Software Requirements Specification (SRS)

Automated Pedestrian Collision Avoidance System (APCA)

Authors: Team GReEN; Garret Smith, Rebecca Collins, Eric Austin, Nikhil Andrews

Customer: Mr. David Agnew, Continental Automotive Systems

Instructor: Dr. Betty Cheng, Michigan State University

1 Introduction

This software requirements specification (SRS) document provides a short overview of the documents purpose, scope, definitions, acronyms and abbreviations, and the overall organization of this document.
Section 2 gives the overall description of the APCA System. This section includes the product perspective, product function, user characteristics, constraints, assumptions and dependencies, and apportioning of requirements.
The following sections after that we provide the specific and modeling requirements of our system (Sections 3 and 4).A demonstration using a prototype; how to run the prototype and sample scenarios are also included, as well as references (Sections 5 and 6).

1.1 Purpose

The purpose of the SRS is to describe the specifics of the Automated Pedestrian Collision Avoidance system (APCA); how the system behaves, system accomplishments, specification and constraints under which the system operates. This document is mainly intended for those developers working on such systems, but in our case more specifically intended for Mr. David Agnew of Continental Automotive Systems.

1.2 Scope

The Automated Pedestrian Collision Avoidance System (APCA) is an autonomous system that is designed to avoid collisions with pedestrians. This system provides an automated control over the vehicle's braking system (BBW or Brake-By-Wire) that responds to potential collisions while minimizing loss of efficiency. The system also has control over the vehicle's acceleration in order to return to original speed after avoiding collision. To sum up, APCA is designed to provide additional safety to the driver and pedestrians.

1.3 Definitions, acronyms, and abbreviations

- **APCA** – Automated Pedestrian Collision Avoidance
- **Pedestrian Sensor** – A stereo camera which provides pedestrian recognition, relative location, speed, and relative direction with respect to the vehicle.
- **Cycle Time** – time between pedestrian sensor outputting packets. In the project overview, it is given as 100 ms.
- **Brake-by-Wire** – a sub-system that responds to deceleration requests by interrupting the steady speed (cruise control) and then applying brakes.
- **Vehicle** – the autonomous vehicle for this application
- **"Lost time"** – time difference (in seconds) between system on and system off to reach a common point beyond the pedestrian with controlled vehicle back again at steady state velocity.
- **Fail Safe** – or fail operational mode, increases the response time to reach requested decel. In the project overview, it is given that response time increases from 200 ms to 900 ms.

1.4 Organization

The second section of this document, Overall Description, gives a general idea of how the product functions. Some of the key concepts described are: interface constraints, the intended users, and assumptions.
The third section of this document, Specific Requirements, gives an enumerated list of requirements that the system fulfills.
The fourth section of this document, Modeling Requirements, explains how the software is designed to meet the requirements. In this section, diagrams (Use case, Class, and State) are used to specify how the software functions.
The fifth section, Prototype, provides a demonstration on how to use and run the prototype and includes sample scenarios of the system.
The sixth section, References, gives a list of all documents referenced.
The Last section of this document, Point of Contact, gives information on how to obtain more information regarding this document and project.

2 Overall Description

Autonomous driving is an area of great interest to the auto industry. An essential task of an automated car is to avoid collisions. This is currently performed by human drivers who respond to hazardous situations. However, an autonomous vehicle cannot rely on human intervention when threats arise. The APCA system uses a sensor to detect pedestrians and applies braking automatically if necessary to avoid a collision with the pedestrian.

2.1 Product Perspective

Describe the context for the product
Is it one element that is part of a bigger system? If so, then give a pictorial representation or diagram (e.g., data flow diagram – DFD, block diagram) that describes how your product fits.
Interface Constraints:
System interfaces
User interfaces
HW interfaces
SW interfaces
Communication interfaces
Other types of constraints:
Memory
Operations
Site adaptation operations (customization that is done on-site).

Start of your text.

