iSCAML
Format Specification

"independent Scenario Advanced Meta Language"

DOCUMENT CHANGE RECORD

I = Issued; C = Checked
AI = Approved (iMAR)
AC = Approved (Customer)

Rev.	Paragraph	Comments		Date	Name	Function
1.00	All	Document created	I	17.03.20	DoNe	DE
1.01	1, 1.5, 0	Features and examples added	I	06.04.20	DoNe	DE
1.02	1.5	VerdictLevel added	I	08.04.20	DoNe	DE
1.03	2.3.1	Events added: controller error, time to closest approach	I	15.04.20	DoNe	DE
1.04	All	Description: Object line added	I	09.06.20	DoNe	DE
1.05	3	Examples: Object lines added	I	09.06.20	DoNe	DE
1.06	All	Expanded frame and unit description, minor bug fixes	I	17.06.20	DoNe	DE
1.07	All	Updated structs, corrected data types	I	26.11.20	DoNe	DE
1.08	1.4.1.2, 2	Bugfix: Reserved fields added; adapted lane offset in example	I	07.12.20	DoNe	DE
1.09	1.4.1.11	Description for Event ControllerError adapted	I	16.12.20	DoNe	DE
1.10	2	Changed parameter names, following CamelCase convention, corrected data types for several parameters Object description expanded for ObjectTypes and relevant parameters	I	10.03.21	AlMe	DE
1.11	Appendix A	Geometric Dependencies added	I	12.04.21	ToSt	DE
1.12	1.5, 2	New Phase type added: SyncPos; Phase lines: ActionBits replace previous Reserved fields	I	03.05.21	DoNe	DE
1.13	2.4, 2.5	New Event type added: Internal State; changed data type for ActionBits	I	07.05.21	DoNe	DE
1.14	2, 3	Value adaptions for AbsOrRel, HeadingType; name adaptions from “or_” to “Heading”	I	07.06.21	DoNe	DE
1.15	3	Added transfer info	I	03.12.21	DoNe	DE
1.16	all	Minor adaptiations, formatting, clarification	C/AI	04.12.21	EvH	CEO

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abbreviations and acronyms

Table 1: Abbreviations

Expression	Description
VUT	Vehicle Under Test
TSV	Traffic Simulation Vehicle
iSCAML	independent SCenario Advanced Meta Language
FTP	File Transfer Protocol

INTRODUCTION

This document contains the specification of iMAR's **i**ndependent **SC**enario **A**dvanced **M**eta **L**anguage **iSCAML**. The purpose of iSCAML is to describe the behavior of individual movable and non-movable participants within a traffic scenario on a proving ground. The language is designed to provide a seamless support of ISO 22133.

iSCAMl follows the approach to distinguish between map and maneouvers. The map contains the lanes and additional information and for iSCAML the usage of OpenDrive format is used. iSCAML contains the logical description of the complex or simple traffic scenario to be executed on the proving ground. Interfaces from well-known simulation tools are available to generate data in iSCAML description.

The iSCAML file contains the necessary information for the participant to determine its trajectories. It contains the necessary dependencies to the other objects (e.g. trigger/action relations).

iSCAML is neither limited to the number of parallel operated scenarios nor the number of objects being operated within the scenario.

Within a scenario that contains multiple participants (objects), each participant owns an individual iSCAML based scenario description. It can be transferred to each object on several ways:

- Transmission of the individual information from the proving ground control center upfront test execution to each object via a file. The standardized File Transfer Protocol (FTP) is used for most efficient execution via the IP based network between Control Center and Moveable and Non-Moveable Objects.
- Transmission of additional individual information from the proving ground control center to each object via a file (by FTP) during test execution. The iSCAML description contains information when and how to switch on the Object from one iSCAML scenario file to another on (e.g. event controlled)
- Transmission of the individual information from the proving ground control center upfront test execution to each object via the SLTM ISO 22133 message or a sequence of messages (line by line or ensemble of lines). This method requires more data transmission capacity due to message overhead.
- Transmission of additional individual information from the proving ground control center to each object via an ISO 22133 message or a sequence of messages during test execution.

Using iSCAML language, traffic scenario descriptions can be updated upfront or during test execution. The update during test execution can be performed by sending additional iSCAML lines or ensembles of lines to the object to be appended to the information being already present on the object - but as an alternative also another iSCML scenario description (file) can be uploaded via FTP to the object together with a command which defines when the object shall switch from the current iSCAML scenario description (file) to the just uploaded new iSCAML scenario description.

iSCAML follows the same approach as well-known scenario descriptions: It splits the description of the motion of an object into a ground based map (OpenDrive as road description; https://www.asam.net/standards/detail/opendrive/), where the lanes are defined, and a motion description, which contains all dynamic information like speed, trigger/action relations etc.

In the following description the usage of the wording "iSCAML description file" and "iSCAML scenario description" is equivalent, independent whether the scenario description is stored within a file, a table, a list or something else. The usage of the text file is recommended because it is also easily human readable (compared e.g. to XML). iMAR is willing to provide an iSCAML parser for C/C++ as example.

The used coordinate frames as the same as used in ISO 22133. They are based on DIN ISO 8855 / DIN ISO 70'000.

1 DETAILED FORMAT DESCRIPTION

Every participant of a traffic scenario is controlled by an individual iSCAML based scenario description that contains the trajectory information for itself. This information has to be present on the object (vehicle, traffic light, wind engine etc.) [1]. In the case of a file representation the name of the file shall have a prefix *iScenarioData_*, followed by a specific name that functions as an ID within a scenario. This ID is named *VehicleID.* In case that the information is distributed via the dedicated ISO 22133 message, this ID is content of the message.

Examples: *iScenarioData_TSV1*, *iScenarioData_VUT*, …

In case of file transmission of the scenario description from Control Center to the moveable and non-moveable objects, the iSCAML file is a text file that consists of individual lines. Unix style must be used to indicate a new line. Therefore, a line feed (<LF>; 0x0A; dez. 10) should be used, but no carriage return (<CR>; 0x0D; dez. 13).

Clarification:
A scenario typically consists of the motion of several movable objects and activities of non-movable objects. Each object can react on conditions of other objects. The behaviour of each object can be the root cause (trigger) for the activity (action) of another object. The behaviour of each object can be described in an individual iSCAML scenario file, which can contain itself several trajectory Phases, which all together and over all objects describe the entire scenario.

There are five general types that can be described by a line: Object, GeoReference, Path, Event, Phase.

- **Object**
  contains information about a participant's (object's) dimensions, and whether it is specified as a static or a dynamic participant. If the participant is static, its position is given as well. The participant is identified by using the VehicleID.
- **GeoReference**
  contains a PROJ string that describes the applied coordinate system for position information within the individual iSCAML file. A detailed description is given in 1.2.
- **Path**
  contains a string to describe a path (folder) of the local file system (in case the scenario is represented as file).
- **Event**
  contains information about a condition that is checked to be fulfilled. There are multiple options to check for positions, velocities, etc. A detailed description is given in 1.3.
- **Phase**
  contains information about a trajectory segment and a list of events that are applied during the execution of the trajectory segment. There are multiple options for the usage of a trajectory segment. A detailed description is given in 1.5.

Only one (1) line as GeoReference is currently supported. There can be multiple lines for Object, Path, Event and Phase.

Example for the usage of Phase and Event:

> In the first place, the participant uses a Phase that describes the movement along a road with constant velocity. This Phase can be linked to an Event that checks whether a timeout of 20 s is

[1] In the case of a file representation within the iMAR implementation, the file has to be located in directory */mnt/data/update of the local, miniaturized iVCTRL vehicle control computer*. The name of the file has a prefix *iScenarioData_*, followed by a specific name that functions as an ID within a scenario. This ID is named *VehicleID.*

reached for this trajectory. When the 20 s are over, the Event is active and forces a transition to a new trajectory element (Phase), e.g. a deceleration phase.

Any line that doesn't start with one of the key words Object, GeoReference, Path, Event or Phase, will be ignored.

Position info of a participant is given for its reference position. This position is specified as:

- Vehicle: lateral center of rear axle, projected on the ground.
- Other participants: geometrical center of participant, projected on the ground.

In general, using the message protocol based ISO 22133 protocol (instead of sending the scenario file via FTP from Control Center to the object), each line or an ensemble of multiple lines can be transmitted via one message.

1.1 Object

The line starts by using the text *Object* as an indicator. The following text consists of additional parameters. Every additional parameter must start with one space, followed by the parameter's name as a string, followed by the symbol =, followed by a value. Dependent on the parameter, the value can be an integer, a float, a double or a string.

Every line describes specifications of one single participant. Every participant whose VehicleID is used within the Event or Phase lines of an iSCAML file needs to be specified by an Object line in the same file.

Field Number	Parameter Name	Parameter Data Type	Description
1	Object	*n.a.*	Indicates line type. No symbol = and no parameter value.
2	ID	Unsigned Integer	Every line of type Object needs an individual ID. Start index: 1. Increase index by 1 for every additional Object line.
3	VehicleID	String	Contains the VehcileID of the specified participant.
4	ObjectCategory	Integer	Specifies the category of the object: • 100: Car • 101: Van • 102: Truck • 103: Trailer • 104: Semitrailer • 105: Bus • 106: Motorbike • 107: Bicycle • 108: Train • 109: Tram • 200: Pedestrian • 201: Wheelchair • 202: Animal • 300: Barrier • 301: GuardRail • 302: Other

Field Number	Parameter Name	Parameter Data Type	Description
5	MaxSpeed	Float	Object's allowed maximum speed in m/s for this operation
6	MaxAccel	Float	Object's allowed maximum acceleration in m/s² for this operation
7	MaxDecel	Float	Object’s allowed maximum deceleration in m/s² for this operation
8	Mass	Float	Mass of the object (currently not used, use ***0*** for the parameter’s value)
9	CenterX	Double	x-component of vector from participant’s reference position to its geometric center. Frame: partici-pant’s body frame (x: forward, y: left, z: up according to DIN ISO 8855). Unit: m.
10	CenterY	Double	y-component of vector from participant’s reference position to its geometric center. Frame: partici-pant’s body frame (x: forward, y: left, z: up according to DIN ISO 8855). Unit: m.
11	CenterZ	Double	z-component of vector from participant’s reference position to its geometric center. Frame: partici-pant’s body frame (x: forward, y: left, z: up according to DIN ISO 8855). Unit: m.
12	DimensionW	Double	Dimension of participant, referring to the y-axis of its body frame (width). Unit: m.
13	DimensionL	Double	Dimension of participant, referring to the x-axis of its body frame (length). Unit: m.
14	DimensionH	Double	Dimension of participant, referring to the z-axis of its body frame (height). Unit: m.
15	PosType	Integer	Specifies the format that is used in fields 16..21 to specifiy the participant’s position: • Value=-1: participant is non-static (moveable), therefore no constant position info is given in Object line. To get navigation of this participant in real-time during test, use additional communication between participants. • Value=0: participant is static (non-moveable object), position is given in OpenX-World coordinates, see Table 3. • Value=3: participant is static, position is given in OpenX-Road coordinates, see Table 4. • Value=5: participant is static, position is given in OpenX-Lane coordinates, see Table 5.

