OpenODD WP02 – Specification/Format

ASAM webinar

Dr Xizhe Zhang (Jason)

10 June 2021

ASAM Association for Standardization of Automation and Measuring Systems

Content

- Goal for WP02
- Why we need an ODD format
- Requirement clusters
- Example using the conditional statement cluster

Goal for WP02

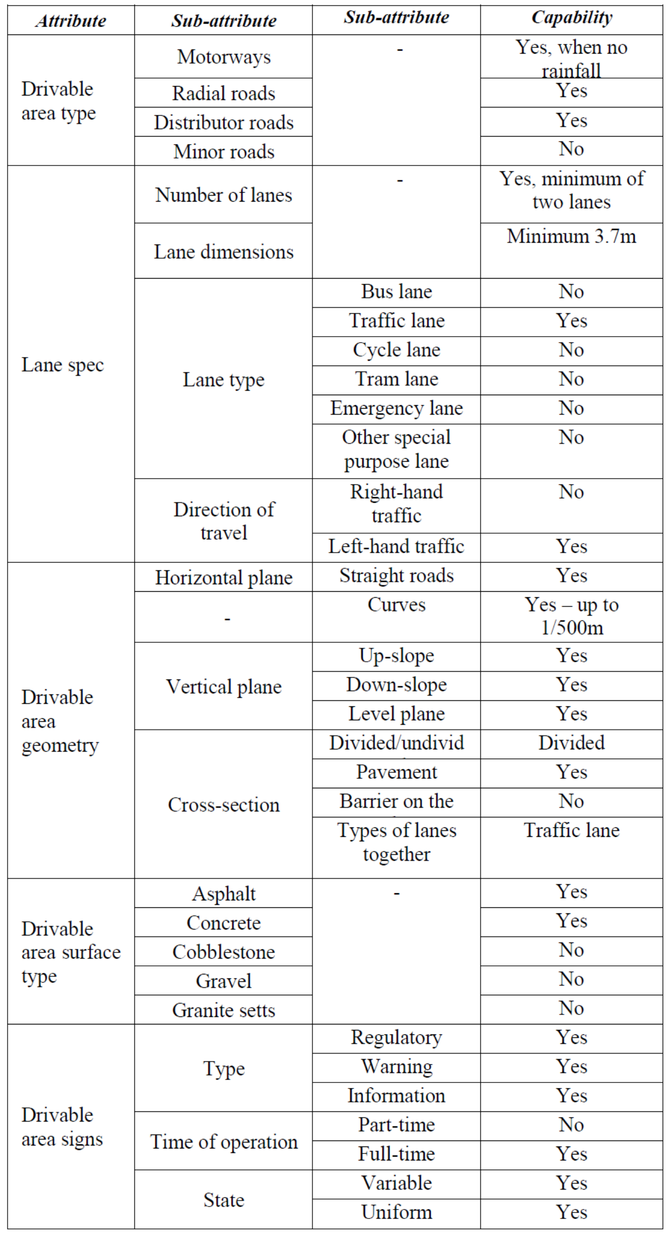
*This WP will describe the **semantic** and **syntactic** description of the ODD description for format. It will include the capability to describe **conditional** ODD description using some or all ODD **attributes**. The format should be able to be used to simulation execution, but not limited to simulation*

Definition of ODD by SAE J3016 (2018) –

"Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics".

Why we need an ODD format

- Using a tabular format example from PAS 1883 Annex A [1]
- Lack of grammar/rules
- Need to interpretation of missing attributes
- Challenge to illustrate dependencies (conditional ODD statement)
- Limited implementation potentials
- Not necessary compact
- Mixture of data properties, units, classes
- …



