoT device. We can also collect the data(e.g., driver's heart rate) from IoT devices into EPCIS server.

GS1 video can show the additional information while watching the video by using GS1 keys inside the video's metadata (MPEG7 XML). For example, when a viewer wants to get more information about a product on the screen, a list of related services appears on the side of the screen. He can then navigate to related services such as shopping, product review, and travel information.

We also offer a mobile application, called GS1 Car Manager, for car lifecycle management. When external vehicle events(e.g., replacement of consumables, car repair) occur, users can store events by scanning the barcode or manually inserting GS1 keys and vocabularies related to the event.

This whole architecture illustrates that the GS1 Connected Car platform has strength in the collection of various vehicle data, the discovery of global services, the connectivity with existing IoT devices, media infotainment, and finally car lifecycle management.

C. Automotive Lifecycle Open Data Standards

Fig. 3 shows the lifecycle of vehicles from production to disposal, expressed in business vocabularies (i.e., business steps). We aim to set a standard user vocabulary set for the vehicle industry, adopting the vocabulary defined in the GS1 CBV standard as much as possible and redefining the vocabulary that does not exist in the CBV standard. In the distribution process, cars go through events such

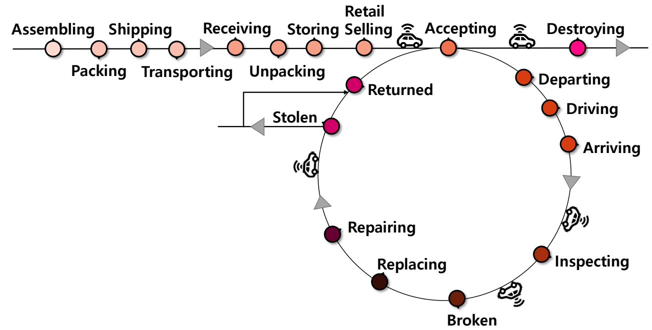


Fig. 3: The Lifecycle of Vehicles

as *Assembling, Packing, Shipping, Transporting, Receiving, Unpacking, Storing*, and *Retail Selling*. Regarding after-sale maintenance, events such as *Accepting, Departing, Driving, Arriving, Inspecting, Broken, Replacing, Repairing, Stolen*, and *Came back* occur. The lifecycle will continue even if the vehicle owner changes. When the car is discarded, the lifecycle of the car eventually ends with a *Destroying* event. The detailed definition of each event is described in Table II.

In the case of *Driving* event, data such as ECU status, OEM-CAN data, and driver's health information are created. In the case of *Replacing* event, data such as the GTIN of the car, the GTIN of the replaced product, the GLN of the repair shop, and the CPID of the replaced automobile parts

TABLE II: The Description of Car Lifecycle Events

Event	Description
Assembling	Combining one or more parts to make one complete car.
Packing	Putting a car in a large container for shipping.
Shipping	The overall process of moving from a facility to an area where it will await transport pick-up, loading and departing
Transporting	Process of moving a car from one location to another using a vehicle (e.g., a ship, a train, a lorry, an aircraft)
Receiving *	An event the car arrived at the warehouse
Unpacking	Removing a car from a larger container
Storing	Keeping a car as a stock in the warehouse
Retail Selling	Selling a car to a customer
Accepting	An event that set ownership of a car
Departing	An event where a car leaves a specific place to go to its destination
Driving *	An event where a car drives on the road
Arriving	An event where a car arrives at a specific destination
Inspecting	An event that checks whether the car itself has defects or not
Broken *	An event where a car is defective.
Replacing	An event where a part of car is substituted or exchanged for another one.
Repairing	An event where a malfunctioning part is repaired, without replacing it by a new one.
Stolen *	An event where a car is stolen from a stranger
Returned *	An event where a stolen car is taken back and returned to the owner.
Destroying	An event where a car is damaged and can no longer function as a car