Field Number	Parameter Name	Parameter Data Type	Description
16..21	*Diverse, see specific specification*	*Diverse, see specific specification*	These fields are used to specify the object's position, depending on value of *PosType:* • PosType=-1: every field 16..21 is reserved. Set every value to ***0***. • PosType=0: set fields 16..21 according to Table 3. • PosType=3: set fields 16..21 according to Table 4. • PosType=5: set fields 16..21 according to Table 5.

Table 2: Object line syntax definition

Field Number	Parameter Name	Parameter Data Type	Description
16	X	Double	East-component of position info in m. Frame origin: specified GeoReference.
17	Y	Double	North-component of position info in m. Frame origin: specified GeoReference.
18	Z	Double	Up-component of position info in m. Frame origin: specified GeoReference.
19	Roll	Float	Roll angle in rad. Frame: ENU (see DIN ISO 8855)
20	Pitch	Float	Pitch angle in rad. Frame: ENU (see DIN ISO 8855)
21	Yaw	Float	Yaw angle in rad. Frame: ENU (see DIN ISO 8855)

Table 3: Object Position Info: OpenX-World coordinates

Field Number	Parameter Name	Parameter Data Type	Description
16	RoadID	String	Value for parameter Road ID of position info (see specification of OpenDrive for details).
17	S	Double	Value for parameter S of position info (see specification of OpenDrive for details)
18	T	Double	Value for parameter T of position info (see specification of OpenDrive for details)
19	Heading	Double	Heading rotation of participant in rad with respect to the track's orientation (tangent on track). Corresponding axis: Up-axis of track at given position.
20	HeadingType	Integer	Set value to ***1*** *(parameter for future use)*.
21	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 4: Object Position Info: OpenX-Road coordinates

Field Number	Parameter Name	Parameter Data Type	Description
16	RoadID	String	Value for parameter Road ID of position info, See specification of OpenDrive for details.
17	LaneID	Integer	Value for parameter Lane ID of position info, See specification of OpenDrive for details.
18	Offset	Double	Value for parameter Offset of position info, See specification of OpenDrive for details.
19	S	Double	Value for parameter S of position info, See specification of OpenDrive for details.
20	Heading	Double	Heading rotation of participant in rad with respect to the track's orientation. Corresponding axis: Up-axis of local ENU-frame with specified GeoReference as origin. Unit: rad.
21	HeadingType	Integer	Set value to ***1***.

Table 5: Object Position Info: OpenX-Lane coordinates

1.2 GeoReference

The line starts by using the text *GeoReference* as an indicator, followed by one empty space. The following text consists of the according PROJ string. Used version of PROJ: 5.2.0. See document reference list for its specification.

Text line example:

GeoReference +proj=tmerc +lon_0=7.40825 +lat_0=49.21814 +ellps=WGS84 +datum=WGS84 +no_defs +zone=32

1.3 Path

The line starts by using the text *Path* as an indicator. The following text consists of two additional parameters. Every additional parameter must start with one space, followed by the parameter's name as a string, followed by the symbol =, followed by a value. The value of the first parameter is given as an unsigned integer. The value of the second parameter is given as a string.

Text line representation:

Path ID=x PathString=x

The second parameter describes a path that can be referenced to by using the ID from the first parameter.

1.4 Event

The line starts by using the text *Event* as an indicator. The following text consists of additional parameters. Every additional parameter must start with one space, followed by the parameter's name as a string, followed by the symbol =, followed by a value. Dependent on the parameter, the value can be an integer, a double or a string.

There is a pool of methods that can be chosen from for the usage in an Event line. Every method uses a unique function to check whether certain conditions are fulfilled. Parameters within the Event line are used to select a certain method and parametrize it.

There also exists the possibility to combine multiple methods within one Event. Then, the Event only initiates a Phase transition if all used methods result with a positive check.

The according syntax is specified in Table 6.

Field Number	Parameter Name	Parameter Data Type	Description
1	Event	*n.a.*	Indicates line type. No symbol = and no parameter value.
2	ID	Unsigned Integer	Every line of type Event needs an individual ID. Start index: 1. Increase index by 1 for every additional Event line.
3..n	*Diverse, see specific specification*	*Diverse, see specific specification*	Parameters according to the used method must be inserted here. Methods are described in 1.4.1 An arbitrary amount of parallel methods is supported. The according entries must be listed consecutively

Table 6: Event Line Syntax Definition

Text line representation:

Event ID=x Method1_Parameter1=x Method1_Parameter2=x … Method2_Parameter1=x …

1.4.1 Event Methods

An overview of all available methods can be found in Table 7.

Method Name	Parameter Specification	Description
Distance Difference	See Table 8	Compares the position difference between two participants to a set limit.
Velocity Difference	See Table 13	Compares the velocity difference between two participants to a set limit.
Phase Timeout	See Table 14	Checks if the time since the current Phase started has reached a set maximum.
Trajectory Ended	See Table 15	Checks if the trajectory segment of the current phase has come to an end.
Velocity Absolute	See Table 16	Compares the velocity of a participant to a set limit.
Distance Absolute	See Table 17	Compares the distance of a participant from a set position to a set limit.
Scenario Timeout	See Table 21	Checks if the time since the scenario started has reached a set maximum.

Relative Distance	See Table 22	Compares the position difference between two participants to a set limit, using one participant's body frame.
Position Absolute	See Table 23	Compares the distance of a participant from a set position to a set maximum.
Position Difference	See Table 27	Compares the position difference between two participants to a set maximum.
Controller Error	See Table 32	Compares controller error info to a set limit.
Time to Closest Approach	See Table 33	Compares time to closest approach to a set limit.
Internal State	See Table 34	Checks a single bit of a participant's internal state (e.g. indicator light state)

Table 7: Event Methods Overview

1.4.1.1 Distance Difference

Event method functionality:

Compares the position difference between two participants to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***2*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID1	String	Contains the VehicleID of the participant Participant1. See Figure 1. Set parameter to the desired number.
4	PosType	Unsigned Integer	Sets the translation from object's position for comparison. See Figure 1. • Value=1: leverarms in local cartesian frame with given GeoReference as its origin, use specification of Table 9 for fields 5..8 • Value=2: leverarms in body frame of Object, use specification of Table 10 for fields 5..8 • Value=4: leverarms in OpenDrive Road coordinates, use specification of Table 11 for fields 5..8 • Value=6: leverarms in OpenDrive Lane coordinates, use specification of Table 12 for fields 5..8
5..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains leverarm info according to specification in *PosType*. See Table 9, Table 10, Table 11, Table 12.
9	RefPosRadius	Float	This parameter describes a distance in m. This parameter is used for a comparison according to description of parameter CheckLogic.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
10	FreeSpace	Unsigned Integer	This parameter sets how the distance measurement is performed: • Value=0: use distance between single points. • Value=1: beware participants' dimensions to determine space between bounding boxes.
11	CheckLogic	Unsigned Integer	Sets rule for comparison. See Figure 1. Active if following condition is fulfilled: • Value=0: absolute value of distance less than RefPosRadius. • Value=1: absolute value of distance less than or equal to RefPosRadius. • Value=2: absolute value of distance equal to RefPosRadius. • Value=3: absolute value of distance more than or equal to RefPosRadius. • Value=4: absolute value of distance more than RefPosRadius.
12	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 8: Event Method Distance Difference: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 1.
6	DX	Double	Contains East-component of leverarm. See Figure 1. Frame origin: specified GeoReference. Unit: m.
7	DY	Double	Contains North-component of leverarm. See Figure 1. Frame origin: specified GeoReference. Unit: m.
8	DZ	Double	Contains Up-component of leverarm. See Figure 1. Frame origin: specified GeoReference. Unit: m.

Table 9: leverarm in local cartesian frame with given GeoReference as its origin

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 1.
6	DX	Double	Contains x-component of leverarm. See Figure 1. Frame: body of Object (x: forward, y: left, z: up). Unit: m.

7	DY	Double	Contains y-component of leverarm. See Figure 1. Frame: body of Object (x: forward, y: left, z: up). Unit: m.
8	DZ	Double	Contains z-component of leverarm. See Figure 1. Frame: body of Object (x: forward, y: left, z: up). Unit: m.

Table 10: leverarm in body frame of Object

Field Number	**Parameter Name**	**Parameter Data Type**	**Parameter Description**
5	Object	String	Contains the VehicleID of the participant Object. See Figure 1.
6	DS	Double	Contains longitudinal adaption of OpenDrive coordinate. See Figure 1. Unit: m.
7	DT	Double	Contains lateral adaption of OpenDrive coordinate. See Figure 1. Unit: m.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 11: leverarm in relative OpenDrive Road coordinate

Field Number	**Parameter Name**	**Parameter Data Type**	**Parameter Description**
5	Object	String	Contains the VehicleID of the participant Object. See Figure 1.
6	DLane	Double	Contains adaption of LaneID of OpenDrive coordinate. See Figure 1. Unit: n.a.
7	Offset	Double	Contains adaption of offset of OpenDrive coordinate. See Figure 1. Unit: n.a.
8	DS	Double	Contains longitudinal adaption of OpenDrive coordinate. See Figure 1. Unit: m.