2.2 Product Functions

Detection: The use of the pedestrian sensor to locate pedestrians in front of the vehicle.
Tracking: Using the scanner and algorithms, track the position of the pedestrian as it moves within the sensors scanning range.
Apply Braking: When a pedestrian steps in front of the vehicle and a collision is possible, a signal is sent to the BBW actuator to begin deceleration.
Restore Velocity: When the pedestrian threat is gone and braking has been applied, tell the vehicle to restore its previous velocity.
Fail Safe: If there is a problem with braking, the fail safe mode is applied. This increases the time allowed for braking to ensure collisions are avoided.

2.3 User Characteristics

The user in the context of this product would be considered the driver of the vehicle. However, the vehicles that will be equipped with the APCA system will be autonomous. Therefore, the driver is assumed to have little or no knowledge of the system. The driver is also expected to have no interaction with the system.

2.4 Constraints

The APCA system must have zero collisions for each scenario presented as stated in the Project Description. The system must also attempt to avoid time loss due to pedestrian avoidance. This must not interfere with safety constraints however. If a braking problem is present, the APCA system must enable the fail safe system and increase time allowed for braking. Additionally, the BBW actuator is configured to respond to commands as quickly as a human. The response time to reach the requested deceleration is 200ms and the brake release time is 100ms. Also, with the given scenarios, the vehicle will always be travelling at a steady velocity of 13.9 m/s and have an acceleration of .25g after braking is applied.

2.5 Assumptions and Dependencies

It is assumed that the APCA system is always on and scanning if the vehicle is running. It is also assumed that the pedestrian will only move at either 0 or 6 kilometer per hour, and their acceleration is infinite. Also, the pedestrian will always be moving at a 90 degree angle to the vehicles path. It is also assumed that the elements of the system are all functioning properly and no abnormal conditions are present.

2.6 Approportioning of Requirements

Based on negotiations with customers, requirements that are determined to be beyond the scope of the current project and may be addressed in future versions/releases.

Start of your text.

3 Specific Requirements

1. The system consists of a pedestrian sensor and a brake-by-wire actuator.
   a. The pedestrian sensor will be a stereo camera and will send a packet every 100 ms. The data in the packet contains the pedestrian location (+/- 0.5 m) relative to the car, and the pedestrian speed (+= 0.2 m/s) and direction (+/- 5 deg).

b. The Brake-by-Wire (BBW), when activated, interrupts the steady state velocity control (cruise control) and applies braking torque at all four wheels of the vehicle. It has a deceleration accuracy of +/- 2%, a response time of 200 to reach requested decel, and a response time of 100 ms to release. A maximum deceleration of 0.7 g (1g = 9.81 m/s^2) is possible.

2. The vehicle and the pedestrian for testing the system have the following properties:

a. The vehicle is an autonomous vehicle that has a normal steady state speed of 50 kph (13.9 m/s) and an acceleration back to steady state speed (after auto brake apply) of 0.25 g. The collision zone is based on the vehicle's width of 2 m.

b. The pedestrian for this application can be in static or in motion and have speeds of 0 kph OR 6 kph and it can be assumed that the pedestrian can change velocity in infinite acceleration. The size of the pedestrian can be considered a circle with 0.5 m diameter.

3. The system sensor should be able to recalculate the braking distance between the vehicle and pedestrian by responding appropriately.

4. The system must return to steady-state velocity after the auto braking maneuver with an acceleration of 0.25 g.

5. When the vehicle is in a potential collision zone, the system should take action immediately to brake automatically and avoid collisions.

6. Based on the scenarios described in the project overview, the system should be tested assuming that is does not know which scenario is occurring.