Note: The events tagged with * are newly defined vocabularies in this paper, and the rest are GS1 standard CBVs.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema  targetNamespace="urn:autoidlabsk:epcisapp:car" xmlns:car="urn:autoidlabsk:epcisapp:car"
            xmlns:xs="http://www.w3.org/2001/XMLSchema" attributeFormDefault="unqualified"
            elementFormDefault="qualified" version="0.1">
    <xs:annotation><xs:documentation>
        <car:copyright>Copyright (C) 2018 AutoIDLabs, All Rights Reserved. </car:copyright>
        <car:disclaimer>AutoIDLabs</car:disclaimer>
        <car:specification>GS1 Car standard Version 0.1</car:specification></xs:documentation></xs:annotation>
    <xs:simpleType name="EngineRPM" id="1.0C">
    <xs:annotation><xs:documentation xml:lang="en">
      Engine RPM shall display revolutions per minute of the engine crankshaft.</xs:documentation>
    </xs:annotation><xs:restriction base="xs:int"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="1_0D_VehicleSpeed" id="1.0D">
    <xs:annotation><xs:documentation xml:lang="en">
      Engine RPM shall display revolutions per minute of the engine crankshaft.</xs:documentation>
    </xs:annotation><xs:restriction base="xs:char"></xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="1_0F_IntakeAirTemparature" id="1.0F">
    <xs:annotation><xs:documentation xml:lang="en">
      IAT shall display intake manifold air temperature. IAT may be obtained directly from a sensor,
      or may be inferred by the control strategy using other sensor inputs.</xs:documentation>
    </xs:annotation><xs:restriction base="xs:char"></xs:restriction>
    </xs:simpleType>
```

(a) OBD-II Data Schema

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE project>
<epcis:EPCISDocument
    xmlns:epcis="urn:epcglobal:epcis:xsd:1" xmlns:car="urn:autoidlabsk:epcisapp:car"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    schemaVersion="1.2" creationDate="2018-02-12T09:05:08.720-04:00">
    <EPCISBody><EventList><ObjectEvent>
        <eventTime>2018-02-12T05:10:12.000Z</eventTime> <!-- time -->
        <eventTimeZoneOffset>+09:00</eventTimeZoneOffset>
        <epcList>
            <epc>urn:epc:id:sgtin:8800026910115.KMHFG41EBCA166142</epc> <!-- vehicle id -->
        </epcList>
        <action>OBSERVE</action>
        <bizStep>urn:epcglobal:cbv:bizstep:replacing</bizStep> <!-- Replacing event was occured -->
        <disposition>damaged</disposition> <!-- option  -->
        <readPoint>
            <id>urn:epc:id:gln:8800026900092</id> <!-- location of uploading this document -->
        </readPoint>
        <bizLocation>
            <id>urn:epc:id:gln:8800026900092</id> <!-- locaion of replacing an engine oil -->
        </bizLocation>
        <extension></extension> <!-- option -->
        <car:vin xmlns:car="urn:autoidlabsk:epcisapp:car">KMHFG41EBCA166142</car:vin>
        <car:engine_oil xmlns:car="urn:autoidlabsk:epcisapp:car">8801470205832</car:engine_oil>
    </ObjectEvent></EventList></EPCISBody>
</epcis:EPCISDocument>
```

(b) EPCIS Document (XML ver.)

Fig. 4: The Examples of OBD-II Data Schema and EPCIS Document

are created. These data are collected and used by service providers.

However, storing data without any rules is not useful at all, since service providers cannot find and use data systematically. We therefore propose a new standard based on GS1 keys, CBVs, and EPCIS document formats for vehicle data. First, we defined standard data XSD schema. As for OBD-II, we followed the SAE standard specification. As for company-specific OEM-CAN data, we used the internal data of Grandeur HG. Fig. 4a shows the schema example of OBD-II. The XSD formatted schema contains the name and ID of data, description, data range restriction, and additional information. Full schemata can be found in our web page[2].

Using defined schemata, we create EPCIS documents to transfer data to EPCIS server. An EPCIS document includes the event-time and timezone, the identifier of an object, the type of action, current business step (CBV), object status, the read-point (location) of the event, and extensions(additional information) related to the event. Fig. 4b shows the EPCIS document example of an engine oil *replacing* event. SGTIN and VIN are used as identifiers for the vehicle, the business step is *Replacing*, the oil change reason is *damaged*, and also GLN is used to identify the location of a repair shop. The extension uses the traditional VIN and engine oil product's GTIN.

When an event occurs, the EPCIS capture application creates an event document such as Fig. 4b and stores it in EPCIS via capture layer of the EPCIS server. Once documents are saved, service providers can access data via EPCIS query layer. For example, they can obtain oil replacing event data of specific customers and develop services for them such as engine oil recommendation for the car model. This information can also be utilized to develop mass production strategies and increase marketing effectiveness, and so on.