Table 12: leverarm in relative OpenDrive Lane coordinate

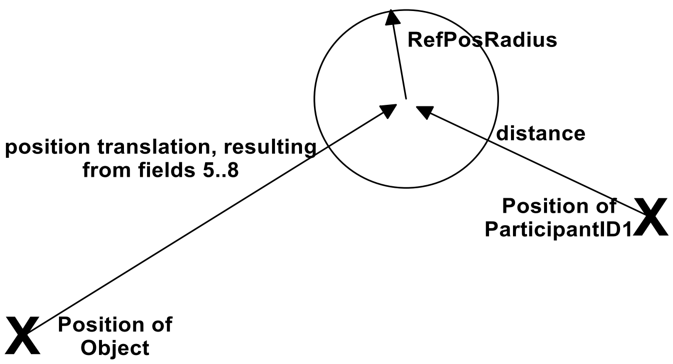

Figure 1: Event Method Distance Difference

1.4.1.2 Velocity Difference

Event method functionality:

Compares the velocity difference between two participants to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***3*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID1	String	Contains the VehicleID of Participant1.
4	ParticipantID2	String	Contains the VehicleID of Participant2.
5	VelDiff	Float	This parameter describes a limit for a velocity difference in m/s. This parameter is used for a comparison according to description of parameter CheckLogic.
6	CheckLogic	Unsigned Integer	Sets rule for comparison. Actual velocity difference is determined by subtracting velocity of Participant2 from velocity of Participant1. Active if following condition is fulfilled: • Value 0: Actual velocity difference less than VelDiff • Value 1: Actual velocity difference less than or equal to VelDiff • Value 2: Actual velocity difference equal to VelDiff • Value 3: Actual velocity difference more than or equal to VelDiff • Value 4: Actual velocity difference more than VelDiff
7	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.
8	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 13: Event Method Velocity Difference: specific parametrization

1.4.1.3 Phase Timeout

Event method functionality:

Checks if the time since the current Phase started has reached a set maximum.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***4*** for the parameter's value to indicate the desired method.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	TimeThreshold	Float	This parameter describes a time in s. This parameter is used for a comparison according to description of parameter CheckLogic.
4	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Value 0: Time span since current Phase started is less than TimeThreshold. • Value 1: Time span since current Phase started is less than or equal to TimeThreshold. • Value 2: Time span since current Phase started is equal to TimeThreshold. • Value 3: Time span since current Phase started is more than or equal to TimeThreshold. • Value 4: Time span since current Phase started is more than TimeThreshold.

Table 14: Event Method Phase Timeout: specific parametrization

1.4.1.4 Trajectory Ended

Event method functionality:

Checks if the trajectory segment of the current phase has come to an end.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***5*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 15: Event Method Trajectory Ended: specific parametrization

1.4.1.5 Velocity Absolute

Event method functionality:

Compares the velocity of a participant to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***6*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID	String	Contains the VehicleID of the participant whose actual velocity info is used for a comparison according to description of parameter CheckLogic. Set parameter to the desired number.
4	VelAbs	Float	This parameter describes a limit for a velocity in m/s. This parameter is used for a comparison according to description of parameter CheckLogic.
5	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Value 0: Participant's velocity less than VelAbs. • Value 1: Participant's velocity less than or equal to VelAbs. • Value 2: Participant's velocity equal to VelAbs. • Value 3: Participant's velocity more than or equal to VelAbs. • Value 4: Participant's velocity more than VelAbs.
6	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 16: Event Method Velocity Absolute: specific parametrization

1.4.1.6 Distance Absolute

Event method functionality:

Compares the distance of a participant from a set position to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***7*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID	String	Contains the VehicleID of the participant. See Figure 2. Set parameter to the desired number.
4	PosType	Unsigned Integer	Sets the frame for parametrized position reference. See Figure 2.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
			• value=0: reference position given in local cartesian frame with given GeoReference as its origin, use specification of Table 18 for fields 5..8. • value=3: reference position given in OpenDrive Road coordinates, use specification of Table 19 for fields 5..8. • value=5: reference position given in OpenDrive Lane coordinates, use specification of Table 20 for fields 5..8.
5..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains reference position info according to specification in *PosType*. See Table 18, Table 19, Table 20.
9	RefPosRadius	Float	This parameter describes a distance in m. This parameter is used for a comparison according to description of parameter CheckLogic.
10	FreeSpace	Unsigned Integer	This parameter sets how the distance measurement is performed: • Value=0: use distance between single points. • Value=1: beware participants' dimensions to determine space between bounding boxes.
11	CheckLogic	Unsigned Integer	Sets rule for comparison. See Figure 2. Active if following condition is fulfilled: • Value 0: absolute value of distance less than RefPosRadius. • Value 1: absolute value of distance less than or equal to RefPosRadius. • Value 2: absolute value of distance equal to RefPosRadius. • Value 3: absolute value of distance more than or equal to RefPosRadius. • Value 4: absolute value of distance more than RefPosRadius.
12	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 17: Event Method Distance Absolute: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	X	Double	Contains east-component of reference position. See Figure 2. Frame origin: specified GeoReference. Unit: m.
6	Y	Double	Contains north-component of reference position. See Figure 2. Frame origin: specified GeoReference. Unit: m.

7	Z	Double	Contains up-component of reference position. See Figure 2. Frame origin: specified GeoReference. Unit: m.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 18: reference position given in local cartesian frame with given GeoReference as its origin

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	RoadID	String	Contains RoadID-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
8	S	Double	Contains S-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
7	T	Double	Contains T-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 19: reference position given in OpenDrive Road coordinates

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	RoadID	String	Contains RoadID-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
6	LaneID	Integer	Contains LaneID-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
7	Offset	Double	Contains Offset-component of reference position. See Figure 2. Frame: OpenDrive coordinates.
8	S	Double	Contains S-component of reference position. See Figure 2. Frame: OpenDrive coordinates.

Table 20: reference position given in OpenDrive Lane coordinates

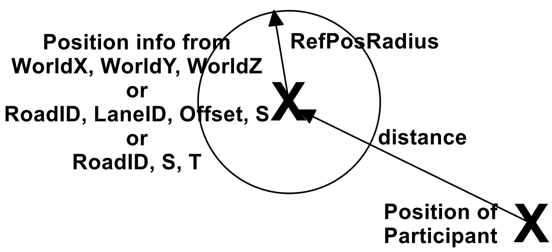

Figure 2: Event Method Distance Absolute

1.4.1.7 Scenario Timeout

Event method functionality:

Check if the time since the scenario started has reached a set maximum.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***8*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	TimeThreshold	Float	This parameter describes a time in s. This parameter is used for a comparison according to description of parameter CheckLogic.
4	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Vaue 0: Time span since the scenario started is less than TimeThreshold. • Value 1: Time span since the scenario started is less than or equal to TimeThreshold. • Value 2: Time span since the scenario started is equal to TimeThreshold. • Value 3: Time span since the scenario started is more than or equal to TimeThreshold. • Value 4: Time span since the scenario started is more than TimeThreshold.

Table 21: Event Method Scenario Timeout: specific parametrization

1.4.1.8 Relative Distance

Event method functionality:

Compares the position difference between two participants to a set limit, using one participant's body frame.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***9*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID	String	Contains the VehicleID of the second participant. First participant is the vehicle that uses this function itself. Set parameter for the second participant to the desired number.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
4	DistanceType	Unsigned Integer	Sets type of distance calculation. Parameter values are defined as follows: • 0: longitudinal distance from second participant to the first one. Frame: body frame of second participant. • 1: lateral distance from second participant to the first one. Frame: body frame of second participant. • 2: absolute value of vector from second participant to the first one.
5	Val	Float	Contains distance limit for comparison as described in CheckLogic. Unit: m.
6	FreeSpace	Unsigned Integer	This parameter sets how the distance measurement is performed: • Value 0: use distance between single points. • Value 1: beware participants' dimensions to determine space between bounding boxes.
7	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Value 0: distance as configured via parameter DistanceType is less than Val. • Value 1: distance as configured via parameter DistanceType is less than or equal to Val • Value 2: distance as configured via parameter DistanceType is equal to Val. • Value 3: distance as configured via parameter DistanceType is more than or equal to Val. • Value 4: distance as configured via parameter DistanceType is more than Val.
8	TimeThreshold	Float	This parameter describes a time in s. This parameter is used for a comparison according to description of parameter CheckLogic.

Table 22: Event Method Relative Distance: specific parametrization

1.4.1.9 Position Absolute

Event method functionality:

Compares the distance of a participant from a set position to a set maximum.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***10*** for the parameter's value to indicate the desired method.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID	String	Contains the VehicleID of the participant. See Figure 3. Set parameter to the desired number.
4	PosType	Unsigned Integer	Sets the frame for parametrized position reference. See Figure 3. • value=0: reference position given in local cartesian frame with given GeoReference as its origin. See Table 24. • value=3: reference position given in OpenDrive Road coordinates. See Table 25. • Value=5: reference position given in OpenDrive Lane coordinates. See Table 26.
5..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains reference position info according to specification in *PosType*. See Table 24, Table 25, Table 26.
9	Tolerance	Float	This parameter describes a distance limit in m. The distance between the configured reference position (PosType and according coordinates) and the position of the selected participant (ParticipantID) is compared to this distance limit. If the actual distance is not larger than the limit, the method's condition is fulfilled.
10	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 23: Event Method Position Absolute: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	X	Double	Contains east-component of reference position. See Figure 3. Frame origin: specified GeoReference. Unit: m.
6	Y	Double	Contains north-component of reference position: See Figure 3. Frame: specified GeoReference. Unit: m.
7	Z	Double	Contains up-component of reference position. See Figure 3. Frame: specified GeoReference. Unit: m.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 24: reference position given in local cartesian frame with given GeoReference as its origin

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	RoadID	String	Contains RoadID-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
6	S	Double	Contains S-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
7	T	Double	Contains T-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter’s value.

Table 25: reference position given in OpenDrive Road coordinates

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	RoadID	String	Contains RoadID-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
6	LaneID	Integer	Contains LaneID-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
7	Offset	Double	Contains Offset-component of reference position. See Figure 3. Frame: OpenDrive coordinates.
8	S	Double	Contains S-component of reference position. See Figure 3. Frame: OpenDrive coordinates.