7. The system should be effective (Zero Collisions allowed) and efficient (minimize lost time).

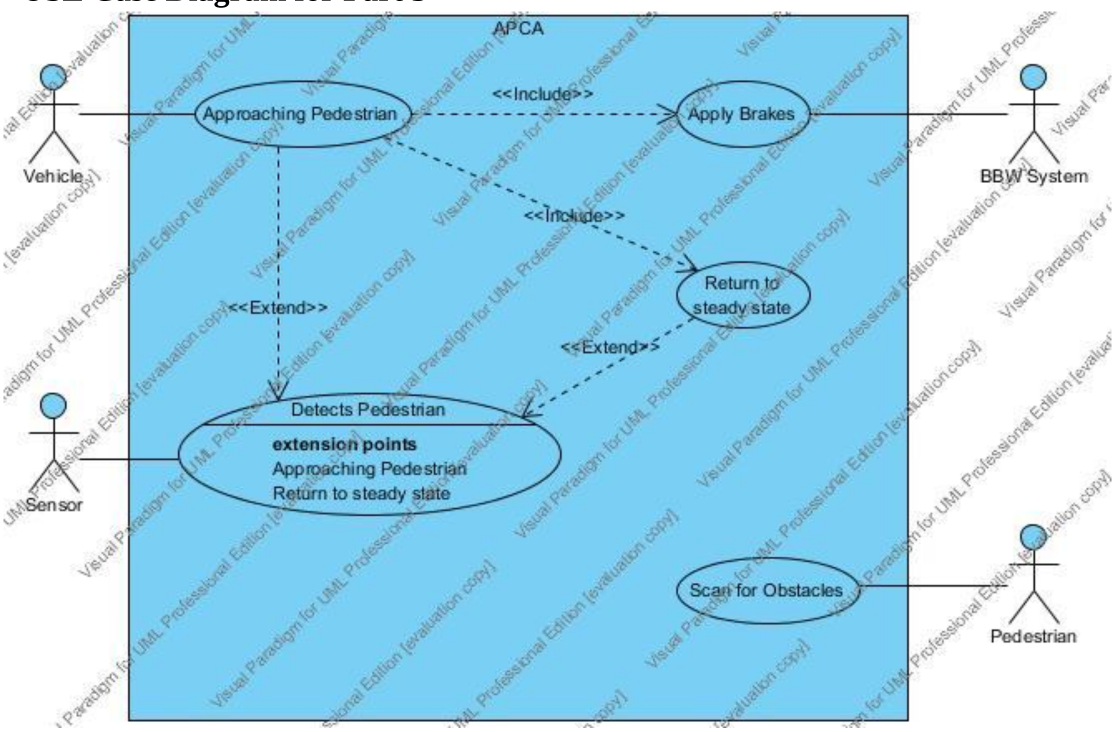
4 Modeling Requirements

This is the specification portion of the requirements document. (Specifying the bridge between the application domain and the machine domain.)
For each new diagram type introduced, describe the notation.
Give and describe use case diagrams
Use the template below to describe each use case.
Each goal may be satisfied by 1 or more use cases
Each use case should refer to 1 or more requirements (in Section 3)
Give and describe a high-level class diagram that depicts the key elements of the system
Include a data dictionary to describe each class, its attributes, its operations, and relationships between classes.
Representative Scenarios of System:
Give English descriptions of representative scenarios for each use cases.
Check: use instances of the class names from class diagram; refer to the terms used in use case diagram

For each scenario, give a corresponding sequence diagram
Check: Objects should be instances of classes in class diagram

Create and explain a state diagram for all key classes that participate in the scenarios (from above).
Check: that all scenarios can be validated against the state diagrams.
Check that the events, actions are modeled in the class diagram.
Check that all variables referenced in the diagrams are declared as attributes in the class diagram.

USE Case Diagram for Part 3



Use Case:	Detects Pedestrian
Actors:	Sensor
Type:	Primary and Essential
Description:	Sensor scans, recognizes and tracks pedestrians in front of it.
Cross-references:	1a, 3, 6, 7

Use Case:	Approaching Pedestrian
Actors:	Vehicle
Type:	Primary and essential
Description:	The Autonomous Vehicle moves in forward direction at a constant set speed approaching a pedestrian.
Includes:	Apply Brakes, Return to Steady State
Extends:	Detects Pedestrian
Cross-references:	2a, 4, 5, 6, 7

