**Diagram

Description automatically generated**

**Domain Model Documentation**

**Pedestrian**

Description: a person walking along a road.

Attributes:

* X: x coordinate position
* Y: y coordinate position
* Diameter: a pedestrian is modeled as being a circle with a diameter of .5 m.
* Velocity: the speed and direction at which the pedestrian moves along the y-axis
* Delay: Time delay before pedestrian begins moving
* Limit: Ending y-position of pedestrian

Operations:

* Move(): this function is used to model a pedestrian moving. It updates the pedestrian’s position based on the amount of time that has passed.

Relationships: A Pedestrian gets in the way of a vehicle and is detected by a vehicle’s Pedestrian Detection Sensor.

**Vehicle**

Description: a road vehicle with 4 wheels powered by an internal combustion engine or electric motor and able to carry a small number of people

Attributes:

* steadySpeed: for our modeling purposes, a vehicle has a normal steady state speed of 50 kph (13.9 m/s)
* width: for our modeling purposes, a vehicle has a width of 2 m, which marks its collision zone
* x: x coordinate position
* y: y coordinate position
* acceleration: the rate at which a vehicle’s speed is changing. For our modeling purposes a vehicle can have a maximum acceleration of .25 g and a maximum deceleration of 0.7g (1 g = 9.81 m/s^2)
* velocity: the speed and direction at which the vehicle moves along the x-axis
* failSafe: a vehicle has a fail safe mode in which the response time to reach a requested deceleration value is increased. This flag indicates whether a vehicle is in fail safe mode.

Operations:

* Move(): this function is used to model a vehicle moving. It updates the vehicle’s position based on the amount of time that has passed.

Relationships: A vehicle has a Pedestrian Detection Sensor, Safety Controller, and Brake-by-Wire Actuator.

**Pedestrian Detection Sensor**

Description: The Pedestrian Detection Sensor is a stereo camera that has pedestrian recognition and tracking capabilities.

Attributes:

* locationAccuracy: the sensor can give a pedestrian’s (x,y) location relative to the car with an accuracy of +/- .5 m.
* speedAccuracy: the sensor can give a pedestrian’s speed with an accuracy of +/- .2m/s
* directionAccuracy: the sensor can give a pedestrian’s direction with an accuracy of +/- 5 degrees

Operations:

* SendData(): this function sends a detected pedestrian’s relative location to the vehicle as well as their velocity to the Pedestrian Collision Avoidance Algorithm.
* Detect(Pedestrian): this function is responsible for detecting a pedestrian object.

Relationships: The Pedestrian Detection Sensor detects a pedestrian and sends the data about that pedestrian to the Pedestrian Collision Avoidance System.

**Safety Controller**

Description: The Safety Controller is a large system within the vehicle that is responsible for the vehicle’s safety operations.

Relationships: The Safety Controller is part of the Vehicle.

**Pedestrian Collision Avoidance System**

Description: The Pedestrian Collision Avoidance System is an algorithm that is responsible for enacting avoidance maneuvers in the event of a potential collision with a pedestrian.

Attributes:

* minDistance: the closest distance that the algorithm will allow the vehicle to get near a pedestrian
* maxDecel: the maximum deceleration of the vehicle is 0.7 g

Operations:

* CheckCollision(): this function is the meat and potatoes of the Pedestrian Collision Avoidance System. It will take in the information about a pedestrian sent from the Pedestrian Detection Sensor and determine if avoidance action needs to be taken. If action does need to be taken, the function will issue an ApplyBrakes() request.
* IsMalfunction(): function that is responsible for disabling the Pedestrian Collision Avoidance System in the event that the Pedestrian Detection Sensor malfunctions, the Brake-by-Wire Actuator malfunctions, or the data sent by the Pedestrian Detection Sensor is determine to be corrupt.
* ApplyBrakes(): this function sends a request to the Brake-by-Wire Actuator to apply a given deceleration value.
* ReleaseBrakes(): this function sends a request to the Brake-by-Wire Actuator to return to steady state velocity.

Relationships: The Pedestrian Collision Avoidance System is part of the Safety Controller of a Vehicle.

**Brake-by-Wire Actuator**

Description: The Brake-by-Wire Actuator responds to deceleration requests by interrupting the steady state velocity control and then applying brake torque via electro-mechanical actuators at all four wheels of the vehicle.

Attributes:

* decelAccuracy: for our modeling purposes, the Brake-by-Wire Actuator is able to apply a specified deceleration value within an accuracy of +/- 2%
* decelTime: it takes 200 ms to reach a requested deceleration value
* releaseTime: it takes 100 ms to get back to the acceleration to steady state value

Operations:

* SlowDown(): adjusts the vehicle’s acceleration to slow it down
* ReturnToSteady(): adjusts the vehicle’s acceleration to return it to it’s steady state velocity

Relationships: A Vehicle has a Brake-by-Wire Actuator. The Brake-by-Wire Actuator receives requests from the Pedestrian Collision Avoidance System.