Table 26: reference position given in OpenDrive Lane coordinates

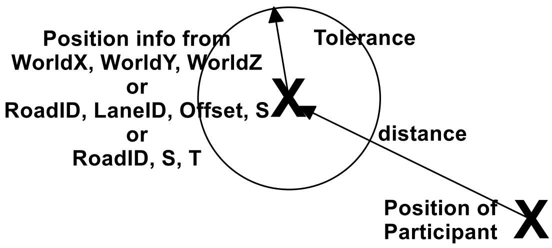

Figure 3: Event Method Position Absolute

1.4.1.10 Position Difference

Event method functionality:

Compares the position difference between two participants to a set maximum.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***11*** for the parameter’s value to indicate the desired method.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID1	String	Contains the VehicleID of the participant Participant1. See Figure 1. Set parameter to the desired number.
4	PosType	Unsigned Integer	Sets the frame for dx, dy, dz. See Figure 1. • Value=1: leverarms in local cartesian frame with given GeoReference as its origin, use specification of Table 28 for fields 5..8 • Value=2: leverarms in body frame of Object, use specification of Table 29 for fields 5..8 • Value=4: leverarms in OpenDrive Road coordinates, use specification of for fields 5..8 • Value=6: leverarms in OpenDrive Lane coordinates, use specification of for fields 5..8
5..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains leverarm info according to specification in *PosType*. See Table 28, Table 29.
9	Tolerance	Float	This parameter describes a distance limit in m. The distance between the configured reference position (PosType and according coordinates) and the position of Participant 1 is compared to this distance limit. If the actual distance is not larger than the limit, the method's condition is fulfilled.
10	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 27: Event Method Position Difference: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 4.
6	DX	Double	Contains east-component of leverarm. Frame origin: specified GeoReference. Unit: m.
7	DY	Double	Contains north-component of leverarm. Frame origin: specified GeoReference. Unit: m.
8	DZ	Double	Contains up-component of leverarm. Frame origin: specified GeoReference. Unit: m.

Table 28: leverarm in local cartesian frame with given GeoReference as its origin

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 4.
6	DX	Double	Contains x-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.
7	DY	Double	Contains y-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.
8	DZ	Double	Contains z-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.

Table 29: leverarm in body frame of Object

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 4.
6	DS	Double	Contains longitudinal adaption of OpenDrive coordinate. Unit: m.
7	DT	Double	Contains lateral adaption of OpenDrive coordinate. Unit: m.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter’s value.

Table 30: leverarm as relative OpenDrive Road coordinate

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	Object	String	Contains the VehicleID of the participant Object. See Figure 4.
6	DX	Double	Contains x-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.
7	DY	Double	Contains y-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.
8	DZ	Double	Contains z-component of leverarm. Frame: body of Object (x: forward, y: left, z: up). Unit: m.

Table 31: leverarm as relative OpenDrive Lane coordinate

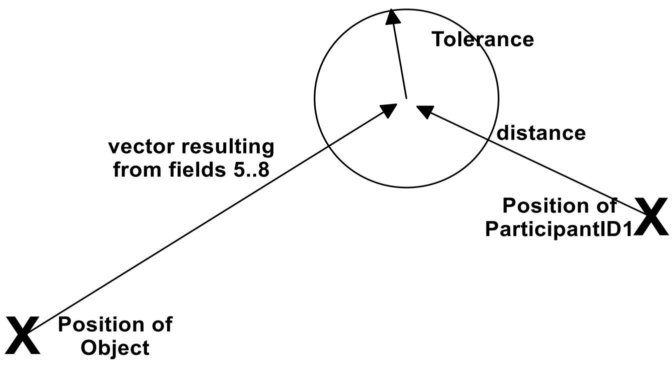

Figure 4: Event Method Position Difference

1.4.1.11 Controller Error

Event method functionality:

Compares controller error info for own vehicle to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***15*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	Mode	Unsigned Integer	Indicates which info is checked: • value=0: lateral position error; unit: m; current actual position and alignment are used for reference; if desired position is located to the left, lateral error is positive • value=1: longitudinal position error; unit: m; current actual position and alignment are used for reference; if desired position is located to the front, longitudinal error is positive • value=2: velocity error; unit: m/s; if desired velocity is greater than actual velocity, error is positive • value=3: time difference between desired and actual position, unit: s if actual velocity is non-zero and positive, the longitudinal position error is divided by actual velocity; result indicates timing error to reach the desired position
4	LimitVal	Float	Information that is determined according to *Mode* will be compared to this value.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Value 0: chosen information of selected participant is less than *LimitVal* • Value 1: chosen information of selected participant is less than or equal to *LimitVal* • Value 2: chosen information of selected participant is equal to *LimitVal* • Value 3: chosen information of selected participant is more than or equal to *LimitVal* • Value 4: chosen information of selected participant is more than *LimitVal*
6	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 32: Event Method Controller Error: specific parametrization

1.4.1.12 Time to Closest Approach

Event method functionality:

Compares time to closest approach to a set limit.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***14*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID1	String	Contains the VehicleID of the first participant whose information is evaluated.
4	ParticipantID2	String	Contains the VehicleID of the second participant whose information is evaluated.
5	TimeBarrier	Double	Time to closest approach between both participants will be compared to this value. Unit: s.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
6	CheckLogic	Unsigned Integer	Sets rule for comparison. Active if following condition is fulfilled: • Value 0: time to closest approach between both participants is less than *LimitVal* • Value 1: time to closest approach between both participants is less than or equal to *LimitVal* • Value 2: time to closest approach between both participants is equal to *LimitVal* • Value 3: time to closest approach between both participants is more than or equal to *LimitVal* • Value 4: time to closest approach between both participants is more than *LimitVal*

Table 33: Event Method Time to Closest Approach: specific parametrization

1.4.1.13 Internal State

Event method functionality:

Checks a single bit of a participant's internal state (e.g. indicator light state)

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***1*** for the parameter's value to indicate the desired method.
2	CheckTimes	Unsigned Integer	This parameter sets how often this method will be used to trigger a Phase transition. Set parameter to the desired number. If the value is set to ***0***, no limit is used.
3	ParticipantID	String	Contains the VehicleID of the participant whose information is evaluated.
4	BitNr	Unsigned Integer	Selects the single bit from the participant's internal state that is checked, possible value: 0 to 7
5	BitVal	Unsigned Integer	If this value is identical to the compared bit, the event's condition is fulfilled; otherwise not; possible values: 0, 1
6	TimeThreshold	Float	Time setting in s. Condition must be fulfilled for this time span to set the method as active.

Table 34: Event Method Internal State: specific parametrization

1.5 Phase

The line starts by using the text *Phase* as an indicator. The following text consists of additional parameters. Every additional parameter must start with one space, followed by the parameter's name as a string, followed by the symbol =, followed by a value. Dependent on the parameter, the value can be an integer, a double or a string.

There is a pool of methods that can be chosen from for the usage in a Phase line. Every method uses a unique function to generate a trajectory segment. Parameters within the Phase line are used to select a certain method and parametrize it.

Additionally, a Phase line is indicated by a Verdict level. This level is evaluated by the software in the tower control software.

Moreover, a Phase line can be used to define possible transitions to other Phases by using predefined Events.

The according syntax is specified in Table 35.

Field Number	Parameter Name	Parameter Data Type	Description
1	Phase	*n.a.*	Indicates line type. No symbol = and no parameter value.
2	ID	Unsigned Integer	Every line of type Phase needs an individual ID. Start index: 1. Increase index by 1 for every additional Phase line. When a scenario starts, the trajectory of the Phase with the ID 1 is the starting trajectory. If a transition to the Phase with the ID 0 is triggered, the scenario is over.
3..n	*Diverse, see specific specification*	*Diverse, see specific specification*	Parameters for trajectory method selection and parametrization. For specific specifications of available trajectory methods, see 1.5.1. Only one trajectory method can be selected within one Phase line.
n+1	VerdictLevel	Unsigned Integer	This level is evaluated by the software in the tower control software. It is used to specify a certain restriction to validate the test data. If value is zero, this info is not used/default.
n+2	ActionBits	Unsigned Integer	A constant value during a phase that is sent to control center (iARGUS-CMD). Can be used to trigger an action like horn or indicator lights. Action depends on interpretation by vehicle.
n+3	Reserved	Integer	Currently not used, use ***0*** for the parameter’s value.

Field Number	Parameter Name	Parameter Data Type	Description
n+4..	*Diverse, see specific specification*	*Diverse, see specific specification*	Parameters to define possible Phase transitions, triggered by Events. For specific specification of this Phase transition, see 1.5.2. An arbitrary amount of parallel definitions is supported. The according entries must be listed consecutively.

Table 35: Phase Line Syntax Definition

Text line representation:

Phase ID=x Method1_Parameter1=x Method1_Parameter2=x … ReactToEventID=x FollowingPhaseID=x ReactToEventID=x FollowingPhaseID=x …

1.5.1 Phase Trajectory Methods

An overview of all available methods can be found in Table 36.

Trajectory methods may have an automatic ending. E.g. an acceleration phase is completed at some point according to its parametrization. If a trajectory method is finished and no used Event returned with fulfilled conditions, the Phase line won't be leaved. Instead, the behavior of the method *Constant Continuation* (see 1.5.1.2) will be used until a Phase transition is triggered.

