Sleep Detection System for Vehicle Drivers

Case Study Analysis

Shashi Shivaraju

Srihith Alwala

CPSC 8750

4/25/19

Table of Contents

List of Figures

Acknowledgement

The success and outcome of this case study required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this case study.

We thank Dr. John D. McGregor, for providing us an opportunity to do this case study in Software Architecture and giving us all support and guidance, which made us complete the case study duly. We are extremely thankful to Mr. Vimal Vakeel for providing this project of Sleep detection system for Vehicle Drivers.

Abstract

This report documents the results of applying the Architecture Analysis and Design Language (AADL) in designing and developing a sleep detection system for vehicle drivers. While driving fatigue and microsleep at the wheel are often the cause of serious accidents. However, these initial signs of fatigue can be detected before a critical situation arises. The proposed sleep detection system for drivers can do this by tracking eye movements using sensors like camera and infrared sensors [1].

1 Introduction

This document presents the results of a case study of the application of the Architecture Analysis and Design Language (AADL) to the Sleep Detection System for Vehicle Drivers.

Model-based software design is the application of model-based engineering techniques (i.e., the use of models and abstractions to perform typical engineering tasks) to design, verify and validate software. Model-based software design relies on analytical practices using analysis and modelling languages and supporting tools. AADL is a Society of Automotive Engineers (SAE) standard for predictable model-based engineering of real-time and embedded computer systems which was first released in 2003.[2]

The AADL and its supporting tools such as the Open Source AADL Tool Environment (OSATE) [SEI 2010][3] have been designed to design and capture the architecture of embedded software systems in terms of the application software as a runtime architecture deployed on a computer system. This allows the software architect to develop a thorough understanding of and insight into (1) critical characteristics vital to a system's correct operation and (2) the impact the runtime architecture and computer system deployment on the non-functional system properties. These characteristics include considerations such as sensor/command data latency and update rates; CPU throughput; synchronous/asynchronous task management; and data-bus packet definitions and update rates. The SAE AADL industry standard for modelling and analysis of embedded software system architectures was chosen because of its ability to support analysis of non-functional properties, such as robustness, safety, performance, and security. An example for AADL model is shown in Figure 1

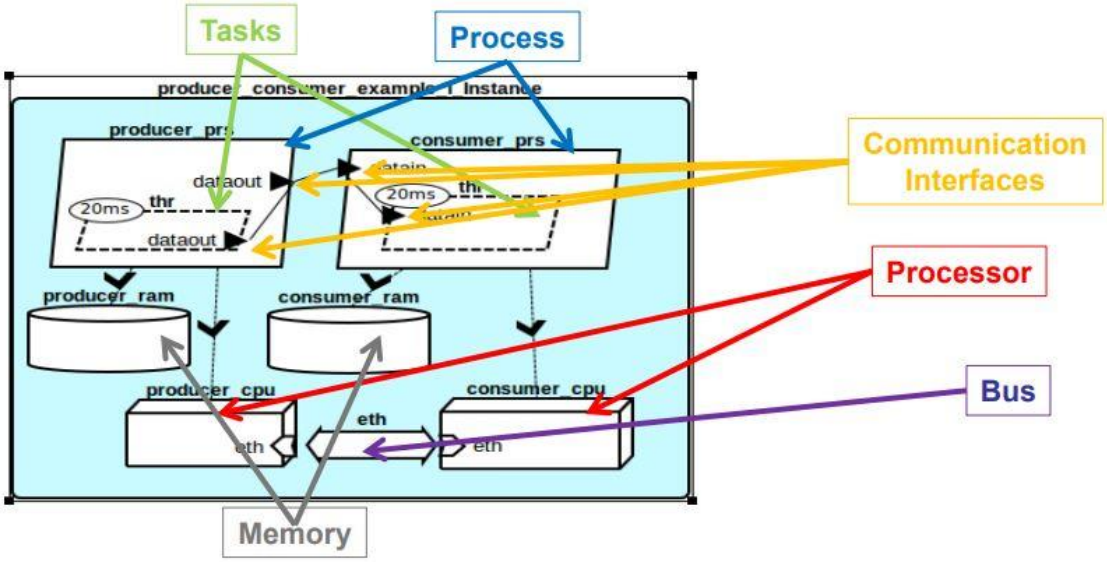

Figure 1:An example of AADL model

The model and analyses presented in this report are results of a case study effort that is applying the AADL to represent and analyse the proposed sleep detection system for vehicle drivers.