Attribute	*Sub-attribute*	*Sub-attribute*	*Capability*
Drivable area type	Motorways	-	Yes, when no rainfall
	Radial roads		Yes
	Distributor roads		Yes
	Minor roads		No
Lane spec	Number of lanes	-	Yes, minimum of two lanes
	Lane dimensions		Minimum 3.7m
	Lane type	Bus lane	No
		Traffic lane	Yes
		Cycle lane	No
		Tram lane	No
		Emergency lane	No
		Other special purpose lane	No
	Direction of travel	Right-hand traffic	No
		Left-hand traffic	Yes
Drivable area geometry	Horizontal plane	Straight roads	Yes
	-	Curves	Yes – up to 1/500m
	Vertical plane	Up-slope	Yes
		Down-slope	Yes
		Level plane	Yes
	Cross-section	Divided/undivid	Divided
		Pavement	Yes
		Barrier on the	No
		Types of lanes together	Traffic lane
Drivable area surface type	Asphalt	-	Yes
	Concrete		Yes
	Cobblestone		No
	Gravel		No
	Granite setts		No
Drivable area signs	Type	Regulatory	Yes
		Warning	Yes
		Information	Yes
	Time of operation	Part-time	No
		Full-time	Yes
	State	Variable	Yes
		Uniform	Yes

[1] - "Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification," *The British Standards Institution, BSI PAS 1883*. 2020.

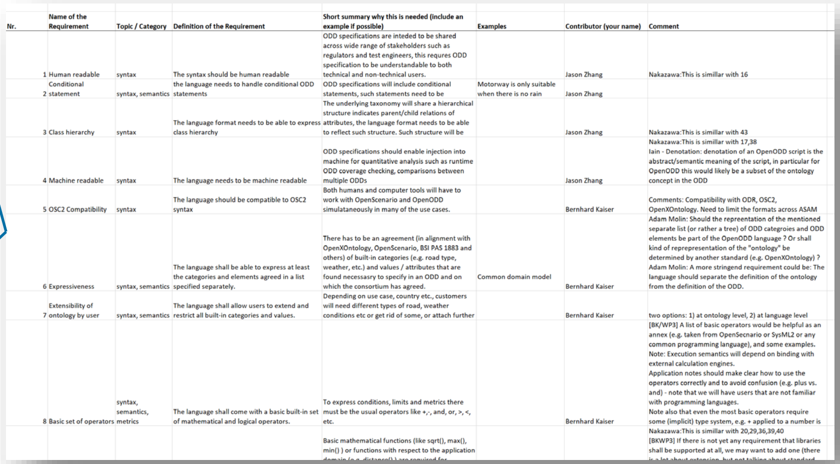
Requirement clusters

- Development engineer
- Test engineer
- Data scientists
- Tool developer
- Scenario editor/creator
- Data annotation engineer
- Safety engineer
- Infrastructure operator

15 requirement clusters

- Human readability
- Machine readability /query
- Composability
- Parameterization/ templating
- Compatible with OpenX
- Conditional statement
- Extensibility of ontology
- Binary boundary
- Class/ metric/ datatype/ units
- Probability/ uncertainty/ risk
- Class hierarchy/ expressiveness/ abstraction
- Extensibility by functions
- Operators
- Permissive/ restrictive
- Integration with scenario based testing workflow