[2]http://autoidlabsk.org/gs1cc/schemas

D. *Global Automotive Application Service*

In the connected car ecosystem, a variety of connected car services would be provided to users. Not only car manufacturers but also other IT companies, even personal could provide connected car applications. Naturally, the necessity of distributing and discovering global services has been arising. We leverage the ONS to provide a feature that can find globally available services of a car, supporting different languages, countries and car models. An ONS client, which is in the car dashboard application, can search car-related services with a given GS1 identification code and return the available service list to users. Also, connected car application distributors can easily register their applications in an ONS server and make them visible to users.

To do this, we assigned a GTIN code to our GS1 Connected Car model for GS1 identification. The ONS query processing is illustrated in Fig. 2. First, the dashboard platform requests a list of connected car services to the locally deployed ONS client with Application Unique String (AUS). AUS contains environmental information such as GTIN code, language setting, and country. The ONS client then converts AUS to an FQDN string and sends a DNS query to the local caching ONS server. If the local server does not have domain information, it recursively searches for answers based on the DNS infrastructure. Finally, ONS infrastructure returns a DNS NAPTR record containing a set of connected car services to the ONS client. The ONS client interprets the records and sends the list of services to the car dashboard. Through this process, users can access up-to-date car-related servics.

The service list includes the services type and related data. Service types could be car manuals, emergency road services, infotainment, mobile commerce, GS1 company prefixes, EPCIS, and more. Relevant data is presented in Uniform Resource Locators (URLs) and can be used in conjunction with regular expressions to provide optional services based on user's environment setting.

E. *Connecting the Car to Everything*

The GS1 Connected Car platform can connect external IoT devices that users own. This feature expands the range of available services with more flexibility and scalability. Fig. 2 shows the automated Mash-up sequence of IoT devices using the IoT-MAP platform, which is our previous work [9]. Firstly, IoT-MAP discovers an IoT device and its identification. The identification information can be a GS1 key or another unique identifier. Next, it identifies the class of the detected device and queries the ONS server with the identifier to retrieves the driver bundle URL. Lastly, IoT-MAP downloads and installs the driver that abstracts physical IoT devices to logical objects. This bundle is the aggregation of basic functions similar to the interface scheme of GATT or LWM2M. Note that this whole process is done automatically, creating an abstracted object that can be used by other applications. By abstracting physical devices, connected car service providers can focus only on the functionality of the device, not the additional attributes such as model number, manufacturer, and connectivity.

With our Mash-up service, cars can provide healthcare services using user-owned Smart Watches which provides heart rate and athletic activity data. IoT-Map lets the car dashboard show the healthcare information from Smart Watch in real-time. Not only Smart Watch but also other IoT devices that support heart rate measurement can be abstracted as heart rate providers for connected car services. In addition, user data from mashed-up IoT devices can be utilized to identify drivers and provide user-customized services such as vehicle lighting, music recommendations, and seat position adjustments.

F. Infotainment to World Business Connection

As autonomous driving technologies have been advancing, passengers would have more opportunity to interact with the automotive infotainment system. In such environment, the link between automotive infotainment and business becomes important. We suggest GS1 Video [10] service, our previous work, as a solution of creating the bidirectional channel between business and users.

The GS1 video player associates specific video areas with related data and services when video playback begins. For example, suppose a user is curious about a new smartphone in a movie. When he touches the smartphone on the screen, the video player shows new smartphone data and related services such as online shopping, finding the closest smartphone store, and more. This information is received by querying region-embedded GS1 keys (e.g., GTIN) to the ONS server as shown in Fig. 2. GS1 keys are inserted into specific areas of the MPEG-7 XML so that the player can read them. An annotation tool for media providers is also provided to mark GS1 keys in videos.

We only need embedded GS1 keys to support various business services related to objects, which is much advantageous than manually adding hard-coded information in each video scene. The received information is always up-to-date since it is from the ONS server maintained by manufacturers, shopping malls, and service providers. Besides, GS1 video is not limited to merchandise sales. For example, if GLN is annotated, it provides location-based services such as travel.

G. Car Lifecycle History Management

As shown in Fig. 3, cars undergo events from assembling to destroying. Managing events is crucial to increase the product visibility, analyze event-related data, and predict future events of vehicles.

Thus, we propose GS1 Car Manager, a mobile application for vehicle event data management. It fetches vehicle event data from public or private EPCIS servers of manufacturers, personal, and others, and then it provides the comprehensive view of a car to users. Users can also enter vehicle events and details into the application when certain types of event data are not automatically collected from vehicle platform. For example, when a driver changes engine oil by himself in a garage (*Replacing*) and wants to store this event, he can use the GS1 Car Manager's user interface for event data capturing.