Method Name	Parameter Specification	Description
Standing at Position	See Table 37	Participant stands still at parametrized position. Trajectory segment has no automatic ending.
Constant Continuation	See Table 42	Participant's start conditions for position, alignment and velocity result from previous trajectory segment. Velocity is constant during this segment. Trajectory segment has no automatic ending.
Constant Velocity Distance/Time Limit	See Table 43	Participant's start condition for position, alignment and velocity result from previous trajectory segment. Velocity is constant during segment. Trajectory segment ends automatically.
Acceleration/Deceleration	See Table 44	Participant's start condition for position, alignment and velocity result from previous trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Trajectory segment ends automatically.
Acceleration/Deceleration Time Limit	See Table 45	Participant's start condition for position, alignment and velocity result from previous

Method Name	Parameter Specification	Description
		trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Possibility of parametrization by time. Trajectory segment ends automatically.
Acceleration/Deceleration Distance Limit	See Table 46	Participant's start condition for position, alignment and velocity result from previous trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Possibility of parametrization by distance. Trajectory segment ends automatically.
Lane Offset	See Table 47	Participant's start condition for position, alignment and velocity result from previous trajectory segment. Lateral dynamics during the trajectory segment depend on its parametrization. Longitudinal velocity is constant during this segment. Trajectory segment ends automatically.
Target Hunter	See Table 48	Participant holds constant longitudinal and lateral distance to another participant. Position, alignment and dynamics result automatically from this condition. Trajectory segment has no automatic ending.
Read Trajectory File	See Table 49	Participant follows a trajectory as defined in a used file. This file contains trajectory info as a time sequence of data as velocity, heading, etc.
Set Init Condition	See Table 50	Position and alignment of participant is specified, together with its initial velocity and acceleration at this position. Constant acceleration is assumed for the vehicle's continued movement. Trajectory segment has no automatic ending.
Synchronized Position	See Table 55	Participant reaches a defined position with a defined velocity when a referenced participant reaches a defined position. Trajectory segment ends automatically.

Table 36: Phase methods Overview

1.5.1.1 Standing at Position

Phase method functionality:

Participant stands still at parametrized position. Trajectory segment has no automatic ending.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***5*** for the parameter's value to indicate the desired method.
2	PosType	Unsigned Integer	Selects the used type for info of position and alignment. See Figure 1. • Value=0: info given in local cartesian frame with given GeoReference as its origin, use specification of Table 38 for fields 3..8 • Value=2: info given in body frame of Object, use specification of Table 39 for fields 5..8 • Value=3: info given in OpenDrive Road coordinates, use specification of Table 40 for fields 5..8 • Value=5: info given in OpenDrive Lane coordinates, use specification of Table 41 for fields 5..8
3..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains position and alignment info according to specification in *PosType*. See Table 38, Table 39, Table 40, Table 41.

Table 37: Phase Trajectory Method Standing at Position: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	X	Double	Contains east-component of desired position. Frame origin: specified GeoReference. Unit: m.
4	Y	Double	Contains north-component of desired position. Frame: specified GeoReference. Unit: m.
5	Z	Double	Contains up-component of desired position. Frame: specified GeoReference. Unit: m.
6	Roll	Double	Contains desired roll angle. Frame: ENU (see DIN ISO 8855). Unit: rad.
7	Pitch	Double	Contains desired pitch angle. Frame: ENU (see DIN ISO 8855). Unit: rad.
8	Yaw	Double	Contains desired yaw angle. Frame: ENU (see DIN ISO 8855). Unit: rad.

Table 38: info given in local cartesian frame with given GeoReference as its origin

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	Object	String	Contains the VehicleID of the participant, whose position is used as the base to set the desired position.
4	DX	Double	Contains x-component of leverarm from Object's position to the desired position, using the Object's body frame (x: forward, y: left, z: up). Unit: m.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	DY	Double	Contains y-component of leverarm from Object's position to the desired position, using the Object's body frame (x: forward, y: left, z: up). Unit: m.
6	DZ	Double	Contains z-component of leverarm from Object's position to the desired position, using the Object's body frame (x: forward, y: left, z: up). Unit: m.
7	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
8	HeadingType	Integer	Set value to ***1***.

Table 39: info given in body frame of Object

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	RoadID	String	Contains RoadID-component of desired position. Frame: OpenDrive coordinates.
4	S	Double	Contains S-component of desired position. Frame: OpenDrive coordinates.
5	T	Double	Contains T-component of desired position. Frame: OpenDrive coordinates.
6	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
7	HeadingType	Integer	Set value to ***1.***
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 40: info given in OpenDrive Road coordinates

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	RoadID	String	Contains RoadID-component of desired position. Frame: OpenDrive coordinates.
4	LaneID	Integer	Contains LaneID-component of desired position. Frame: OpenDrive coordinates.
5	Offset	Double	Contains Offset-component of desired position. Frame: OpenDrive coordinates.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
6	S	Double	Contains S-component of desired position. Frame: OpenDrive coordinates.
7	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
8	HeadingType	Integer	Set value to ***1***.

Table 41: info given in OpenDrive Lane coordinates

1.5.1.2 Constant Continuation

Phase Method functionality:

Participant's start conditions for position, alignment and velocity result from previous trajectory segment. Velocity is constant during this segment. Trajectory segment has no automatic ending.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***7*** for the parameter's value to indicate the desired method.

Table 42: Phase Trajectory Method Constant Continuation: specific parametrization

1.5.1.3 Constant Velocity Distance/Time Limit

Phase method functionality:

Participant's start condition for position, alignment and velocity result from previous trajectory segment. Velocity is constant during segment. Trajectory is finished due to a reached distance or a reached time. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***4*** for the parameter's value to indicate the desired method.
2	PlanType	Unsigned Integer	This value decides what track is used for the trajectory: • value=10: continue to use an OpenDrive map that has been used in the previous trajectory segment.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	VelEndAbsOrRel	Unsigned Integer	If VelEnd is used, VelEndAbsOrRel sets reference for the velocity info: 0: velocity of VelEnd is used directly 1: velocity of VelEnd is added to the participant's start condition
4	VelEnd	Float	Not used, if its value is zero. Otherwise, it will be used as the desired velocity instead of the participant's start condition. Unit: m/s. Handling according description of VelEndAbsOrRel.
5	LimitType	Unsigned Integer	This value decides how to interprete LimitVal: • Value 1: path distance in m • Value 2: duration time in s
6	LimitVal	Float	Parameter to setup trajectory segment. Interpretation according to setting of LimitType.

Table 43: Phase Trajectory Method Constant Velocity Distance/Time Limit: specific parametrization

1.5.1.4 Acceleration/Deceleration

Phase method functionality:

Participant's start condition for position, alignment and velocity result from previous trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***4*** for the parameter's value to indicate the desired method.
2	PlanType	Unsigned Integer	This value decides what track is used for the trajectory: • value=20: continue to use an OpenDrive map that has been used in the previous trajectory segment.
3	MaxAccUnsigned	Float	Sets the maximum for the absolute value of the acceleration/deceleration. Unit: m/s²
4	VelEndAbsOrRel	Unsigned Integer	This value defines the interpretation of parameter VelEnd: • value=0: value of VelEnd will be used directly as target velocity. • value=1: start velocity of participant will be increased by VelEnd during this trajectory segment.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	VelEnd	Float	Contains the velocity value that is used for the planning of this trajectory segment, according to VelEndAbsOrRel. Unit: m/s.
6	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 44: Phase Trajectory Method Acceleration/Deceleration: specific parametrization

1.5.1.5 Acceleration/Deceleration Time Limit

Phase method functionality:

Participant's start condition for position, alignment and velocity result from previous trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Possibility of parametrization by time. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***4*** for the parameter's value to indicate the desired method.
2	PlanType	Unsigned Integer	This value decides what track is used for the trajectory: • value=30: continue to use an OpenDrive map that has been used in the previous trajectory segment.
3	MaxAccUnsigned	Float	Sets the maximum for the absolute value of the acceleration/deceleration. Unit: m/s²
4	VelEndAbsOrRel	Unsigned Integer	This value defines the interpretation of parameter VelEnd: • value=0: value of VelEnd will be used directly as target velocity. • value=1: start velocity of participant will be increased by VelEnd during this trajectory segment.
5	VelEnd	Float	Contains the velocity value that is used for the planning of this trajectory segment, according to VelEndAbsOrRel. Unit: m/s
6	TimeDiff	Float	Sets the time durance of the trajectory segment in s.

Table 45: Phase Trajectory Method Acceleration/Deceleration Time Limit: specific parametrization

1.5.1.6 Acceleration/Deceleration Distance Limit

Phase method functionality:

Participant's start condition for position, alignment and velocity result from previous trajectory segment. Longitudinal dynamics during the trajectory segment depend on its parametrization. Possibility of parametrization by distance. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***4*** for the parameter's value to indicate the desired method.
2	PlanType	Unsigned Integer	This value decides what track is used for the trajectory: • value=40: continue to use an OpenDrive map that has been used in the previous trajectory segment.
3	MaxAccUnsigned	Float	Sets the maximum for the absolute value of the acceleration/deceleration. Unit: m/s²
4	VelEndAbsOrRel	Unsigned Integer	This value defines the interpretation of parameter VelEnd: • value=0: value of VelEnd will be used directly as target velocity. • value=1: start velocity of participant will be increased by VelEnd during this trajectory segment.
5	VelEnd	Float	Contains the velocity value that is used for the planning of this trajectory segment, according to VelEndAbsOrRel. Unit: m/s
6	Distance	Float	Sets maximum distance for this trajectory segment. Unit: m

Table 46: Phase Trajectory Method Acceleration/Deceleration Distance Limit: specific parametrization

1.5.1.7 Lane Offset

Phase method functionality:

Participant's start condition for position, alignment and velocity result from previous trajectory segment. Lateral dynamics during the trajectory segment depend on its parametrization. Longitudinal velocity is constant during this segment. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***6*** for the parameter's value to indicate the desired method.
2	PlanType	Unsigned Integer	This value decides what track is used for the trajectory: • value=12: continue to use an OpenDrive map that has been used in the previous trajectory segment.
3	MaxLatAccUnsigned	Float	Sets the maximum for the absolute value of the lateral acceleration. Unit: m/s²
4	AbsOrRel	Unsigned Integer	Use ***0*** for the parameter's value.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
5	LatOffset	Float	Value specifies lateral distance for lane offset. Unit: m. value<0: lane offset to the right value>0: lane offset to the left
6	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 47: Phase Trajectory Method Lane Offset: specific parametrization

1.5.1.8 Target / Hunter

Phase method functionality:

