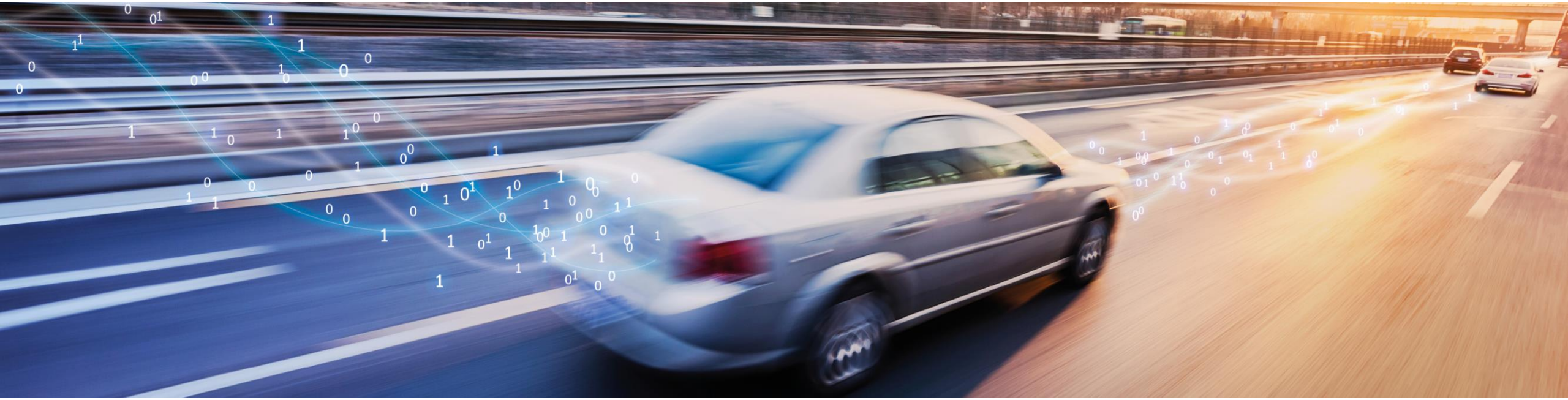
OSI 4.0

Road Model, Performance & Packaging

Pierre R. Mai

PMSF IT Consulting

2022-01-18
Teams



OSI 4.0: Road Model

Goals and Status

Status Quo

- Road model well-suited for phen. sensor models
- Missing details for road surface/shape
- Not a good match for traffic participant models
- Some alignment issues with OpenDRIVE

Goals

- Support traffic participants with logical information:
  - Possible paths across intersections
  - Multiple logical lanes on a single physical road/lane
  - Traffic rule information
- Add approximate road surface shape information
- Better alignment with OpenDRIVE
- Fix currently underdefined parts (e.g. lane boundary height/width orientation)

Current Status

- Road Model discussion group established
- Multiple approaches have been examined
- Current set of proposals is coming together
- Points to additional layers of information to be added:
  - LogicalLane information
  - Surface lines or other proposal for approximate road surface description
  - More clarifications on OpenDRIVE -> OSI mapping
- Potentially many can be added as minor/incremental changes
- However might still need/want incompatible changes for OSI 4.0.0, e.g. for cleanup.

OSI 4.0: Performance & Packaging

Goals and Status

Status Quo

- OSI uses ProtoBuf for encoding layer

- Amount of transmitted data is increasing with:
  - Increased level of detail
  - Additional information (e.g. for new model kinds)
  - Physical/Reflection-based sensor models
  - Larger use cases

- ProtoBuf encoding trades higher encoding overhead (especially on modern CPUs) for minimizing data sizes (original use-case: inter-datacenter comm).
- OSI usage is mostly single system image, which is latency/cpu-limited, not necessarily bandwith-limited.
- Better fitting encoding, like FlatBuffers would reduce overheads, simplify transfer/in-place use for OSI.

Goals

- Enable use of OSI data definitions with FlatBuffers
- Determine actual performance benefits
- Determine effort to port models/implementations
- Decide on main encoding for OSI 4.0.0:
  Switch to FlatBuffers, or stay with ProtoBuf

Current Status

- Initial port of OSI to FlatBuffers in OSI 3.4.0 done
- Current experience with reflection-based sensor models indicates we are running into performance limits with ProtoBuf.
- Workpackage to port some models and benchmark defined and service provider selected.
- Kick-Off of WP and community discussion @today!

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
  - Trade-Off: More involved API to support the in-place and iterative encoding vs. ProtoBuf API

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

Thank you for your attention!

Pierre R. Mai
PMSF IT Consulting

Phone: +49-8161-97696-11
Email: pmai@pmsf.de