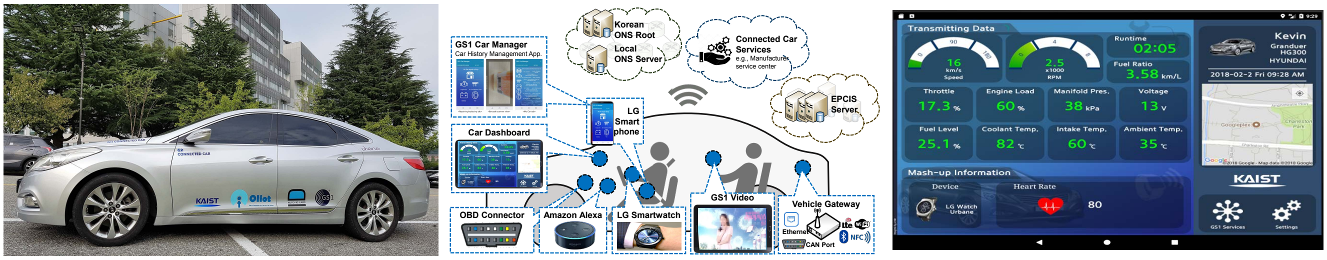
IV. Implementation

We have implemented the proposed GS1 Connected Car platform in a real car as demonstrated in Fig. 5a and named it Chérie. The car model is *Grandeur (Azera) HG* of Hyundai Motor Company. Fig. 5b shows installed devices inside the car and deployed servers. We mounted a Samsung Galaxy Tab S3 on the center console as a car's dashboard. Also, we deployed a Samsung Galaxy Book 12 on rear passenger seats for the GS1 video service. A standard OBD-II connector was attached to the OBD port under the steering wheel. Moreover, Amazon Echo Dot, a smart speaker, was installed under the center console for gathering driver's information. The driver wore an LG Smart Watch for the Mash-up service. An LG smartphone G6 was used for the GS1 Car Manager demonstration. Finally, we implemented an EPCIS server with web-based capture and query interfaces on the Amazon Web Service (AWS) and also Root ONS and Local ONS servers on the KT Ucloud.

Fig. 5c shows the in-car dashboard of the GS1 Connected Car. This Android-based dashboard application supports the integrated view of real-time in-vehicle data, service discovery (ONS), Mash-up services, user environment setting, and also maps. For the real-time in-vehicle data (i.e., OBD-II data) reception, we utilized the part of the OBD-II open-source [3]. The dashboard application establishes a connection with an OBD-II connector via Bluetooth and sends OBD-II data-request commands to the connector. After receiving raw OBD-II data, the application parses and interprets data. Finally, it prints out the readable data as shown in Fig. 6a. At the same time, these outputs plus time information are translated into XML EPCIS document, then transmitted to EPCIS capturing interface by POST/HTTP. Note that we also offer web-based capture and query interfaces for EPCIS. Fig. 6f represents the stored event data on the web query interface.

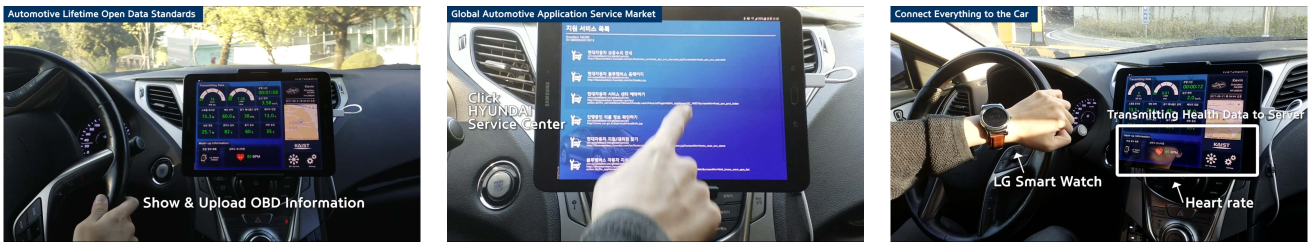
When a user clicks the *Service Search* button on the dashboard application, it requests and receives the list of available services from the ONS server as shown in Fig. 6b. For demonstration, we have registered several car services on root ONS such as finding the nearest service center, vehicle diagnosis Q&A, checking recall information, and searching car manuals. Note that these services are provided according to the user's environment setting such as country, language, car model. Since our environment setting was Korea, Korean, and *Hyundai Grandeur HG*, services which are only available in Korea, provided in Korean, and related to *Hyundai Grandeur HG* were listed.

