# ASAM Open Simulation Interface

## Currently Planned Features and Enhancements

**Pierre R. Mai**
**PMSF IT Consulting**

2020-02-21
Höhenkirchen,
Germany

ASAM
Association for Standardization of
Automation and Measuring Systems

# Who Am I?

**Pierre R. Mai, PMSF IT Consulting**
pmai@pmsf.de

- **OSI CCB Member**
- **Interim ASAM Simulation Domain Coordinator**

- **MAP FMI Advisory Board Member**
- **MAP SSP Founding Member**

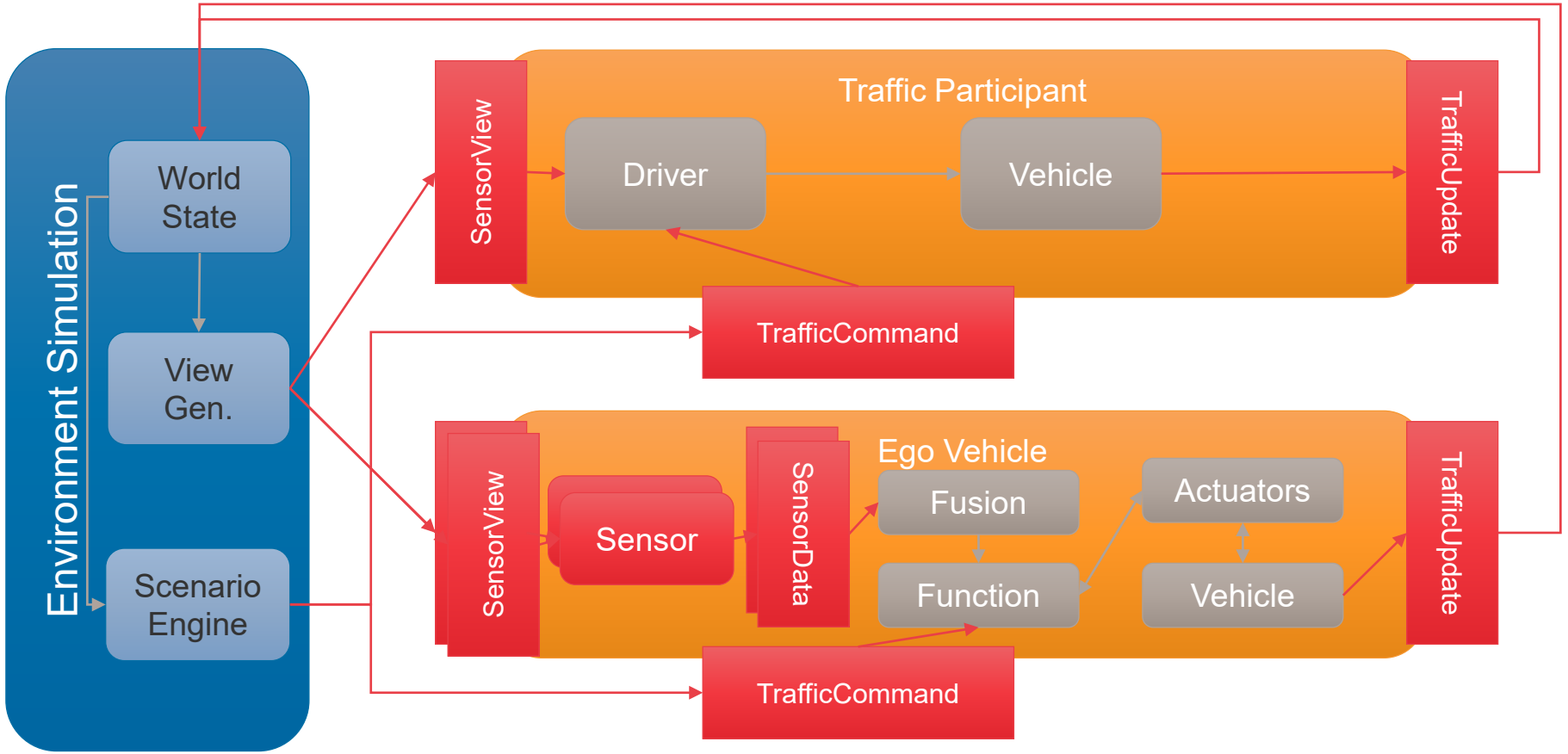
# Topics of currently planned enhancements to OSI/OSMP

| | |
|---|---|
| 1 | Traffic Participant, Enhanced Vehicle Data |
| 2 | Enhancements for Physical Sensor Modeling |
| 3 | Harmonization with ISO 23150 |
| 4 | Harmonization with ASAM OpenX |
| 5 | Performance and Encodings |

# Traffic Participant, Vehicle Data

# Traffic Participant

SETLevel4To5 Project Input



# Traffic Participant

- A **Traffic Participant** is a new OSMP Model Type:
  - Inputs:
    - 1..n OSI::SensorViews
    - 0..1 OSI::TrafficCommand
  - Outputs:
    - 1 OSI::TrafficUpdate