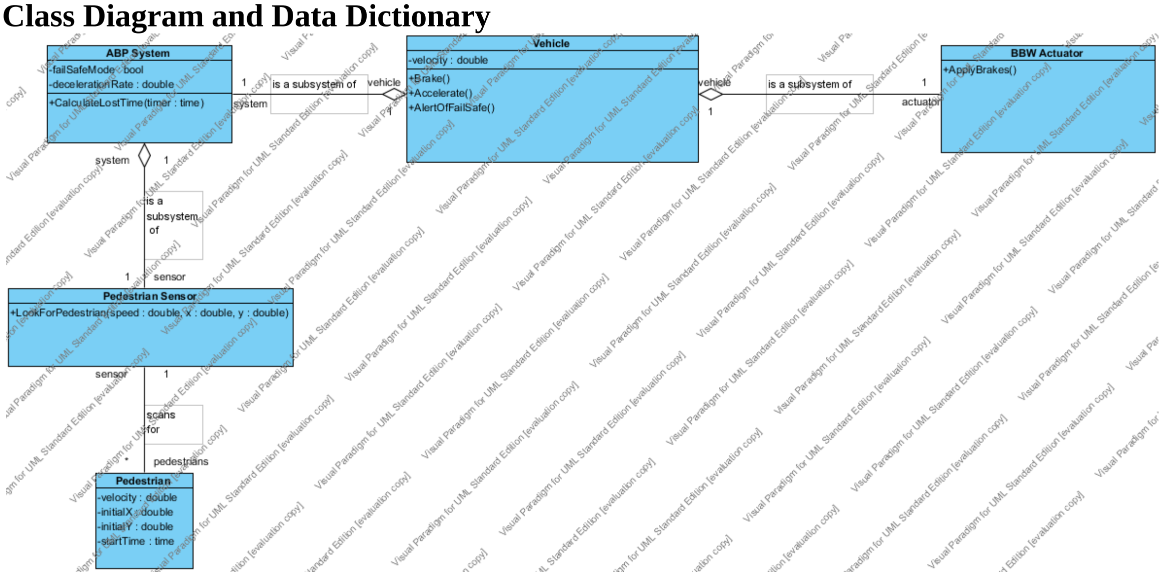
Related Use-Cases:	Activates Apply Brakes and Return to Steady State (if necessary) use cases

Use Case:	Return to Steady State
Actors:	Sensor
Type:	Primary and Essential
Description:	If sensor detects pedestrian, the BBW system applies brakes and the system then needs to return to the original steady state velocity.
Extends:	Detects Pedestrian
Cross-references:	2a, 4, 6, 7

Use Case:	Apply Brakes
Actors:	BBW
Type:	Primary and Essential
Description:	Once the pedestrian has been detected, the BBW sub-system responds to deceleration requests and automatically starts braking accordingly until collision has been avoided.
Cross-references:	1b, 3, 5, 6, 7

Use Case:	Scan for obstacles
Actors:	Pedestrian
Type:	Primary and essential
Description:	The pedestrian can be in static or in motion, can change speed with infinite acceleration. When moving, the pedestrian only moves in right angle to the vehicles path.
Cross-references:	2b, 6, 7

