

자율주행차 데이터 모델 및 서비스 인터페이스 표준화 동향

이원석 웹(PG605) 의장, 한국전자통신연구원 책임연구원
신성필 한국전자통신연구원 선임연구원
차홍기 한국전자통신연구원 선임연구원

1. 머리말

최근 시장 전망에 따르면 현재 인터넷이 연결된 차량 수는 전 세계에 1억 대 정도로 추정되며, 2025년까지 4억 대까지 증가할 것으로 예상된다. 또한 각 차량이 하루에 약 25기가바이트(GB)의 데이터를 생성할 것으로 예상됨에 따라, 차량 데이터 모델 및 인터페이스 표준은 차량 데이터에 대한 통합 비용과 복잡성을 효율적으로 줄이기 위한 필수 표준으로 인식되고 있다. 이러한 변화를 인지한 W3C[1]와 COVESA(구 GENIVI Alliance)[2]는 2012년부터 차량이 인터넷에 연결되는 커넥티드카 환경에 필수적으로 요구되는 차량 데이터 모델과 차량 데이터에 차량 외부의 클라우드나 응용이 접근 가능하게 하는 표준 인터페이스 개발에 대한 협력을 시작했다. COVESA는 차량 관련 기업을 중심으로 모인 표준화 단체로 차량 데이터 포맷 표준인 VSS (Vehicle Signal Specification)[3] 개발을 담당한다. W3C는 COVESA의 데이터 포맷 표준을

기반으로 차량 데이터 접근을 위한 웹 기반 오픈 API 표준인 VISS(Vehicle Information Service Specification)[4] 개발을 담당한다. 본고에서는 COVESA VSS 및 W3C VISS 표준을 중심으로 자율주행차 데이터 모델 및 인터페이스 표준에 대한 소개, 자율주행차 데이터 모델 및 인터페이스 표준화 동향을 소개한다. 그리고 마지막으로 향후 계획에 대해 알아본다.

2. COVESA VSS (Vehicle Signal Specification) 표준화 동향

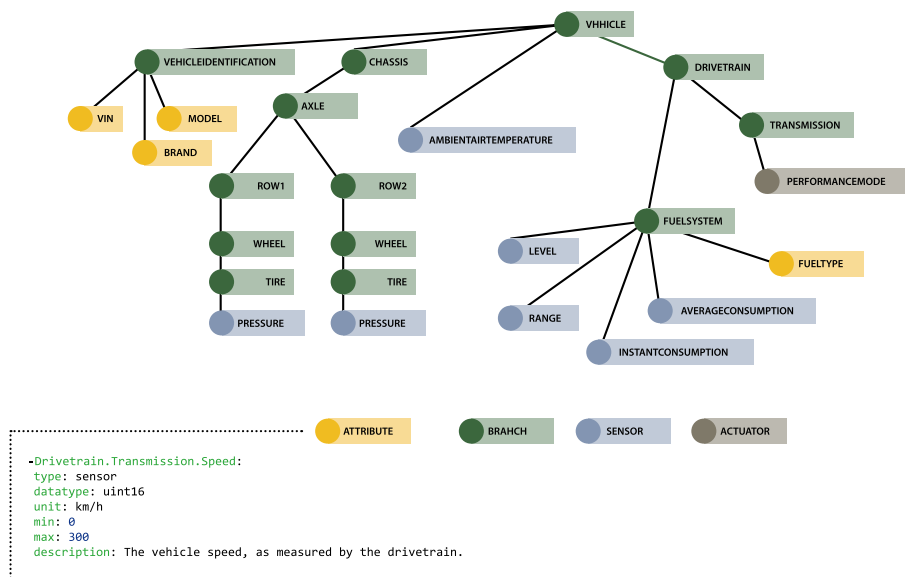
COVESA의 VSS는 차량 데이터를 표현하기 위한 표준 포맷이다. 개발자 친화적이며 확장 가능한 데이터 모델 및 카탈로그로, 업계에 필요한 다양한 지원 도구도 제공한다. VSS는 자율주행차 생태계의 다양한 기업들에게 차량 데이터에 대한 표준화된 표현 방법을 제공하여 상호운용성 지원 및 다양한 통합 작업을 개선하며, 시간과 비용을 획기적으로 절감할 수 있도록 지원한다.