Participant (hunter) holds constant longitudinal and lateral distance to another participant (target). Position, alignment and dynamics result automatically from this condition. Trajectory segment has no automatic ending.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***2*** for the parameter's value to indicate the desired method.
2	ParticipantID	String	Contains the VehicleID of the participant that is used as the target.
3	LatPosDiff	Float	Lateral component of the leverarm from target to hunter. Interpretation dependent on use case: • If Mode=1: Frame: body frame of target. Unit: m. Positive values for positions on the left side of target. • If Mode=2: Frame: track frame. Info is given as number of lanes from target to hunter.
4	LonPosDiff	Float	Longitudinal component of the leverarm from target to hunter. Interpretation dependent on use case: • If Mode=1: Frame: body frame of target. Unit: m. Positive values for position in front of target. • If Mode=2: Frame: track frame. Unit: m
5	Timeout	Float	Sets the maximum time that is tolerated without data updates from the target to the hunter. Unit: s.
6	Mode	Integer	Specifies usage of given leverarms: • Value=1: use LatPosDiff and LonPosDiff in Target's body frame to determine hunter's position • Value=2: use LatPosDiff and LonPosDiff in OpenDrive track frame to determine hunter's position

Field Number	Parameter Name	Parameter Data Type	Parameter Description
7	FreeSpaceLon	Integer	Sets interpretation of LonPosDiff: • Value=0: longitudinal distance between reference positions • Value=1: longitudinal distance between BoundingBoxes
8	FreeSpaceLat	Integer	Sets interpretation of LatPosDiff: • Value=0: lateral distance between reference positions • Value=1: lateral distance between BoundingBoxes

Table 48: Phase Trajectory Method Target Hunter: specific parametrization

1.5.1.9 Read Trajectory File

Phase method functionality:

Participant follows a trajectory as defined in a used file. This file contains trajectory info as a time sequence of data as velocity, heading, etc.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***1*** for the parameter's value to indicate the desired method.
2	PathID	Unsigned Integer	ID of Path line that is used to set the path of the trajectory file. See 1.3.
3	AbsOrRel	Unsigned Integer	Possible adaptions of trajectory data: 0: no adaptions of trajectory data 1: adapt alignment and position of trajectory's data to conditions of previous trajectory segment.

Table 49: Phase Read Trajectory File: specific parametrization

1.5.1.10 Set Init Condition

Phase method functionality:

Position and alignment of participant is specified, together with its initial velocity and acceleration at this position. Constant acceleration is assumed for the vehicle's continued movement. Trajectory segment has no automatic ending.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***8*** for the parameter's value to indicate the desired method.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
2	PosType	Unsigned Integer	Selects the used type for info of position and alignment. See Figure 1. • Value=0: info given in local cartesian frame with given GeoReference as its origin, use specification of Table 51 for fields 3..8 • Value=2: info given in body frame of Object, use specification of Table 52 for fields 3..8 • Value=3: info given in OpenDrive Road coordinates, use specification of Table 53 for fields 3..8 • Value=5: info given in OpenDrive Lane coordinates, use specification of Table 54 for fields 3..8
3..8	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains position and alignment info according to specification in *PosType*. See Table 51, Table 52, Table 53, Table 54.
9	InitVel	Float	Absolute value for velocity of participant at given position. Assumption: movement only along headed direction. Unit: m/s. This initial value together with the integral of the acceleration result in the velocity values along the further course of this trajectory segment.
10	InitAcc	Float	Absolute value for acceleration of participant at given position. Assumption: movement only along headed direction. Unit: m/s². This value is assumed to be constant while this trajectory segment is used.
11	MoveAlongTrack	Integer	This parameter decides whether the further trajectory for this participant will use the references OpenDrive map to move along this track. • Value=0: No usage of an external map. Move along straight line. • Value=1: Movement of participant will be along the OpenDrive map.

Table 50: Phase Trajectory Method Set Init Position: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	X	Double	Contains east-component of desired position. Frame origin: specified GeoReference. Unit: m.
4	Y	Double	Contains north-component of desired position. Frame origin: specified GeoReference. Unit: m.
5	Z	Double	Contains up-component of desired position. Frame origin: specified GeoReference. Unit: m.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
6	Roll	Double	Contains desired roll angle. Frame: ENU (see DIN ISO 8855). Unit: rad.
7	Pitch	Double	Contains desired pitch angle. Frame: ENU (see DIN ISO 8855). Unit: rad.
8	Yaw	Double	Contains desired yaw angle. Frame: ENU (see DIN ISO 8855). Unit: rad.

Table 51: info given in local cartesian frame with given GeoReference as its origin (see Table 50)

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	Object	String	Contains the VehicleID of the participant, whose position is used as the base to set the desired position.
4	DX	Double	Contains x-component of leverarm from Object’s position to the desired position, using the body frame of Object (x: forward, y: left, z: up). Unit: m.
5	DY	Double	Contains y-component of leverarm from Object’s position to the desired position, using the body frame of Object (x: forward, y: left, z: up). Unit: m.
6	DZ	Double	Contains z-component of leverarm from Object’s position to the desired position, using the body frame of Object (x: forward, y: left, z: up). Unit: m.
7	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object’s heading. • value=3.1416: desired heading is directed opposite to Object’s heading.
8	HeadingType	Integer	Set value to ***1***.

Table 52: info given in body frame of Object (see Table 50)

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	RoadID	String	Contains RoadID-component of desired position. Frame: OpenDrive coordinates.
4	S	Double	Contains S-component of desired position. Frame: OpenDrive coordinates.
5	T	Double	Contains T-component of desired position. Frame: OpenDrive coordinates.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
6	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
7	HeadingType	Integer	Set value to ***1***.
8	Reserved	Integer	Currently not used, use ***0*** for the parameter's value.

Table 53: info given in OpenDrive Road coordinates (see Table 50)

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	RoadID	String	Contains RoadID-component of desired position. Frame: OpenDrive coordinates.
4	LaneID	Integer	Contains LaneID-component of desired position. Frame: OpenDrive coordinates.
5	Offset	Double	Contains Offset-component of desired position. Frame: OpenDrive coordinates.
6	S	Double	Contains S-component of desired position. Frame: OpenDrive coordinates.
7	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
8	HeadingType	Integer	Set value to ***1.***

Table 54: info given in OpenDrive Lane coordinates (see Table 50)

1.5.1.11 Synchronized Position

Phase method functionality:

Participant reaches a defined position with a defined velocity when a referenced participant reaches a defined position. Trajectory segment ends automatically.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	TypeID	Unsigned Integer	Use ***9*** for the parameter's value to indicate the desired method.
2	MasterID	String	VehicleID of referenced participant

Field Number	Parameter Name	Parameter Data Type	Parameter Description
3	PosTypeMaster	Unsigned Integer	Selects the used type for info of final position and alignment for participant that acts as the master. See Figure 1. • Value=5: info given in OpenDrive Lane coordinates, use specification of Table 56 for fields 4..9
4..9	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains position and alignment info according to specification in *PosTypeMaster*. See Table 56.
10	PosTypeOwn	Unsigned Integer	Selects the used type for info of final position and alignment for own participant. See Figure 1. • Value=5: info given in OpenDrive Lane coordinates, use specification of Table 56 for fields 11..16
11..16	*Diverse, see specific specification*	*Diverse, see specific specification*	Contains position and alignment info according to specification in *PosTypeOwn*. See Table 56.
17	VelEnd	Float	Specifies desired velocity for own participant at final position.
18	VelEndType	Unsigned Integer	Use **0** for the parameter's value.
19	RelVelType	Unsigned Integer	Use **1** for the parameter's value.

Table 55: Phase Trajectory Method Synchronized Position: specific parametrization

Field Number	Parameter Name	Parameter Data Type	Parameter Description
4/11	RoadID	String	Contains RoadID-component of desired position. Frame: OpenDrive coordinates.
5/12	LaneID	Integer	Contains LaneID-component of desired position. Frame: OpenDrive coordinates.
6/13	Offset	Double	Contains Offset-component of desired position. Frame: OpenDrive coordinates.
7/14	S	Double	Contains S-component of desired position. Frame: OpenDrive coordinates.
8/15	Heading	Double	Following values are applicable: • value=0: desired heading corresponds to Object's heading. • value=3.1416: desired heading is directed opposite to Object's heading.
9/16	HeadingType	Integer	Set value to ***1.***

Table 56: info given in OpenDrive Lane coordinates

1.5.2 Verdict Level

This level is evaluated by the software in the tower control software. It is used to specify a certain restriction to validate the test data.

As described in Table 35, the Verdict level is set by an unsigned integer that refers to an external detailed specification. This specification is located at the tower control center. The specification is a text file that uses a structure that is related to iSCAML. If the value of the VerdictLevel setting of iSCAML is zero, this specification is not used and the restriction therefore interpreted as default.

The specification file consists of single lines, just as iSCAML. There are three types that can be described by a line: VerdictLevel, Event, Object.

- **VerdictLevel**
  contains an ID that is referred to by iSCAML. This type uses Events to specify and check for certain restrictions. See 1.5.2.1 for detailed description.
- **Event**
  contains information about a condition that is checked to be fulfilled. There are multiple options to check for positions, velocities, etc. See 1.5.2.2 for detailed description. Event lines can be used by VerdictLevel lines.
- **Object**
  contains information about dimensions of participants. If a participant is static, its position is given, too. See 1.5.2.3 for detailed description.

The VerdictLevel setting of iSCAML is transmitted by the participant towards the tower control software, see document *ICD iARGUS – Protocol Description* for details.

Additionally, every participant transmits information as e.g. actual position or controller error to the tower control software. This information is used by the tower control software to check whether the set Verdict level is fulfilled.

1.5.2.1 VerdictLevel line

Every line starts by using the text *VerdictLevel* as an indicator. The following text consists of additional parameters. Every additional parameter must start with one space, followed by the parameter's name as a string, followed by the symbol =, followed by a value. Dependent on the parameter, the value can be an integer, a double or a string.

The first parameter contains the identification of a Verdict level. Every line is identified by an individual ID. This ID is an unsigned integer. Starting at 1, every additional line must increase this ID by one.

The following parameters are used to specify the according restrictions of a Verdict level. To specify and check the restrictions, the Event methods from 1.4.1 are used. It is possible to refer to multiple Events lines within one VerdictLevel line. As soon as any refered Event line is active, the VerdictLevel is violated.

