AgentInterface Class Reference

abstract

Agent Interface within the openPASS framework. More...

```
#include <agentInterface.h>
```

Public Member Functions

	AgentInterface (const AgentInterface &)=delete
	AgentInterface (AgentInterface &&)=delete
AgentInterface &	**operator=** (const AgentInterface &)=delete
AgentInterface &	**operator=** (AgentInterface &&)=delete
virtual int	**GetAgentId** () const =0
virtual int	**GetSpawnTime** () const =0
virtual AgentVehicleType	**GetVehicleType** () const =0
virtual double	**GetPositionX** () const =0
virtual double	**GetPositionY** () const =0
virtual double	**GetWidth** () const =0
virtual double	**GetLength** () const =0
virtual double	**GetHeight** () const =0
virtual double	**GetVelocityX** () const =0
virtual double	**GetVelocityY** () const =0
virtual double	**GetDistanceCOGtoFrontAxle** () const =0
virtual double	**GetWeight** () const =0
virtual double	**GetHeightCOG** () const =0
virtual double	**GetWheelbase** () const =0
virtual double	**GetMomentInertiaRoll** () const =0
virtual double	**GetMomentInertiaPitch** () const =0
virtual double	**GetMomentInertiaYaw** () const =0
virtual double	**GetFrictionCoeff** () const =0
virtual double	**GetTrackWidth** () const =0
virtual double	**GetDistanceCOGtoLeadingEdge** () const =0
virtual double	**GetAccelerationX** () const =0
virtual double	**GetAccelerationY** () const =0
virtual double	**GetYawAngle** () const =0
virtual std::vector< int >	**GetCollisionPartners** () const =0
virtual void	**SetPositionX** (double positionX)=0
virtual void	**SetPositionY** (double positionY)=0
virtual void	**SetWidth** (double width)=0
virtual void	**SetLength** (double length)=0
virtual void	**SetHeight** (double height)=0
virtual void	**SetVelocityX** (double velocityX)=0
virtual void	**SetVelocityY** (double velocityY)=0
virtual void	**SetDistanceCOGtoFrontAxle** (double distanceCOGtoFrontAxle)=0
virtual void	**SetWeight** (double weight)=0
virtual void	**SetHeightCOG** (double heightCOG)=0

virtual void	**SetWheelbase** (double wheelbase)=0
virtual void	**SetMomentInertiaRoll** (double momentInertiaRoll)=0
virtual void	**SetMomentInertiaPitch** (double momentInertiaPitch)=0
virtual void	**SetMomentInertiaYaw** (double momentInertiaYaw)=0
virtual void	**SetFrictionCoeff** (double frictionCoeff)=0
virtual void	**SetTrackWidth** (double trackWidth)=0
virtual void	**SetDistanceCOGtoLeadingEdge** (double distanceCOGtoLeadingEdge)=0
virtual void	**SetAccelerationX** (double accelerationX)=0
virtual void	**SetAccelerationY** (double accelerationY)=0
virtual void	**SetYawAngle** (double yawAngle)=0
virtual void	**RemoveAgent** ()=0 Requests removal of agents at next time step.
virtual void	**UpdateCollision** (int collisionPartnerId)=0
virtual bool	**Unlocate** ()=0
virtual bool	**Locate** ()=0
virtual void	**SetBrakeLight** (bool brakeLightStatus)=0
virtual bool	**GetBrakeLight** ()=0
virtual bool	**InitAgentParameter** (int id, int agentTypeId, int spawnTime, const AgentSpawnItem *agentSpawnItem, const SpawnItemParameterInterface &spawnItemParameter)=0
virtual bool	**IsValid** () const =0
virtual int	**GetAgentTypeId** () const =0
virtual void	**SetIndicatorState** (IndicatorState indicatorState)=0 Sets to Inidcator in a specific state.
virtual IndicatorState	**GetIndicatorState** ()=0 Retrieve the state of the indicator.
virtual int	**GetAgentLaneId** () const =0 Retrieve the lane ID of the agent.
virtual int	**GetAgentLaneIdLeft** ()=0 Retrieve the lane ID left of the agent. Return INFINITY if there is no lane.
virtual int	**GetAgentLaneIdRight** ()=0 Retrieve the lane ID right of the agent. Return INFINITY if there is no lane.
virtual int	**GetAgentLaneNumber** ()=0 Retrieves the lane number of the agent.
virtual bool	**IsAgentInWorld** ()=0
virtual void	**ReinitCarInQueue** ()=0 The Vehicle will be reset in queue.
virtual bool	**IsAgentAtEndOfRoad** ()=0
virtual void	**ReenterAgentAtStart** ()=0
virtual void	**SetPosition** (Position pos)=0
virtual double	**GetDistanceToStartOfRoad** () const =0
virtual Position	**GetPositionByDistance** (double distance) const =0
virtual double	**GetLaneWidth** ()=0
virtual double	**GetLaneWidthLeft** ()=0
virtual double	**GetLaneWidthRight** ()=0
virtual double	**GetCurvature** ()=0
virtual double	**GetCurvatureInDistance** (double distance)=0