Requirement list



Nr.	Name of the Requirement	Topic / Category	Definition of the Requirement	Short summary why this is needed (include an example if possible)	Examples	Contributor (your name)	Comment
1	Human readable	syntax	The syntax should be human readable	ODD specifications are inteded to be shared across wide range of stakeholders such as regulators and test engineers, this requres ODD specification to be understandable to both technical and non-technical users.		Jason Zhang	Nakazawa:This is simillar with 16
2	Conditional statement	syntax, semantics	the language needs to handle conditional ODD statements	ODD specifications will include conditional statements, such statements need to be	Motorway is only suitable when there is no rain	Jason Zhang	
3	Class hierarchy	syntax	The language format needs to be able to express class hierarchy	The underlying taxonomy will share a hierarchical structure indicates parent/child relations of attributes, the language format needs to be able to reflect such structure. Such structure will be		Jason Zhang	Nakazawa:This is simillar with 43
4	Machine readable	syntax	The language needs to be machine readable	ODD specifications should enable injection into machine for quantitative analysis such as runtime ODD coverage checking, comparisons between multiple ODDs		Jason Zhang	Nakazawa:This is simillar with 17,38 Iain - Denotation: denotation of an OpenODD script is the abstract/semantic meaning of the script, in particular for OpenODD this would likely be a subset of the ontology concept in the ODD
5	OSC2 Compatibility	syntax	The language should be compatible to OSC2 syntax	Both humans and computer tools will have to work with OpenScenario and OpenODD simulataneously in many of the use cases.		Bernhard Kaiser	Comments: Compatibility with ODR, OSC2, OpenXOntology. Need to limit the formats across ASAM
6	Expressiveness	syntax, semantics	The language shall be able to express at least the categories and elements agreed in a list specified separately.	There has to be an agreement (in alignment with OpenXOntology, OpenScenario, BSI PAS 1883 and others) of built-in categories (e.g. road type, weather, etc.) and values / attributes that are found necessasry to specify in an ODD and on which the consortium has agreed.	Common domain model	Bernhard Kaiser	Adam Molin: Should the repreentation of the mentioned separate list (or rather a tree) of ODD categroies and ODD elements be part of the OpenODD language ? Or shall kind of reprepresentation of the "ontology" be determined by another standard (e.g. OpenXOntology) ? Adam Molin: A more stringend requirement could be: The language should separate the definition of the ontology from the definition of the ODD.
7	Extensibility of ontology by user	syntax, semantics	The language shall allow users to extend and restrict all built-in categories and values.	Depending on use case, country etc., customers will need different types of road, weather conditions etc or get rid of some, or attach further		Bernhard Kaiser	two options: 1) at ontology level, 2) at language level
8	Basic set of operators	syntax, semantics, metrics	The language shall come with a basic built-in set of mathematical and logical operators.	To express conditions, limits and metrics there must be the usual operators like +,-, and, or, >, <, etc.		Bernhard Kaiser	[BK/WP3] A list of basic operators would be helpful as an annex (e.g. taken from OpenSecnario or SysML2 or any common programming language), and some examples. Note: Execution semantics will depend on binding with external calculation engines. Application notes should make clear how to use the operators correctly and to avoid confusion (e.g. plus vs. and) - note that we will have users that are not familiar with programming languages. Note also that even the most basic operators require some (implicit) type system, e.g. + applied to a number is
				Basic mathematical functions (like sqrt(), max(), min()) or functions with respect to the application			Nakazawa:This is simillar with 20,29,36,39,40 [BKWP3] If there is not yet any requirement that libraries shall be supported at all, we may want to add one (there

Requirement clusters

- Human readability
- Machine readability /query
- Composability
- Parameterization/ templating
- Compatible with OpenX
- Conditional statement
- Extensibility of ontology
- Binary boundary
- Class/ metric/ datatype/ units
- Probability/ uncertainty/ risk
- Class hierarchy/ expressiveness/ abstraction
- Extensibility by functions
- Operators
- Permissive/ restrictive
- Integration with scenario based testing workflow

Conditional statement cluster

Summary:
Language specification needs to support conditional statements for expressing reduced ODD. This can be tackled by allowing the language to impose (extra) constraints on the operational range of specific ODD elements in any level of the ODD hierarchy.

Example challenge:
Considering the taxonomy from PAS 1883, we would like to express that only *Motorway* is suitable and it is only suitable when there is no *Rain* (conditional ODD statement), within the *Geometry Up-Slope* is not suitable.

Explained using query semantics:
The example challenge can be illustrated using a query semantics

*SELECT * FROM situations WHERE (Motorway AND NOT Rain) AND NOT Up-Slope*
Motorway ^ ¬Rain ^ ¬Up-Slope

Example syntax 1: (we are considering all entries of syntax based on discussion, and also from members input)

SUITABLE Motorway EXCEPT WHEN Rain
UNSUITABLE Up-Slope