VSS는 차량과 클라우드 간의 표준 인터페이스, 액세스 프로토콜 및 공통 차량 데이터 모델의 공동 개발 및 채택에 관한 글로벌 OEM의 논의 창구로 자리 잡은 Connected Vehicle Interface Initiative (CVII)[5]의 일환으로 개발된 산업 관련 데이터 표준이다. 최근 보고서에 따르면 커넥티드카는 3개 이상의 네트워크와 70-100개의 컨트롤러, 50-70개의 센서를 갖추고 있으며, 차량의 작동 상태를 나타내거나 차량에 관한 정보를 제공하기 위한 3,000개 이상의 개별 데이터를 생성한다. 평균 차량은 하루에 25GB의 데이터를 생성하며, 5G 또는 LTE 연결을 통해 강력한 엣지 컴퓨팅 장치로 클라우드와 데이터를 송수신할 수 있다.

VSS는 차량 신호에 대한 공통 이해를 위한 목적으로 개발한 데이터 모델 표준이다. 업계 내에서 개발자뿐만 아니라 비즈니스 관련 참여자도 VSS 모델을 통해 쉽게 자동차 데이터에 관한 공통 이해를 공유하여 의사결정에 도움이 되도록

록 하는 것이 본 표준의 궁극적인 목적이다. VSS는 이와 같은 목적을 달성하기 위해 도메인 분류(domain taxonomy) 체계 기반의 데이터 모델링 방법론을 채택하였다. 구조화된 데이터 모델을 가지며, 상위 도메인에서 다수의 하위 도메인으로 부모-자식 관계(parent-child relationship)를 형성하는 피쳐들로 데이터가 표현된다. 이렇게 트리 형태로 구조화된 데이터는 JSON (JavaScript Object Notation), YAML (Yet Another Markup Language)[6] 등 다양한 데이터 파일 형식으로 쉽게 저장될 수 있다는 장점이 있다[7]. 아래 [그림 1]은 VSS 데이터 모델 구조에 대한 예이다.

VSS는 차량 데이터 표현을 ‘트리’ 구조의 논리적 택소노미(공식적으로 Directed Acyclic Graph) 형태로 정의한다. 이는 주요 차량 구조(예: Vehicle 하위에 Vehicleidentification, Chasses, Fuelsystem, Drivetrain)가 트리의 상단에 있고 이를 구성하는 구성 요소가 자식 노



[그림 1] VSS 데이터 모델에 대한 구조 예

드로 정의된다. 트리의 각 자식 노드는 논리적 구성 요소로 더 분해되고 리프 노드(leaf nodes)에 도달할 때까지 프로세스가 반복된다. 리프 노드는 가지의 끝에 있는 노드로, 단일 신호 또는 데이터 속성 값을 나타내기 때문에 분해할 수 없다. 트리를 구성하는 노드의 종류는 다음과 같이 4가지가 있다. Attribute는 기본적인 고정 값을 갖는 노드로 차량 모델별로 고정되는 값인 VIN, 브랜드, 모델명 등이 예가 될 수 있다. Branch는 하위 노드를 갖는 분기 노드를 의미한다. Sensor는 읽기 전용으로 지속적으로 변경되는 값을 갖는 노드를 의미하며, 차량의 속도 값이나 연료 레벨 등이 예가 될 수 있다. 마지막으로 Actuator는 읽기와 쓰기가 가능한 노드로 제어를 위해 사

용되며 차문 윈도우 open/close, 트렁크 open/close 등이 될 수 있다. COVESA VSS 데이터 모델 내에서 신호는 점 표기법(dot notation)(예: drivetrain.fuelsystem.fueltype)을 사용하여 경로에 따라 이름이 지정된다.

VSS 표준은 YAML을 사용하여 정의하며 파일 확장자로 '.vspec'을 사용한다. 아래의 [그림 2]는 YAML로 정의된 VSS 표준의 예를 보여준다.

또한 COVESA는 VSS 기반의 데이터 모델을 업계에서 쉽게 활용할 수 있도록 다양한 변환 도구들을 제공하고 있다. 현재 제공되는 변환 틀은 아래 [그림 3]과 같이 활용되며 binary, CSV, JSON, YAML, DDS-IDL, FrancaIDL, GraphQL 포맷으로의 변환을 지원한다.