virtual bool	**IsSpecialAgent** () const =0
virtual double	**GetDistanceToFrontAgent** (int laneId)=0
virtual double	**GetDistanceToRearAgent** (int laneId)=0
virtual **AgentInterface** *	**GetAgentInFront** (int laneId) const =0
virtual **AgentInterface** *	**GetAgentBehind** (int laneId) const =0
virtual double	**GetDistanceToAgent** (**AgentInterface** *otherAgent)=0
virtual void	**RemoveSpecialAgentMarker** ()=0
virtual void	**SetSpecialAgentMarker** ()=0
virtual bool	**ExistLaneLeft** ()=0
virtual bool	**ExistLaneRight** ()=0
virtual void	**SetObstacleFlag** ()=0
virtual double	**GetVelocityLateral** ()=0
virtual void	**GetAgentsDirectlyInFront** (double PeripheralPreviewDistance, **AgentInterface** *&agentFront, **AgentInterface** *&agentFrontLeft, **AgentInterface** *&agentFrontRight)=0
virtual double	**GetDistanceToSpecialAgent** ()=0
virtual bool	**IsObstacle** ()=0
virtual double	**GetDistanceToEndOfLane** (double sightDistance) const =0
virtual bool	**PerceiveMinimumSpeedOfPlatoonInLaneLeft** (double MesoscopicPreviewDistance, int &iLane, double &laneSpeedDifferential) const =0
virtual bool	**PerceiveMinimumSpeedOfPlatoonInLaneRight** (double MesoscopicPreviewDistance, int &iLane, double &laneSpeedDifferential) const =0
virtual void	**ObtainGroundTruthObjectLaneExistences** (AreaOfInterest aoi, **AgentInterface** *&agentAOI, bool &hasRightLane, bool &hasLeftLane, double PreviewDistance, double _carLengthEffective)=0
virtual double	**GetLaneDepartureFromLeftLaneBoundary** ()=0
virtual double	**GetLaneDepartureFromRightLaneBoundary** ()=0
virtual double	**GetVelocityAbsolute** ()=0
virtual void	**SetCarInfo** (CarInfo *carInfo)=0
virtual CarInfo *	**GetCarInfo** () const =0
virtual double	**GetDistanceToEndOfRamp** (int laneId)=0
virtual double	**GetPositionLateral** () const =0
virtual void	**SetCarInfoExtra** (void *extraInfo)=0
virtual void *	**GetCarInfoExtra** ()=0
virtual void	**AssignCarInfo** (double accSensDist)=0
virtual void	**AssignCarInfoExtra** ()=0
virtual CarInfo *	**GenerateCarInfo** ()=0
virtual double	**GetDistanceFrontAgentToEgo** ()=0
virtual bool	**HasTwoLeftLanes** ()=0
virtual bool	**HasTwoRightLanes** ()=0
virtual LaneChangeState	**EstimateLaneChangeState** (double thresholdLooming)=0
virtual std::list< **AgentInterface** * >	**GetAllAgentsInLane** (int laneID, double minDistance, double maxDistance, double AccSensDist)=0
virtual bool	**IsBicycle** () const =0
virtual double	**GetLaneDirection** () const =0
virtual void	**Unregister** () const =0
virtual bool	**IsFirstCarInLane** () const =0

Detailed Description

Agent Interface within the openPASS framework.