[3]https://github.com/pires/android-obd-reader



(a) The GS1 Connected Car (b) Installed Devices and Servers (c) Automotive Dashboard Application

Fig. 5: The Implementation of the GS1 Connected Car Platform



(a) Collecting vehicle data (b) Discovered connected car services through ONS (c) IoT Mash-up service with a LG Smart Watch

(d) GS1 Video Demonstration with a TV commercial (e) The Use of Car History Management Application (f) The web-based display of the stored EPCIS Data

Fig. 6: The Demonstration of the GS1 Connected Car Platform with a Real Car. (Some parts in figures are blurred due to copyright issues)

Mash-up service is also combined with the in-car dashboard as demonstrated in Fig. 6c. We used an LG Smart Watch as an IoT device. When a user wears an LG Smart Watch and gets in the car, the in-car dashboard scans the Smart Watch and asks for the device driver bundle of it using an ONS query. After the automatic bundle installation, the user's real-time heart rate is displayed on the dashboard and stored to the EPCIS server.

For the demonstration of GS1 video, we used the Korean mineral water TV commercial as shown in Fig. 6d. When a user touches the water bottle of the screen, the product specification and the purchase link are popped up on the right side of the screen.

With Amazon Alexa, we also have implemented a *Skill*[4] which recognizes driver's boarding events. When a user gets in the car and tells *'(user's name) gets on the car'* to Echo Dot, Alexa recognizes the boarding and captures an event containing data such as user's name, GS1 GSRN key, timestamp, VIN, and current onboard status into EPCIS. In the case of getting off, the user says *'(user's name) gets off the car'* to Alexa, and then it also captures the information. We used the node.js to write execution functions.

The GS1 Car Manager in section III-G was implemented based on Android Nougat 7.1.1 and performed in the LG smartphone G6 as shown in Fig. 6e. My Car tab is the status view of car's parts. It shows which kinds of parts are applied to the car as a result of repairing or replacing events. Repairing/Replacing tab is an action page used for inserting vehicle-related information on corresponding events. An embedded barcode reader scans GS1 keys which represent product or location information related to events. Fig. 6e shows the operation of GS1 Car Manager when replacing oil at a repair shop. First, the application scans the GLN of the repair shop (or selects self-maintenance). Next, it scans the GTIN of a new engine oil product. Then, the application generates an EPCIS document of the *Replacing* event and captures it to the EPCIS server. My Car tab reflects the real-time changes to the car.

V. Case Study

In this section, we present four example cases within our GS1 Connected Car ecosystem. The following examples

[4]In Alexa, services created by third-party developers are called 'Skills'.

illustrate the expected benefits of the GS1 Connected Car:

CASE 1: Big Data Analysis of Vehicle Data The massive OEM-independent vehicle data are collected in EPCIS repositories in standardized forms, and verified organizations can access data for the Big data analysis. Car manufacturers can use drivers' driving patterns and styles to catch new demands of consumers, develop advanced manufacturing technologies, and adjust marketing strategies. Smart city and government can also retrieve statistical data to foster better transportation infrastructure and policies. Insurance companies could analyze customers' history, accident information, and the characteristics of car models to make better business models and broader insurance coverage.

CASE 2: Mash-up with Healthcare Devices - Emergency Smart devices could be utilized as if they were built-in accessories of cars. A driver gets in a car and built-in Amazon Alexa recognizes the driver's identity. The driver's Smart Watch is automatically connected to a car dashboard application. The real-time heart rate of the driver is monitored and showed on the dashboard. When the heart rate pattern suddenly changes into emergency condition, the ONS-discovered healthcare service immediately contacts a rescue center and sends the driver's information such as identification, real-time health status, location, chronic disease, etc. As a result, the driver can be provided quick initial treatment.

CASE 3: Reliable Car Lifecycle Record A man wants to buy a used car; however, he may not believe the inspection record of the car. In the GS1 Connected Car ecosystem, all data generated from a vehicle could be collected including past owners, total mileage, accident history, repairing and maintenance information, the replacement of consumables, and so on. The driver can access and manage the lifecycle of his car overall.