- Covers both **traffic agents** and **EGO vehicles**:
  - Traffic agents will contain driver/vehicle models as suitable for task
  - EGO vehicles will contain more internal structure:
    - Standardized sensor models (OSI::SensorView -> OSI::SensorData)
    - HAD functions (fusion/perception, decision making, planning, etc.)
    - Actuator and internal sensor models
    - Vehicle dynamics

# Vehicle Data Enhancements

- Enhancement of MovingObject Vehicle attributes:
  - Externally visible vehicle attributes, e.g.
    - Position and orientation of wheels/wheelcarriers,
    - Wheel speeds
    - …
- In Vehicle Data attributes:
  - Internally available in vehicle data, e.g.
    - Engine RPM,
    - Bus voltage
- …

# Physical Sensor Modeling

# Enhancements for Physical Sensor Modeling

## Current Status

- Full support for phenomenological sensor modeling using object-list level ground truth input
- Initial support for physical sensor modeling:
  - RadarSensorView
  - LidarSensorView
  - CameraSensorView

## Future Enhancements

- Add deeper support for physical sensor modeling, e.g.
  - More
  - Fine-tune definitions of provided data

# Harmonization with ISO 23150

# Harmonization with ISO 23150

ISO/CD 23150: Road vehicles – Data communication between sensors and data fusion unit for automated driving functions – Logical Interface

## Current Status

- ISO 23150 is currently in Committee Draft Stage (Stage 30.60)
- OSI SensorData is partially synchronized with earlier draft versions of ISO 23150

## Future Enhancements

- Goal is full harmonization with ISO 23150 CD and future DIS and IS releases

# Harmonization with ASAM OpenX

# Harmonization with ASAM OpenX Standards

## Current Status

- OSI takes OpenDRIVE road network data into account, but abstracted for real-time communication usage
- No full alignment on various attributes, e.g. traffic signs, traffic signals, …

## Future Enhancements

- **Near Term**
  - Align road network data with ASAM OpenDRIVE 1.6 where appropriate
  - Align OSI::TrafficCommand with ASAM OpenSCENARIO 1.0 release

- **Medium Term**
  - Based on common ontology across OpenSCENARIO, OpenDRIVE and OSI, align domain models
  - Use common IDL across standard specifications (potentially based on OpenSCENARIO 2.0 language)

# Performance and Encodings

# Performance and Encodings

IDLs and Encodings

## Current Status

- OSI uses subset of Google Protocol Buffers Proto2 IDL
- Support for Proto3 conversion, with on-the-wire compatiblity

- Bad interaction between C++ unified namespaces on Linux and Google Protocol Buffers when used with FMI
- Performance characteristics of Protocol Buffers not well aligned with fast sensor simulation

## Future Enhancements

- **Near Term**
  - Add support for Google Flatbuffer encoding based on Proto2 IDL

- **Medium Term**
  - Examine support for other encodings
  - Switch to common IDL with other ASAM OpenX standards

# Protocol Buffers Performance Characteristics

- **Google Protocol Buffers:**
  - Encoding designed for intra- and inter data-center request/response communication
    => optimized for smaller size vs. encoding speed/complexity
    (encoding is data-dependent, producing frequent branch prediction misses, no fixed data layout)
  - No in-place data access (requires full decoding prior to data access)
  - No in-place data mutation (requires full re-encoding even for minor changes)

- **Compared e.g. to Google Flatbuffers:**
  - Support for in-place data access
  - Support for in-place data mutation
  - Very fast encoding/decoding performance

# Protocol Buffers Performance Characteristics

- Performance comparison:

| | FlatBuffers (binary) | Protocol Buffers LITE | Rapid JSON | FlatBuffers (JSON) | pugixml | Raw structs |
|---|---|---|---|---|---|---|
| Decode + Traverse + Dealloc (1 million times, seconds) | 0.08 | 302 | 583 | 105 | 196 | 0.02 |
| Decode / Traverse / Dealloc (breakdown) | 0 / 0.08 / 0 | 220 / 0.15 / 81 | 294 / 0.9 / 287 | 70 / 0.08 / 35 | 41 / 3.9 / 150 | 0 / 0.02 / 0 |
| Encode (1 million times, seconds) | 3.2 | 185 | 650 | 169 | 273 | 0.15 |
| Wire format size (normal / zlib, bytes) | 344 / 220 | 228 / 174 | 1475 / 322 | 1029 / 298 | 1137 / 341 | 312 / 187 |
| Memory needed to store decoded wire (bytes / blocks) | 0 / 0 | 760 / 20 | 65689 / 4 | 328 / 1 | 34194 / 3 | 0 / 0 |
| Transient memory allocated during decode (KB) | 0 | 1 | 131 | 4 | 34 | 0 |
| Generated source code size (KB) | 4 | 61 | 0 | 4 | 0 | 0 |
| Field access in handwritten traversal code | typed accessors | typed accessors | manual error checking | typed accessors | manual error checking | typed but no safety |
| Library source code (KB) | 15 | some subset of 3800 | 87 | 43 | 327 | 0 |

Quelle: FPL https://google.github.io/flatbuffers/flatbuffers_benchmarks.html

Pierre R. Mai
Owner / Director, PMSF IT Consulting

Phone: +49 8161 976 96 - 11
Email: pmai@pmsf.de

ASAM