This interface provides access to agent parameters, properties, attributes and dynamic states.

Member Function Documentation

virtual void AgentInterface::AssignCarInfo (double accSensDist)

pure virtual

Sets a carInfo of the current car state with the acceleration sense distance.

Returns

virtual void AgentInterface::AssignCarInfoExtra ( )

pure virtual

Sets a carInfoExtra of the current car state.

Returns

virtual LaneChangeState AgentInterface::EstimateLaneChangeState (double thresholdLooming)

pure virtual

Retrieve an estimated lane change state.

Returns

virtual bool AgentInterface::ExistLaneLeft ( )

pure virtual

Returns true if a left lane exists.

Returns

virtual bool AgentInterface::ExistLaneRight ( )

pure virtual

Returns true if a right lane exists.

Returns

virtual CarInfo* AgentInterface::GenerateCarInfo ( )

pure virtual

Generates a CarInfo of current state of the car.

Returns

virtual double AgentInterface::GetAccelerationX ( ) const

pure virtual

Retrieves forward acceleration of agent

virtual double AgentInterface::GetAccelerationY ( ) const

pure virtual

Retrieves sideward acceleration of agent

virtual AgentInterface* AgentInterface::GetAgentBehind (int laneId) const

pure virtual

Returns the AgentInterface of the next agent behind in a specific lane.

Returns

virtual int AgentInterface::GetAgentId ( ) const

pure virtual

Retrieves id of agent

virtual AgentInterface* AgentInterface::GetAgentInFront (int laneId) const

pure virtual

Returns the AgentInterface of the next agent in front in a specific lane.

Returns

virtual void AgentInterface::GetAgentsDirectlyInFront (double PeripheralPreviewDistance, AgentInterface *& agentFront, AgentInterface *& agentFrontLeft, AgentInterface *& agentFrontRight)

```
virtual void AgentInterface::GetAgentsDirectlyInFront ( double            PeripheralPreviewDistance,
                                                        AgentInterface *& agentFront,
                                                        AgentInterface *& agentFrontLeft,
                                                        AgentInterface *& agentFrontRight
                                                      )
```

pure virtual

Retrieve a pointer to an agentInterface of the next agents in front in sight.

Returns

virtual int AgentInterface::GetAgentTypeId ( ) const

pure virtual

Retrieves type of agent

Returns

Id of agent type

```
virtual std::list<AgentInterface*> AgentInterface::GetAllAgentsInLane ( int     laneID,
                                                                        double  minDistance,
                                                                        double  maxDistance,
                                                                        double  AccSensDist
                                                                      )
```

pure virtual

Get a list of all agents in a lane.

Returns

```
virtual bool AgentInterface::GetBrakeLight ( )
```

pure virtual

Returns the status of the brake light.

```
virtual CarInfo* AgentInterface::GetCarInfo ( ) const
```

pure virtual

Returns the internal CarInfo object.

Returns

```
virtual void* AgentInterface::GetCarInfoExtra ( )
```

pure virtual

Returns the extra information of the car.

Returns

```
virtual std::vector<int> AgentInterface::GetCollisionPartners ( ) const
```

pure virtual

Retrieves list of collisions partners of agent.

```
virtual double AgentInterface::GetCurvature ( )
```

pure virtual

Returns the curvature of a lane an agent is on.

Returns

```
virtual double AgentInterface::GetCurvatureInDistance ( double distance )
```

pure virtual

Returns the curvature of a lane an agent is on in a distance.

Returns

```
virtual double AgentInterface::GetDistanceCOGtoFrontAxle ( ) const
```

pure virtual

Retrieves distance from COG to front axle of agent

virtual double AgentInterface::GetDistanceCOGtoLeadingEdge ( ) const

pure virtual

Retrieves distance from COG to leading edge

virtual double AgentInterface::GetDistanceFrontAgentToEgo ( )

pure virtual

Returns the distance of the front agent to ego.

Returns

virtual double AgentInterface::GetDistanceToAgent (AgentInterface * otherAgent)

pure virtual

Returns the distance to another agent. (negative if other agent is behind)

Returns

virtual double AgentInterface::GetDistanceToEndOfLane (double sightDistance) const

pure virtual

Returns the distance to the end of the lane or the sightDistance, if end of lane is far away.