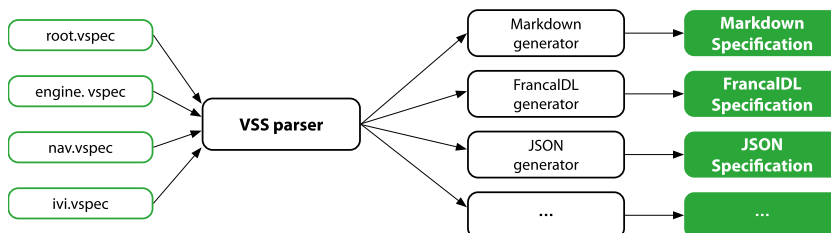
```
Vehicle:
  description: High-level vehicle data.
  type: branch
  uuid: ccc825f94139544dbb5f4bfd033bece6

Vehicle.ADAS:
  description: All Advanced Driver Assist Systems data.
  type: branch
  uuid: 14c2b2e1297b513197d320a5ce58f42e

Vehicle.ADAS.ABS:
  description: Antilock Braking System signals.
  type: branch
  uuid: 219270ef27c4531f874bbda63743b330

Vehicle.ADAS.ABS.IsActive:
  datatype: boolean
  description: Indicates if ABS is enabled. True = Enabled. False = Disabled.
  type: actuator
  uuid: 433b7039199357178688197d6e264725
```

[그림 2] YAML 기반 VSS 표준 정의 예



[그림 3] 현재 지원되는 VSS 변환 틀 및 활용

VSS 표준은 1년에 두 번 공식적으로 릴리즈되고 있으며, 2022년 8월 VSS 버전 3.0 릴리즈 후 2023년 공식적으로 릴리즈된 버전과 변경된 내용은 아래와 같다.

- VSS 버전 3.1 (2023년 2월 24일)[8]:

구조체 지원이 추가되었으나, 본 릴리스에서는 실험적인 지원으로 JSON 내보내기(exporter)에서만 지원된다. VSS 메타데이터 속성 중 거의 사용되지 않는 속성인 actuator 및 sensor를 삭제했다. 고유식별자(UUID)에 대한 명시적 생성을 위한 새로운 매개변수 --uuid를 추가했다. VSS 툴인 vspec2c 및 vspe2ocf를 더 이상 지원하지 않는다. 유닛 파일(unit file)을 지정하기 위한 -u 매개변수가 추가되었다. 일부 차량 데이터 정의가 리팩토링 되었다.

- VSS 버전 4.0 (2023년 5월 22일 릴리즈)[9]:

VSS 구문에서 구조체를 지원하지만 VSS 툴 4.0에서는 모든 내보내기(exporter) 기능을 아직 지원하지 못한다. 좌석(seat), 문(door), 미러(mirror) 및 기타 가지(branch)에 대한 인스턴스 처리가 변경되었다. VSS 메타데이터 속성 중 VSS 3.1에서 사용이 중단된 actuator 및 sensor가 VSS 구문에서 제거되었다. 세부적인 VSS 표준의 변경 사항으로는 주차 브레이크에 대한 자동 적용 추가, 전동 배터리 및 연료 시스템에 새로운 센서 추가 등을 들 수 있다. 창 위치와 차일드락(ChildLock)을 액추에이터로 변경하였고, 배터리 셀 전압 정보를 추가하였다. 다양한 차량 데이터에 누락된 단위를 추가하였고, IsScreenAlwaysOn, LastActionTime 및 DisplayOffTime 신호를 추가했다.

3. W3C VISS(Vehicle Information Service Specification) 표준화 동향

W3C는 2012년 COVESA와 협력하여 오토모티브 및 웹 플랫폼 BG를 설립하였고, 차량용 웹 응용을 위한 브라우저 기반의 차량정보 접근 API (Vehicle Information Access API)와 차량 데이터(Vehicle Data) 표준 초안 개발을 진행했다. 그 후 BG에서 개발된 표준 초안이 네이티브 클라이언트 지원이 어렵고, 표준 수정 시 지속적으로 브라우저 수정이 필요하다는 이슈가 제기되면서 2015년 설립된 오토모티브 WG은 이 표준안을 더 이상 개발하지 않기로 결정했다. 대신 오토모티브 WG은 2015년 하반기부터 새로운 표준으로 웹소켓 기반 차량정보 접근을 위한 인터페이스 표준인 VISS를 개발하기 시작하였다. VISS 표준은 차량 데이터 포맷으로 COVESA의 VSS 표준을 사용한다.