See Table 57: VerdictLevel Line for detailed description.

Field Number	Parameter Name	Parameter Data Type	Description
1	VerdictLevel	*n.a.*	Indicates line type. No symbol = and no parameter value.
2	ID	Unsigned Integer	Every line of type VerdictLevel needs an individual ID. Start index:

Field Number	Parameter Name	Parameter Data Type	Description
			1. Increase index by 1 for every additional VerdictLevel line.
3..	EventID	Unsigned Integer	Refers to defined Event line. For specific specification of these Events, see 1.5.2.2. An arbitrary amount of parallel definitions is supported. The according entries must be listed consecutively.

Table 57: VerdictLevel Line

Text line representation:

VerdictLevel ID=x EventID=x EventID=x ...

1.5.2.2 Event line

These lines' specification is identical to 1.4.

1.5.2.3 Object line

These lines' specification is identical to 1.1.

1.5.2.4 Example for a VerdictLevel Specification used by Tower Control Software

See Table 58 as an example for a VerdictLevel specification.

```
# VerdictLevel definitions
VerdictLevel ID=1 EventID=1
VerdictLevel ID=2 EventID=2 EventID=3

# Object definitions
Object ID=1 VehicleID=TSV1 ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0
Object ID=2 VehicleID=VUT ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0

# Event definitions
Event ID=1 Type=6 CheckTimes=0 ParticipantID=TSV1 VelAbs=10 CheckLogic=0 TimeThreshold=0.00
Event ID=2 Type=9 CheckTimes=0 ParticipantID=VUT DistType=0 Val=15.00 Freespace=0 CheckLogic=4 TimeThreshold=0.00
Event ID=3 Type=6 CheckTimes=0 ParticipantID=TSV1 VelAbs=10 CheckLogic=4 TimeThreshold=0.00 Type=6 CheckTimes=0 ParticipantID=TSV1 VelAbs=15 CheckLogic=0 TimeThreshold=0.00
```

Table 58: Example for a VerdictLevel Specification

Detailed explanation:

- A participant, e.g. TSV1 refers to a VerdictLevel with the value of 1 during a Phase of its iSCAML file. This value refers to the line *VerdictLevel ID=1..* of the VerdictLevel Specification that is used

by the tower control software. In Table 58 can be seen that *EventID=1* is used within that line. This Event is active if the velocity of participant TSV1 falls below 10 m/s.
Result: If TSV1 refers to a VerdictLevel with the value of 1 during a certain Phase of its iSCAML file and its actual velocity during that phase falls below 10 m/s, the VerdictLevel is violated and an according info can be displayed/logged by the tower control software.

- A participant, e.g. TSV2 refers to a VerdictLevel with the value of 2 during a Phase of its iSCAML file. This value refers to the line *VerdictLevel ID=2..* of the VerdictLevel Specification that is used by the tower control software. In Table 58 can be seen that *EventID=2* and *EventID=3* are used within that line.
  EvenID=2 is active if VUT is faster than 15 m/s.
  EvenID=3 is active if TSV1 is faster than 15 m/s or slower than 10 m/s.
  Result: If TSV2 refers to a VerdictLevel with the value of 2 during a Phase of its iSCAML file and VUT is faster than 15 m/s and TSV1 is faster than 15 m/s or slower than 10 m/s, the VerdictLevel is violated and an according info can be displayed/logged by the tower control software.

1.5.3 Phase Transition

A Phase line contains the signal text *Phase*, followed by its individual ID, followed by a trajectory segment description and a Verdict level and finally by an arbitrary amount of definitions for possible phase transitions. For every possible Phase transition, a separate set of the structure from Table 59 must be used. These parameters must be listed consecutively within the Phase line.

Field Number	Parameter Name	Parameter Data Type	Parameter Description
1	ReactToEventID	Unsigned Integer	ID of Event line that is used to check if a trigger occurs.
2	FollowingPhaseID	Unsigned Integer	In case that the conditions of the Event with the ID from ReactToEventID are fulfilled, a Phase transition occurs to the Phase with the ID FollowingPhaseID.

Table 59: Phase Transition: specification for a single transition possibility

2 EXAMPLES

For an easier access, this chapter offers multiple examples of iSCAML scenario descriptions, here represented as a file.

2.1 Lane change

In Table 60 an example for an iSCAML file is displayed. The file could have a name like *iScenarioData_TSV1*. Then, the VehicleID of the participant that uses this file would be *TSV1*.

```
# Object definitions

Object ID=1 VehicleID=TSV1 ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0

Object ID=2 VehicleID=VUT ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0


# GeoReference definitions

GeoReference +proj=tmerc +lon_0=7.40825 +lat_0=49.21814 +ellps=WGS84 +datum=WGS84 +no_defs +zone=32


# Event definitions

Event ID=1 Type=4 CheckTimes=0 TimeThreshold=5.00 CheckLogic=4

Event ID=2 Type=5 CheckTimes=0 Reserved=0

Event ID=3 Type=9 CheckTimes=0 ParticipantID=VUT DistType=0 Val=25.00 Freespace=0 CheckLogic=4 TimeThreshold=0.00

Event ID=4 Type=6 CheckTimes=0 ParticipantID=VUT VelAbs=9 CheckLogic=4 TimeThreshold=3.00 Type=6 CheckTimes=0 ParticipantID=VUT VelAbs=11 CheckLogic=0 TimeThreshold=3.00


# Phase definitions

Phase ID=1 TypeID=8 PosType=5 RoadID=1001 LaneID=-3 Offset=0.50 S=10.00 Heading=0 HeadingType=1 InitVel=0 InitAcc=0 MoveAlongTrack=1 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=1 FollowingPhaseID=2

Phase ID=2 TypeID=4 PlanType=20 MaxAccUnsigned=2.00 VelEndAbsOrRel=0 VelEnd=20.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=3

Phase ID=3 TypeID=7 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=3 FollowingPhaseID=4

Phase ID=4 TypeID=6 PlanType=12 MaxLatAccUnsigned=0.90 AbsOrRel=0 LatOffset=-3.70 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=5

Phase ID=5 TypeID=4 PlanType=20 MaxAccUnsigned=1.00 VelEndAbsOrRel=0 VelEnd=10.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=6

Phase ID=6 TypeID=7 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=4 FollowingPhaseID=7

Phase ID=7 TypeID=4 PlanType=20 MaxAccUnsigned=1.00 VelEndAbsOrRel=0 VelEnd=0.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=0
```

Table 60: example iSCAML file: lane change

In the following a detailed description is added that explains the resulting behaviour of this participant *TSV1*:

Scenario handling starts by using the trajectory segment of the Phase with the ID=1:

1. Phase with ID=1 describes a trajectory of type *SetInitCondition*. The parametrized OpenDrive co-ordinate is used as the desired position for *TSV1*.
   This position is held with zero velocity until Event with the ID=1 (*Phase Timeout*) is active. There-fore, after 5 seconds, this Event is active and a Phase transition takes place to Phase with the ID=2.

2. This Phase describes an *Acceleration/Deceleration*. The acceleration via 2 m/s² to reach the velocity of 20 m/s will be performed. Afterwards, the used Event with the ID=2 (*Trajectory Ended*) will become active and trigger a Phase transition to Phase with the ID=3.
3. This Phase describes a *Constant Continuation*. So, *TSV1* will continue to drive with constant velocity of 20 m/s along the OpenDrive lane. When Event with the ID=3 (*Relative Distance*) is active, a Phase transition to Phase with the ID=4 occurs. This happens as soon as *TSV1* is located more than 25 m in front of the participant *VUT*.
4. Phase ID=4 describes a *LaneOffset*. *TSV1* will move to the right to perform a lane change of 3.70 m. A Phase transition is triggered after this trajectory segment is finished. The following Phase has the ID=5.
5. This Phase describes an *Acceleration/Deceleration*. The deceleration via 1 m/s² to reach the velocity of 10 m/s will be performed. Afterwards, the used Event with the ID=2 (*Trajectory Ended*) will become active and trigger a Phase transition to Phase with the ID=6.
6. This Phase describes a *Constant Continuation*. So, *TSV1* will continue to drive with constant 10 m/s along the OpenDrive lane. When Event with the ID=4 is active, a Phase transition to Phase with the ID=7 occurs. Event with ID=4 uses a combination of two event methods. Both methods must be fulfilled to set the Event active. This happens if the participant *VUT* holds a velocity between 9 to 11 m/s for more than 3 s.
7. This Phase describes an *Acceleration/Deceleration*. The deceleration via 1 m/s² to reach the velocity of 0 m/s will be performed. Afterwards, the used Event with the ID=2 (*Trajectory Ended*) will become active and trigger a Phase transition to Phase with the ID=0. This transition will end the scenario execution for the *TSV1*.

Remark:
As mentioned in 1.5.1, if a trajectory segment ends automatically (e.g. a limited acceleration trajectory), method *Constant Continuation* (see 1.5.1.2) will be used until an according Event triggers a Phase transition.
Therefore, one could wonder why Phase ID=3 exists at all: After Phase ID=2 is finished, an automatic transition to Phase ID=3 takes place, which is a Constant Continuation itself.
A suggestion could be to substitute the two lines of Phase ID=2 and ID=3 by the following line (of course, IDs of following Phase lines must be adapted):

```
Phase  ID=2  TypeID=4  PlanType=20  MaxAccUnsigned=2.00  VelEndAbsOrRel=0  VelEnd=20.00  Reserved=0  VerdictLevel=0
       ActionBits=0 Reserved=0 ReactToEventID=3 FollowingPhaseID=3
```

It is important to mention that this could lead to a different behavior:
Using the original description, Event ID=3 will be checked for transition triggering after the acceleration segment.
Using the adapted version, this Event will be already checked by the start of the acceleration segment.

2.2 Multiple Parallel Events

As mentioned in 1.5.2, it is possible to check for multiple Events in parallel during a Phase.

Imagine following scenario:

A TSV drives along a lane, using an OpenDrive map. After an acceleration phase, the TSV should drive with constant speed until a faster VUT comes from behind and the distance of both vehicles falls below a certain limit. This should trigger a sharp braking maneuver of the TSV till it reaches stand still. If the VUT can't approach to fulfil this condition, the TSV should decelerate moderately after a certain distance to reach stand still. The according iSCAML file is shown in Table 61. The file could have a name like *iScenarioData_TSV1*. Then, the VehicleID of the participant that uses this file would be *TSV1*.