Returns

virtual double AgentInterface::GetDistanceToEndOfRamp (int laneId)

pure virtual

Returns the distance to the end of the ramp

Returns

virtual double AgentInterface::GetDistanceToFrontAgent (int laneId)

pure virtual

Returns the distance to the next agent in front in a specific lane.

Returns

virtual double AgentInterface::GetDistanceToRearAgent (int laneId)

pure virtual

Returns the distance to the next agent behind in a specific lane.

Returns

virtual double AgentInterface::GetDistanceToSpecialAgent ( )

pure virtual

Retrieve the distance to the special vehicle.

Returns

virtual double AgentInterface::GetDistanceToStartOfRoad ( ) const

pure virtual

Retrieve the distance to the start of the road.

Returns

virtual double AgentInterface::GetFrictionCoeff ( ) const

pure virtual

Retrieves friction coefficient

virtual double AgentInterface::GetHeight ( ) const

pure virtual

Retrieves height of agent boundary box

virtual double AgentInterface::GetHeightCOG ( ) const

pure virtual

Retrieves distance from ground to COG of agent

virtual double AgentInterface::GetLaneDepartureFromLeftLaneBoundary ( )

pure virtual

Retrieve the lane departure from the left lane boundary.

Returns

virtual double AgentInterface::GetLaneDepartureFromRightLaneBoundary ( )

pure virtual

Retrieve the lane departure from the right lane boundary.

Returns

virtual double AgentInterface::GetLaneDirection ( ) const

pure virtual

Returns the current direction angle of the lane.

Returns

virtual double AgentInterface::GetLaneWidth ( )

pure virtual

Returns the width of a lane an agent is on.

Returns

virtual double AgentInterface::GetLaneWidthLeft ( )

pure virtual

Returns the width of a lane left of the agent.

Returns

virtual double AgentInterface::GetLaneWidthRight ( )

pure virtual

Returns the width of a lane right of the agent.

Returns

virtual double AgentInterface::GetLength ( ) const

pure virtual

Retrieves length of agent boundary box

virtual double AgentInterface::GetMomentInertiaPitch ( ) const

pure virtual

Retrieves moment of inertia (pitch axis)

virtual double AgentInterface::GetMomentInertiaRoll ( ) const

pure virtual

Retrieves moment of inertia (roll axis)

virtual double AgentInterface::GetMomentInertiaYaw ( ) const

pure virtual

Retrieves moment of inertia (yaw axis)

virtual Position AgentInterface::GetPositionByDistance (double distance) const

pure virtual

Returns a Position of an agent calculated by the distance from start.

Returns

virtual double AgentInterface::GetPositionLateral ( ) const

pure virtual

Returns the lateral position.

Returns

virtual double AgentInterface::GetPositionX ( ) const

pure virtual

Retrieves x-coordinate of agent.

virtual double AgentInterface::GetPositionY ( ) const

pure virtual

Retrieves y-coordinate of agent

virtual int AgentInterface::GetSpawnTime ( ) const

pure virtual

Retrieves time of spawn event of this agent

Returns

Spawn time

virtual double AgentInterface::GetTrackWidth ( ) const

pure virtual

Retrieves distance between wheels on the same axle

virtual AgentVehicleType AgentInterface::GetVehicleType ( ) const

pure virtual

Retrieves type of vehicle of agent

virtual double AgentInterface::GetVelocityAbsolute ( )

pure virtual

Retrieve the absolute velocity.

Returns

virtual double AgentInterface::GetVelocityLateral ( )

pure virtual

Retrieve the part-velocity acting perpendicular to the road direction

Returns

virtual double AgentInterface::GetVelocityX ( ) const

pure virtual

Retrieves forward velocity of agent

virtual double AgentInterface::GetVelocityY ( ) const

pure virtual

Retrieves sideward velocity of agent

virtual double AgentInterface::GetWeight ( ) const

pure virtual

Retrieves weight of agent

virtual double AgentInterface::GetWheelbase ( ) const

pure virtual

Retrieves distance between the centers of the front and rear wheels

virtual double AgentInterface::GetWidth ( ) const

pure virtual

Retrieves width of agent boundary box

virtual double AgentInterface::GetYawAngle ( ) const

pure virtual

Retrieves yaw angle of agent.

virtual bool AgentInterface::HasTwoLeftLanes ( )

pure virtual

Checks whether the agent has two lanes on the left.