VISS는 자율주행차의 IVI (In-Vehicle Infotainment) 시스템에 탑재되는 차량 서버에 구현될 것으로 예상된다. 기본적으로 본 인터페이스는 차량 제조사에서 자사의 차량 클라우드 구축 시 자사에서 판매한 차량들의 데이터를 클라우드로 수집하기 위한 표준 인터페이스로 사용할 수 있다. 즉 차량이나 내외부의 응용, 서비스들이 본 표준 인터페이스를 이용하여 클라우드의 차량 데이터에 접근할 수 있다. 예를 들어 차량 엔진 오일이나 타이어 공기압 등 차량의 상태를 확인하고, 차량의 위치 파악 등이 가능해진다.

W3C는 2018년 VISS 표준을 후보권고안(CR, Candidate Recommendation)으로 승인하였으나 보안 및 프라이버시 고려사항 부분에 대한 이슈로 최종 표준으로 제정되는 것은 보류되었다. 이때 오토모티브 WG은 차기 버전의 VISS 표준

개발을 시작하기로 결정하고, 전송 프로토콜 확장에 대한 유연성 확보를 위해 VISS 버전2 Core 표준[10]과 VISS 버전2 Transport 표준[11]을 분리하여 개발하기로 결정하였다. 따라서 현재 VISS 표준은 크게 VISS version2-Core와 VISS version2-Transport 두 개의 표준으로 개발되고 있으며 이들 표준의 주요 내용 및 현황은 아래와 같다.

- VISS version2-Core 표준:

VISS version2-Core 표준은 주로 VSS 표준의 차량 데이터 모델을 기반으로 한 데이터 접근 방법, CRUDN 오퍼레이션에 대한 표준 인터페이스, 보안 고려사항, 차량 데이터의 효율적인 전송을 위한 필터 요청 기능, 데이터 접근 제어 모델을 정의한다. 차량 데이터 모델은 COVESA의 VSS 데이터 모델을 기반으로 정의하고 있으나, 본 VSS 데이터 모델에 독립적으로 설계되어 있으므로 다른 차량 데이터 모델을 적용하여 활용할 수도 있다. CRUDN 오퍼레이션 지원을 위한 인터페이스 정의는 전송 프로토콜에 독립적으로 설계되었으며 Read, Update, Subscribe, Unsubscribe, Subscription에 대한 메시지 포맷을 정의한다. 보안 고려사항은 TLS 기반의 전송 계층 보안, 데이터 접근 모델 기반의 데이터 보안, 프라이버시 고려사항을 기술하였다. 필터 요청은 클라이언트가 최대한 원하는 데이터만을 필터링하여 받을 수 있도록 지원하는 매우 중요한 기능이다. 현재까지 정의된 필터 요청 기능은 계층적인 VSS 데이터 모델에 대해 패스(path) 표현으로 원하는 차량 데이터를 요청할 수 있는 패스 필터(path filter), 현재부터 특정 과거 시점을 지정하여 차량 데이터를 요청할 수 있는 히스토리 필터(history filter), 특정 시간 주기를 지정하여 원하

는 차량 데이터를 요청할 수 있는 타임 기반 필터(time based filter), 차량 데이터의 값이 특정 범위에 위치한 경우에만 받을 수 있도록 요청하는 범위 필터(range filter), 최종 차량 데이터 값을 기준으로 지정한 고정 변화 값 이상으로 변동이 있을 때만 해당 데이터를 받을 수 있도록 요청하는 체인지 필터(change filter), 커브 로깅 알고리즘을 기반으로 차량 데이터를 받을 수 있도록 요청하는 커브 로깅 필터(Curve logging filter)가 있다. 또한 차량이 제공하는 차량 데이터는 제조사의 전략에 따라 다르기 때문에 차량이 어떤 데이터를 제공하는지 확인할 수 있도록 정적 메타데이터 필터(static metadata filter)와 동적 메타데이터 필터(dynamic metadata filter) 기능을 제공한다. 접근 제어 모델은 차량 데이터에 대한 접근 권한을 관리하는 모델로 기본 개념은 OAuth2.0을 기반으로 설계되었다.