Object definitions

Object ID=1 VehicleID=TSV1 ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0

Object ID=2 VehicleID=VUT ObjectCategory=0 Mass=0 CenterX=0.8 CenterY=0.0 CenterZ=0.8 DimensionW=2.0 DimensionL=5.0 DimensionH=1.8 PosType=-1 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0 Reserved=0

#GeoReference definitions

GeoReference +proj=tmerc +lon_0=7.40825 +lat_0=49.21814 +ellps=WGS84 +datum=WGS84 +no_defs +zone=32

Event definitions

Event ID=1 Type=4 CheckTimes=0 TimeThreshold=5.00 CheckLogic=4

Event ID=2 Type=5 CheckTimes=0 Reserved=0

Event ID=3 Type=9 CheckTimes=0 ParticipantID=VUT DistType=0 Val=25.00 Freespace=0 CheckLogic=0 TimeThreshold=0.00

Phase definitions

Phase ID=1 TypeID=8 PosType=5 RoadID=1001 LaneID=-3 Offset=0.50 S=10.00 Heading=0 HeadingType=1 InitVel=0 InitAcc=0 MoveAlongTrack=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=1 FollowingPhaseID=2

Phase ID=2 TypeID=4 PlanType=20 MaxAccUnsigned=2.00 VelEndAbsOrRel=0 VelEnd=10.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=3

Phase ID=3 TypeID=4 PlanType=10 VelEndAbsOrRel=0 VelEnd=0 LimitType=1 LimitVal=300 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=3 FollowingPhaseID=4 ReactToEventID=2 FollowingPhaseID=5

Phase ID=4 TypeID=4 PlanType=20 MaxAccUnsigned=7.00 VelEndAbsOrRel=0 VelEnd=0.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=0

Phase ID=5 TypeID=4 PlanType=20 MaxAccUnsigned=2.00 VelEndAbsOrRel=0 VelEnd=0.00 Reserved=0 VerdictLevel=0 ActionBits=0 Reserved=0 ReactToEventID=2 FollowingPhaseID=0

Table 61: example iSCAML file: multiple parallel events

In the following a detailed description is added that explains the resulting behaviour of this participant *TSV1*:

Scenario handling starts by using the trajectory segment of the Phase with the ID=1:

1. Phase with ID=1 describes a trajectory of type *SetInitCondition*. The parametrized OpenDrive coordinate is used as the desired position for *TSV1*.
   This position is held with zero velocity until Event with the ID=1 (*Phase Timeout*) is active. Therefore, after 5 seconds, this Event is active and a Phase transition takes place to Phase with the ID=2.
2. This Phase describes an *Acceleration/Deceleration*. The acceleration via 2 m/s² to reach the velocity of 10 m/s will be performed. Afterwards, the used Event with the ID=2 (*Trajectory Ended*) will become active and trigger a Phase transition to Phase with the ID=3.
3. This Phase describes a *Constant Velocity Distance/Time Limit*. TSV1 will continue to drive with constant velocity of 10 m/s along the OpenDrive lane. There are two different Events linked that could lead to a Phase transition:
   a. The distance between VUT and TSV1 falls below 25 m (Event ID=3) what would lead to a transition to Phase ID=4.
      Phase ID=4 describes an *Acceleration/Deceleration*. TSV will decelerate by 7.0 m/s² to reach a velocity of 0 m/s. Afterwards the scenario is finished.
   b. The trajectory segment is finished after TSV1 has reached a travelled distance of 300 m during this segment. This would lead to a transition to Phase ID=5.

Phase ID=5 describes an *Acceleration/Deceleration*. TSV will decelerate by 2.0 m/s² to reach a velocity of 0 m/s. Afterwards the scenario is finished.

3 APPENDIX

3.1 File Transfer

The iSCAML file that contains the scenario based trajectory description must be located on every particpant's (object's) specific device (see first section of chapter 1).

There exist multiple ways to transfer the content of the iSCAML file from a Control Center to the specific device of a participant:

1. Transfer of a complete iSCAML file via FTP from the Control Center to the object.
2. Sending of a specific ISO 22133 message that contains the information of a single iSCAML line or an ensemble of iSCAML lines. This line or these lines can either replace pre-existing lines on the participant's device or they are appended additionally to a pre-existing file.

3.2

3.2 Geometric Dependencies

In Figure 5 the rear axle center point (RAC), the participant reference position according to OpenSCENARIO 1.0 (PRP, equals RAC on street level), the geometric center (GC, OpenSCENARIO 1.0), the point of interest (POI) are shown, the bounding box as well as the lever arms between these points. In Table 62 the configuration owner as well as the starting and ending point for the lever arms are defined. The same is given in Table 63 for the bounding box.

Lever Arm Name	From	To	Frame	Config Owner
LA_Center	PRP	GC	UKS	iSCAML
LA_RAC	NavDev	RAC	NavDev	NavDev
LA_POI	RAC	POI	UKS	iSCAML

Table 62: Lever Arms between points

Bounding Box	From	To	Config Owner
DimensionL / 2	GC	Front Bound	iSCAML
DimensionL / 2	GC	Rear Bound	iSCAML
DimensionW / 2	GC	Left Bound	iSCAML
DimensionW / 2	GC	Right Bound	iSCAML
DimensionH / 2	GC	Upper Bound	iSCAML
DimensionH / 2	GC	Lower Bound	iSCAML

Table 63: Bounding Box relations

NavDev: Navigation Device
GC: Geometrical Center
POI: Point of Interest e.g. controller point
RAC: Rear Axle Center
PRP: Participant Reference Position
PRP = RAC on Street Level for Vehicle

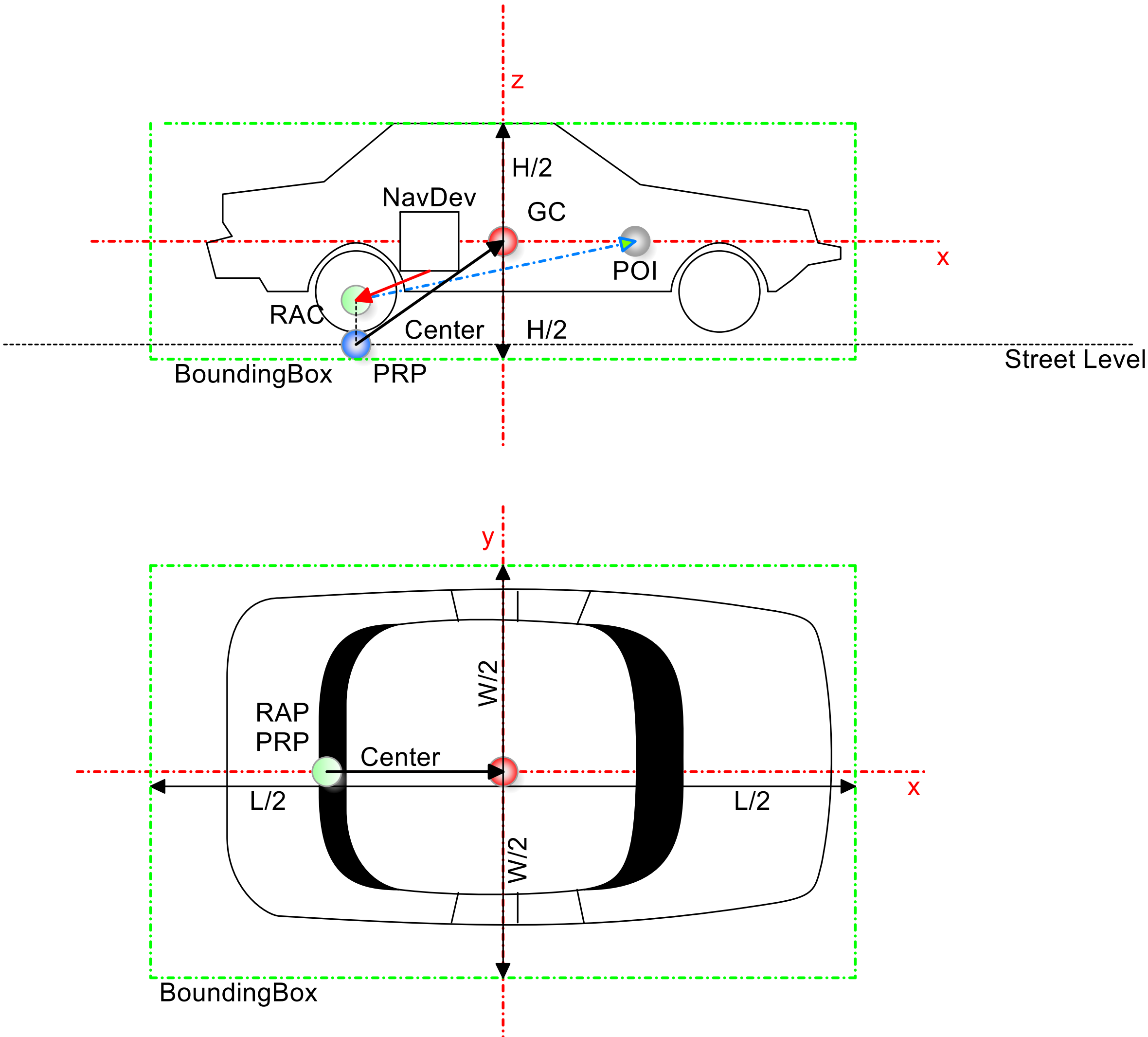

Figure 5: Geometric Considerations

3.3 Usage and Invention of iSCAML

Author of the iSCAML Specification:

iMAR Navigation GmbH
Im Reihersbruch 3
D-66386 St. Ingbert

www.imar-navigation.de
sales@imar-navigation.de

iSCAML is invented by iMAR Navigation GmbH.

It is provided to be used within the ISO 22133 standard.

The usage of iSCAML is free of license for all uses.

The users are not allowed to sell iSCAML to any third party.

iMAR does not take any liability for the proper operation, complete description or usability of iSCAML.