Returns

virtual bool AgentInterface::HasTwoRightLanes ( )

pure virtual

Checks whether the agent has two lanes on the right.

Returns

```
virtual bool
AgentInterface::InitAgentParameter   ( int                                  id,
                                       int                                  agentTypeId,
                                       int                                  spawnTime,
                                       const AgentSpawnItem *               agentSpawnItem,
                                       const SpawnItemParameterInterface &  spawnItemParameter
                                     )
```

pure virtual

Inits all physical and world specific parameters of an agent.

virtual bool AgentInterface::IsAgentAtEndOfRoad ( )

pure virtual

Returns true if agent is agent is at end of road or near the end.

Returns

virtual bool AgentInterface::IsAgentInWorld ( )

pure virtual

Returns true if agent is still in World located.

Returns

virtual bool AgentInterface::IsBicycle ( ) const

pure virtual

Returns true if agent is a bicycle.

Returns

virtual bool AgentInterface::IsFirstCarInLane ( ) const

pure virtual

Returns true if agent is the first car in lane.

Returns

virtual bool AgentInterface::IsObstacle ( )

pure virtual

Return true if obstacle flag is set in agent.

Returns

virtual bool AgentInterface::IsSpecialAgent ( ) const

pure virtual

Returns true if Agent is marked as the special vehicle.

Returns

virtual bool AgentInterface::IsValid ( ) const

pure virtual

Checks if an agent is still valid or if its marked for remove.

virtual bool AgentInterface::Locate ( )

pure virtual

Locate agent in world.

virtual void AgentInterface::ObtainGroundTruthObjectLaneExistences

```
virtual void
AgentInterface::ObtainGroundTruthObjectLaneExistences  ( AreaOfInterest    aoi,
                                                         AgentInterface *&  agentAOI,
                                                         bool &             hasRightLane,
                                                         bool &             hasLeftLane,
                                                         double             PreviewDistance,
                                                         double             _carLengthEffective
                                                       )
```

pure virtual

Obtain basic information about surroundings.

Returns

```
virtual bool
AgentInterface::PerceiveMinimumSpeedOfPlatoonInLaneLeft ( double    MesoscopicPreviewDistance,
                                                          int &     iLane,
                                                          double &  laneSpeedDifferential
                                                        )           const
```

pure virtual

Perceive the miniumum speed of the platoon on the the left lane. Returns true if the speed is successfully perceived.

Returns

```
virtual bool
AgentInterface::PerceiveMinimumSpeedOfPlatoonInLaneRight ( double    MesoscopicPreviewDistance,
                                                           int &     iLane,
                                                           double &  laneSpeedDifferential
                                                         )           const
```

pure virtual

Perceive the miniumum speed of the platoon on the the right lane. Returns true if the speed is successfully perceived.

Returns

```
virtual void AgentInterface::ReenterAgentAtStart ( )
```

pure virtual

Sets the position of an agent to the start of the road.

Returns

```
virtual void AgentInterface::RemoveSpecialAgentMarker ( )
```

pure virtual

Removes the marker of the agent which marks it special.

Returns

```
virtual void AgentInterface::SetAccelerationX ( double  accelerationX )
```

pure virtual

Sets forward acceleration of agent

Parameters

[in] **accelerationX** forward acceleration

```
virtual void AgentInterface::SetAccelerationY ( double  accelerationY )
```

pure virtual

Sets sideward acceleration of agent

Parameters

[in] **accelerationY** sideward acceleration

virtual void AgentInterface::SetBrakeLight (bool brakeLightStatus)

pure virtual

Set the brake light on or off.

virtual void AgentInterface::SetCarInfo (CarInfo * carInfo)

pure virtual

Sets the internal CarInfo object.

Returns

virtual void AgentInterface::SetCarInfoExtra (void * extraInfo)

pure virtual

Sets internal extra information for car.