CASE 4: Prompt Information Provision while Watching Videos When the autonomous driving is commercialized, opportunities for accessing multimedia contents will increase, such as watching videos in automobiles. A driver is watching a TV program about traveling and wants to get more information about it. Then, web page links for additional details appear immediately on the side of the screen. The driver can get the information about the trip, make a reservation for accommodation, purchase related products, or search for local weather.

VI. Conclusion

The GS1 Connected Car and its ecosystem provide standardized data collection, global service discovery, the connection between cars and IoT devices, business-connected infotainment, and vehicles' lifecycle data management. We suggested vehicle data standard schemata and implemented EPCIS, ONS infrastructure, IoT Mash-up service, GS1 video, and the car lifecycle data management application upon the real car and the user's mobile, demonstrating its feasibility. This study has taken a step in the direction of defining the prototype of a connected car platform itself. We are also working on the integration with smart city platforms.

Acknowledgment

This paper was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2016-0-00067, Wise-IoT), International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the MSIP of Korea(2016K1A3A7A03952054), and the ITRC(Information Technology Research Center) support program (IITP-2018-2013-1-00877) supervised by the IITP and funded by the MSIP.

References

[1] McKinsey and Company, "Automotive revolution perspective towards 2030," 2016. [Online]. Available: https://link.springer.com/content/pdf/10.1365/s40112-016-1117-8.pdf

[2] Intel, "Data is the new oil in the future of automated driving," 2016. [Online]. Available: https://newsroom.intel.com/editorials/krzanich-the-future-of-automated-driving

[3] Strategy&, "Connected car report 2016: Opportunities, risk, and turmoil on the road to autonomous vehicles," 2016. [Online]. Available: https://www.strategyand.pwc.com/media/file/Connected-car-report-2016.pdf

[4] I. Global Industry Analysts, "Connected car solutions - global strategic business report," 2018. [Online]. Available: http://www.strategyr.com/Connected_Car_Solutions_Market_Report.asp

[5] S. Senior, C. Rec, H. Nishar, and I. Tom Horton @Amazon Web Services, "Aws connected vehicle solution," 2018. [Online]. Available: https://s3.amazonaws.com/solutions-reference/connected-vehicle-solution/latest/connected-vehicle-solution.pdf

[6] J. Pillmann, C. Wietfeld, A. Zarcula, T. Raugust, and D. C. Alonso, "Novel common vehicle information model (cvim) for future automotive vehicle big data marketplaces," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1910–1915.

[7] GS1, "EPC information services standard 1.2 (ratified standard, sep, 2016)," 2016. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/EPCIS-Standard-1.2-r-2016-09-29.pdf

[8] GS1, "Object name service 2.0.1 (ratified standard, january 31, 2013)," 2013. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/ons_2_0_1-standard-20130131.pdf

[9] S. Heo, S. Woo, J. Im, and D. Kim, "IoT-MAP: IoT mashup application platform for the flexible IoT ecosystem," in *2015 5th International Conference on the Internet of Things (IOT)*, Oct 2015, pp. 163–170. [Online]. Available: http://ieeexplore.ieee.org/document/7356561/

[10] B. Sohn, K. Kwon, and D. Kim, "GS1 Video: Open service system for video using MPEG 7 and GS1 standard," in *2017 IEEE International Conference on Edge Computing (EDGE)*, June 2017, pp. 174–181. [Online]. Available: http://ieeexplore.ieee.org/document/8029272/

[11] GS1, "Gs1 general specifications (release 18, ratified, jan 2018)," 2018. [Online]. Available: https://www.gs1.org/docs/barcodes/GS1_General_Specifications.pdf

[12] GS1, "Core business vocabulary standard 1.2.2(release 1.2, ratified, sep 2016)," 2016. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/CBV-Standard-1-2-2-r-2017-10-12.pdf

[13] J. Byun, S. Woo, Y. Tolcha, and D. Kim, "Oliot epcis: Engineering a web information system complying with epc information services standard towards the internet of things," *Computers in Industry*, vol. 94, pp. 82 – 97, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016636151730458X

[14] B. Sohn, S. Woo, J. Han, H. Cho, J. Byun, and D. Kim, "Gs1 connected car using epcis-ons system," in *2016 IEEE International Congress on Big Data (BigData Congress)*, June 2016, pp. 426–429. [Online]. Available: http://ieeexplore.ieee.org/document/7584972/