Toward a Common Vehicle Data Model

Cloud & Connected Services Workshop Session One

Benjamin Klotz, Gunnar Andersson | 29 October 2019

klotz@eurecom.fr
Toward a Common Vehicle Data Model

Motivation
Motivation

Manifold Air Pressure

< MAP >
Heterogeneous vehicle data

Signal name?
Units?
Timestamps?

- Temperature sensor
- Adaptive cruise control
- Front camera
- Radar
- Oil temperature sensor
- Tire pressure sensor
- Park assistant
- Blind spot detection
- Wheel speed sensor
- Steering angle sensor
- Vehicle height sensor

{"acceleratorPedal":{"position":"4095","ecoPosition":"3"},"brakeContact":"16","speedActual":"0"}, "timeStamp":"2018-01-10T17:01:27.297Z"}
Fragmented IoT standard ecosystem
Challenges

• Heterogeneous data
  • Sources: vehicles, road infrastructure, external APIs…
  • Different brands and models

• Hard to standardize bus signals (OBDII)

• Access control independence (data model vs data instances)
  • Security
  • Privacy
  • Different implementations

• In-vehicle signals vs backend APIs
Avoid the “xkcd 927 effect”
Toward a Common Vehicle Data Model
Gap analysis in today’s standards
Some major standardization initiatives

- ISO 20078 Extended Vehicle
- W3C Vehicle Information Server
- SensorIS
  - Android Auto Vehicle Interface (Vehicle HAL)
  - AutoMat Common Vehicle Information Model
  - Car Connectivity Consortium Car Data
  - IoT initiatives…
ISO Extended Vehicle (ISO 20078)

Motivation
- Increasing demand from 3rd parties to access vehicle data and functionality
- OEMs already equipped vehicles with telematics units and IT-infrastructure to handle connectivity
- Need to define a design and requirements to ensure that security, safety and data privacy (best practices, common methods)

Data model
- For 3rd parties to implement
- RESTful with JSON or XML schema with requirements on several aspects:
  - URI definition,
  - error handling,
  - Naming,
  - interaction pattern

Stakeholders
European OEMs contributing

Metadata
Policies: requirements for 3rd parties on data modeling good practices (e.g. URI use)
SensorIS

Motivation
• Enable broad access, delivery and processing of vehicle sensor data
• Enable easy exchange of vehicle sensor data between all players
• Enable enriched location-based services
• Drive global growth in this field

Data model
• Data messages in categories (which you can create)
• Identifies of submitter, session, message, vehicle fleet, vehicle, and driver
• Developed in google Protocolbuf library

Stakeholders

Metadata
• Units explicitly defined (e.g. “deg_c” for Celsius degrees)
• Policy for category extension to be compatible

And many more
Motivation

• Develop service specifications for exposing vehicle data and other information around vehicle centric functions.

• Not define or mandate implementation details including vehicle, network or sensor protocols

Data model

• Vehicle Signal Specification (VSS) as the per default model

• Alternative data models possible

Metadata

Vehicle Signal Specification (VSS):

• Extension mechanism

• Modeling best practices for signals and attributes
Toward a Common Vehicle Data Model
Vehicle Signal Specification @ GENIVI
Vehicle signal specification (VSS)

### Signal/Attribute
- Body
- ADAS
- Cabin
- Chassis
- Drivetrain
- OBD
- Vehicle

### Attribute
- Weight
- Raindetection

**Example Attributes:**
- **Weight**
- **Raindetection**

### Type
- UInt8

### Unit
- percent

### Description
- “…”

### Value
- restriction or free

### Signal

**Examples:**
- **Gearbox-sensed speed:** Drivetrain.Transmission.Speed
- **Engine speed:** Drivetrain.Engine.Speed
- **GPS-sensed speed:** Cabin.Infotainment.Speed
- **Left door lock:** Body.Row1.Door.Left.IsLocked
- **Right mirror tilt:** Cabin.Mirror.Right.Tilt

**Figure:**
- 451 branches
- 1103 leaves:
  - 43 attributes
  - 1060 signals: including
    - (700 seat-related),
    - 268 with unit
Generation and extensions

- root.vspec
- engine.vspec
- nav.vspec
- ivi.vspec

VSS parser

- Markdown generator
- FrancaIDL generator
- JSON generator
- VSSo generator

- Markdown Specification
- FrancaIDL Specification
- JSON Specification
- Ontological specification

#include vss_23.vspec
- Private.OEM_X.AntiGravity.Power: ...
- Private.OEM_X.TTeleport.TargetLoc: ...
...

#include oem_x_ proprietary.vspec
VSSo: VSS ontology [1]

Graph representation of formal models of:
- Vehicles and their branches
- Sensors, actuators, signals and attributes

SOSA pattern [2]:
- Sensor,
- Observation,
- Sample,
- Actuator

[2] https://www.w3.org/TR/vocab-ssn/
VSS 2.0

• A unified tree combining:
  • Static attributes
  • Sensors
  • Actuators

• Simpler position models:
  • Observe wheels in Row[1,2]
  • Get window position in [LEFT,RIGHT]

• Rbranch:
  • Resource branch
  • Adapted for collections
Toward a Common Vehicle Data Model

Discussion
Which data models are missing?
In standards and this gap analysis
Which access control solutions?
Implying technical requirements
For instance, which signal unit specifications? Restricted to one, multiple or open
What parts of the VSS-based ecosystem should be based on a standard database of named signal?

A “core” specification, in opposition to private extensions
How large parts are proprietary extensions?

To VSS or an equivalent specification
Which policies for future-proofing standards?

Scalability, flexibility, future needs…
How the choice of technical specification can affect the result?
Performance, feasibility
Thank you!

Visit GENIVI:
http://www.genivi.org
http://projects.genivi.org

Contact us:
help@genivi.org
Klotz@eurecom.fr

Next session at 13:45

“The Value of Vehicle Data to Enterprises”