Returns

virtual void AgentInterface::SetDistanceCOGtoFrontAxle (double distanceCOGtoFrontAxle)

pure virtual

Sets distance from COG to front axle of agent

Parameters

[in] **distanceCOGtoFrontAxle** distance from COG to front axle

virtual void AgentInterface::SetDistanceCOGtoLeadingEdge (double distanceCOGtoLeadingEdge)

pure virtual

Sets distance from COG to leading edge

Parameters

[in] **distanceCOGtoLeadingEdge** distance from COG to leading edge

virtual void AgentInterface::SetFrictionCoeff (double frictionCoeff)

pure virtual

Sets friction coefficient

Parameters

[in] **frictionCoeff** friction coefficient

virtual void AgentInterface::SetHeight (double height)

pure virtual

Sets height of agents boundary box

Parameters

[in] **height** Height of agent

virtual void AgentInterface::SetHeightCOG (double heightCOG)

pure virtual

Sets distance from ground to COG of agent

Parameters

`[in]` **heightCOG** distance from ground to COG

virtual void AgentInterface::SetLength (double length)

pure virtual

Sets length of agents boundary box

Parameters

`[in]` **length** Length of agent

virtual void AgentInterface::SetMomentInertiaPitch (double momentInertiaPitch)

pure virtual

Sets moment of inertia for pitch axis

Parameters

`[in]` **momentInertiaPitch** moment of inertia for pitch axis

virtual void AgentInterface::SetMomentInertiaRoll (double momentInertiaRoll)

pure virtual

Sets moment of inertia for roll axis

Parameters

`[in]` **momentInertiaRoll** moment of inertia for roll axis

virtual void AgentInterface::SetMomentInertiaYaw (double momentInertiaYaw)

pure virtual

Sets moment of inertia for yaw axis

Parameters

`[in]` **momentInertiaYaw** moment of inertia for yaw axis

virtual void AgentInterface::SetObstacleFlag ( )

pure virtual

Sets the flag to mark the agent as obstacle.

Returns

virtual void AgentInterface::SetPosition (Position pos)

pure virtual

Set the position of an agent.

Returns

virtual void AgentInterface::SetPositionX (double positionX)

pure virtual

Sets x-coordinate of agent

Parameters

[in] **positionX** X-coordinate

virtual void AgentInterface::SetPositionY (double positionY)

pure virtual

Sets y-coordinate of agent

Parameters

[in] **positionY** Y-coordinate

virtual void AgentInterface::SetSpecialAgentMarker ( )

pure virtual

Sets the marker of the agent which marks it special.

Returns

virtual void AgentInterface::SetTrackWidth (double trackWidth)

pure virtual

Sets distance between wheels on same axle

Parameters

[in] **trackWidth** distance between both front wheels (or both rear wheels)

virtual void AgentInterface::SetVelocityX (double velocityX)

pure virtual

Sets forward velocity of agent

Parameters

[in] **velocityX** Forward velocity

virtual void AgentInterface::SetVelocityY (double velocityY)

pure virtual

Sets sideward velocity of agent

Parameters

[in] **velocityY** Sideward velocity

virtual void AgentInterface::SetWeight (double weight)

pure virtual

Sets weight of agent

Parameters

[in] **weight** agents weight

virtual void AgentInterface::SetWheelbase (double wheelbase)

pure virtual

Sets distance between centers of front and rear wheels

Parameters

[in] **wheelbase** Distance between front and rear wheels.

virtual void AgentInterface::SetWidth (double width)

pure virtual

Sets width of agents boundary box

Parameters

[in] **width** Width of agent

virtual void AgentInterface::SetYawAngle (double yawAngle)

pure virtual

Sets yaw angle of agent

Parameters

[in] **yawAngle** agent orientation

virtual bool AgentInterface::Unlocate ( )

pure virtual

Unlocate agent in world.

virtual void AgentInterface::Unregister ( ) const

pure virtual

Unregisters the agent from the world.

Returns

virtual void AgentInterface::UpdateCollision (int collisionPartnerId)

pure virtual

update list with collision partners

The documentation for this class was generated from the following file:

- agentInterface.h

Generated on Fri Dec 16 2016 09:15:31 for AgentInterface by doxygen 1.8.9.1