- VISS version2-Transport 표준:

VISS version2-Transport 표준은 각각의 전송 프로토콜에 대한 세부 메시지 및 에러 코드를 포함한 전송 레이어의 상태 코드를 정의한다. VISS 버전1 표준에서는 웹소켓만 전송 프로토콜로 정의하여 지원하였으나, VISS version2-Transport 표준에서는 HTTPS 및 MQTT도 전송 프로토콜로 추가하여 이들 프로토콜 기반의 차량 데이터 전송도 지원하도록 확장하였다.

4. 맺음말

인터넷과 연결된 커넥티드카의 수는 2030년까지 빠르게 증가할 것으로 예상되며, 각 차량에서 하루에 약 25GB의 데이터가 발생될 것으로 예상된다. 이에 따라 차량 데이터 모델 및 인터페이스

표준은 차량 데이터에 대한 통합 비용과 복잡성을 효율적으로 줄이기 위한 필수 표준으로 인식되고 있다. W3C와 COVESA는 2012년부터 자율주행차에 필수적으로 요구되는 차량 데이터 모델과 차량 데이터에 차량 외부의 클라우드나 응용이 접근 가능하게 하는 표준 인터페이스 개발에 협력하여 업계에 영향력을 점차 넓히고 있다. COVESA는 올해 5월 VSS 버전 4.0을 릴리즈하였고, W3C 오토모티브 WG은 VISS 버전2 표준들을 개발 중이다. 이들 표준은 차량 클라우드에서 차량 데이터를 수집하는 구간에서, 그리

고 다양한 차량 응용 및 서비스들이 클라우드로부터 차량 데이터를 활용할 때 사용될 것으로 예상된다. 또한 W3C 오토모티브 WG은 차량 데이터를 다른 도메인과 융합하여 활용할 수 있도록 지원하기 위해 VSSo (VSS Ontology) 표준 개발 논의를 진행하고 있다. 본 표준은 향후 차량과 연계된 스마트 시티 등 다양한 다른 도메인과 연동한 서비스 구축을 위해 활용될 것으로 예상되며, 이를 통해 향후 자율주행차 생태계의 빠른 확산을 위한 기반 표준이 될 것으로 예상된다. 

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 정보통신방송표준개발지원사업의 일환으로 하였음 [2022-0-00443, 목적 기반 스마트 자율차량 및 서비스 간 상호 연동 표준 개발]

참고문헌

- [1] W3C, <https://www.w3.org/>
- [2] COVESA, <https://covesa.global/>
- [3] VSS(Vehicle Signal Specification), https://covesa.github.io/vehicle_signal_specification/
- [4] VISS(Vehicle Information Service Specification), <https://www.w3.org/TR/vehicle-information-service/>
- [5] CVII(Connected Vehicle Interface Initiative), <https://wiki.covesa.global/display/WIK4/Common+Vehicle+Interface+Initiative++Home>
- [6] YAML(YAML Ain't Markup Language), <https://en.wikipedia.org/wiki/YAML>
- [7] 신성필, 2023, "차량 데이터 상호운용을 위한 COVESA VSS 규격 표준화 동향", IITP 주간기술동향 2082호 pp.16-28
- [8] VSS v3.1, https://github.com/COVESA/vehicle_signal_specification/releases/tag/v3.1
- [9] VSS v4.0, https://github.com/COVESA/vehicle_signal_specification/releases/tag/v4.0
- [10] Ulf Bjorkengren, Isaac Agudo, 이원석(Wonsuk Lee), 2023, VISS version 2 - Core Editor's Draft, https://raw.githack.com/w3c/automotive/gh-pages/spec/VISSv2_Core.html
- [11] Ulf Bjorkengren, 이원석(Wonsuk Lee), 2023, VISS version 2-Transport Editor's Draft, https://raw.githack.com/w3c/automotive/gh-pages/spec/VISSv2_Transport.html