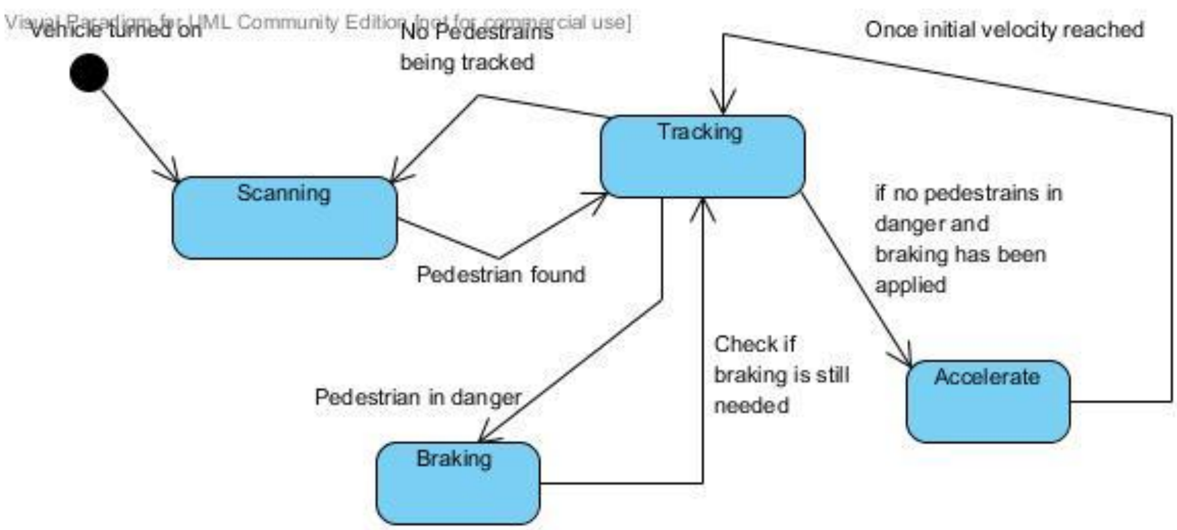
Class Diagram and Data Dictionary



Class	Attributes	Operations	Relationships
ABP System	failSafeMode – a boolean value that will show if the system is in fail safe mode or not. decelerationRate – a double value that will show how fast the vehicle can decelerate.	CalculateLostTime() – an operation that will be run to calculate how much time is lost if and when the vehicle decelerates to avoid a pedestrian.	The ABP System is a subsystem of the Vehicle.
Vehicle	velocity – a double value that determines the velocity of the vehicle.	Brake() – an operation that will be called if the car needs to avoid a pedestrian. Accelerate() – an operation that will be called to bring the vehicle back to its previous speed to cut down on lost time. AlertOfFailSafe() – If fail safe mode is activated, this operation will alert the driver that the car should be taken in for repair.	The Vehicle contains both the ABP System and the BBW Actuator and uses them to detect and avoid possible pedestrian collisions.
Pedestrian Sensor		LookForPedestrian() – an operation by the sensor that will constantly scan for pedestrians that the vehicle may collide with.	The Pedestrian Sensor is a subsystem of the ABP System. It scans for Pedestrians that the car may potentially collide with.
Pedestrian	velocity – the velocity of the pedestrian. initial X – the starting point of the pedestrian in regards to the X axis. initial Y – the starting point of the pedestrian in regards to the Y axis. startTime – the time at which the pedestrian will start		The Pedestrian triggers the Pedestrian Sensor.

	walking across the street (if they do).		
BBW Actuator		ApplyBrakes() – If the vehicle decides to brake, the BBW Actuator will use this operation to slow the vehicle.	The BBW Actuator is a subsystem of the Vehicle.

Below is a state diagram for the APCA system. The system constantly scans for pedestrians as soon as the vehicle is turned on. For each new pedestrian found, it begins tracking. When a pedestrian steps in front of the car, it is seen as being in danger of collision. A signal is then sent to the BBW actuator and braking is applied. As soon as the pedestrian is out of danger, acceleration begins to bring the vehicle up to speed. During acceleration, the tracking module makes sure no pedestrians enter a collision course.



5 Prototype

Our prototype consists of 10 defined scenario buttons, and a chart that gives the details of each scenario. It also contains a form that users can input their own custom variables. At the bottom of the web page is a results box.

5.1 How to Run Prototype

To run our prototype, all you need is a web browser that supports JavaScript.

What system configuration? (Should be accessible through web.) Are there plugins? Are there any OS or networking constraints. Give the URL for the prototype.

5.2 Sample Scenarios

Our Prototype currently does not execute, but following is an example of how we plan on it working. The customer arrives to our webpage and is presented with a table containing all the information of the predefined scenarios we are supposed to test for. Beneath the table are 10 buttons, one for each predefined scenario. Upon clicking the button, we will run our algorithm on the variables to determine if the pedestrian is hit or not, and what the loss of time for each scenario will be. This information will be displayed in the "Results" box at the bottom of the page. There is also a form that allows for custom scenarios to be run. The customer will be able to plug in numbers to the text boxes and click the "Compute Scenario" button to test the system under any conditions.

6 References

Provide list of all documents referenced in the SRS
Identify each document by title, report number, date, and publishing organization.
Specify the sources from which the references can be obtained.
Include an entry for your project website.

Start of your text.

[1] D. Thakore and S. Biswas, "Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks," Proceedings of IEEE Military Communication, Atlantic City, October 2005.

7 Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University (chengb at cse.msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.