The report is organized as follows:

- Section 2 provides an overview of sleep detection system for vehicle drivers.
- Section 3 presents AADL model of sleep detection system for vehicle drivers
- Section 4 provides the architectural analysis of the provided aadl model
- Section 5 provides summary of key insights of this case study

2 Overview of Sleep Detection System for Vehicle Drivers

When driving alone on highways or driving over a long period of time, drivers are inclined to feel bored and sleepy or even fall asleep. Feeling sleepy while driving could result in hazardous traffic accident. So, sleep detection for drivers is crucial for safety of the driver and for those on the road. Particularly important for truck drivers who drive for a living and for people who prefer driving as their preferred mode of transportation. Most of the products for driver anti-sleep system sold in the market is a simplistic audio system making intermittent noises which is quite annoying and inefficient. Some of the existing solutions used in automotive industry are:

- Rest recommendation system developed by Audi
- Driver Alert system developed by Ford
- Attention Assist developed by Mercedes-Benz.

But these systems are either costly or their usage is limited by proprietary.

So, the considered system for case study proposes a cheap and efficient Sleep Detection System for Vehicle Drivers which makes use of use of Kinect detector which has camera sensors with real time image processing and Infrared sensors. The data from sensors are used for eye-lid distance tracking algorithm in order to detect sleepiness and trigger a sound alarm along with flashing LED to notify the user. The functional architecture of the system is shown as a system level block diagram in Figure 2

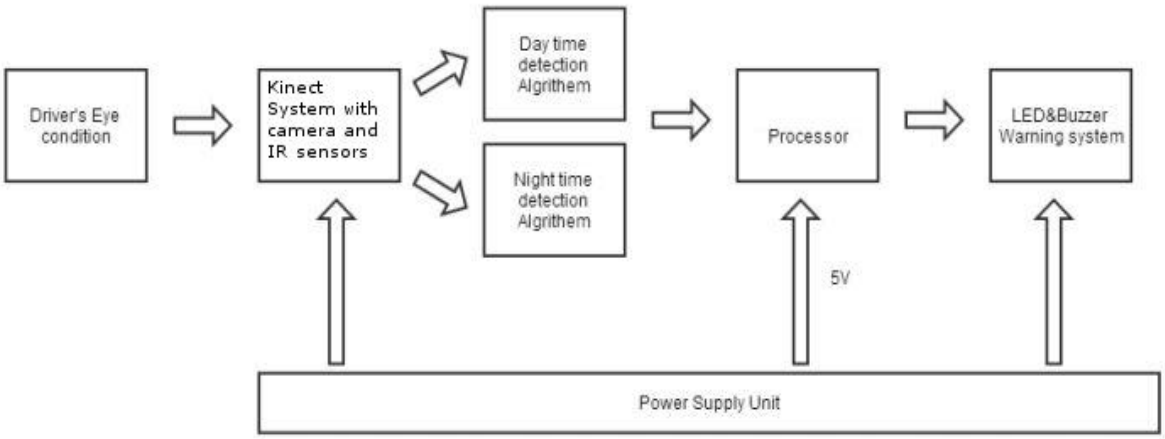

Figure 2: Functional Architecture of the system

The Kinect detector is used to monitor the driver by capturing face images and projecting infrared waves. The data from the camera is provided to a day time detection algorithm and data from IR sensor is provided to night time detection algorithm. The system consists of a processor and alarm system consisting of LEDs and audio outputs. The system is powered by external power supply unit.

Few of the salient functional features of the system are listed below:

- Eye feature extraction to determine sleepiness.
- Real time image processing and eye tracking.
- Sound and flashing LED alarm system to draw driver's attention.
- Minimalistic interference and potential hazard to users driving.
- Portable and compact system with external power supply inlet.

3 AADL model Description of Sleep Detection System for Vehicle Drivers

The AADL model provided for case study is designed using a **Pipeline architecture** and consists of a system named “AntiSleepSystem” with the below subcomponents:

- A Process named “Controller” which has an in-event port and 3 out-vent ports.
- A Processor named “SystemProcessor” with memory sub-component “RAM” and bus access.
- A Device named “KinectDetector” with out-event port and bus access.
- A Device named “AudioSystem” with in-event port and bus access
- A Device named “LEDDisplay” with in-event port and bus access
- A Device named “DomeLights” with in-event port and bus access
- A Bus named “HardwareConnection” which connects all the devices and the processor.

The AADL model of the system is as shown in Figure 3

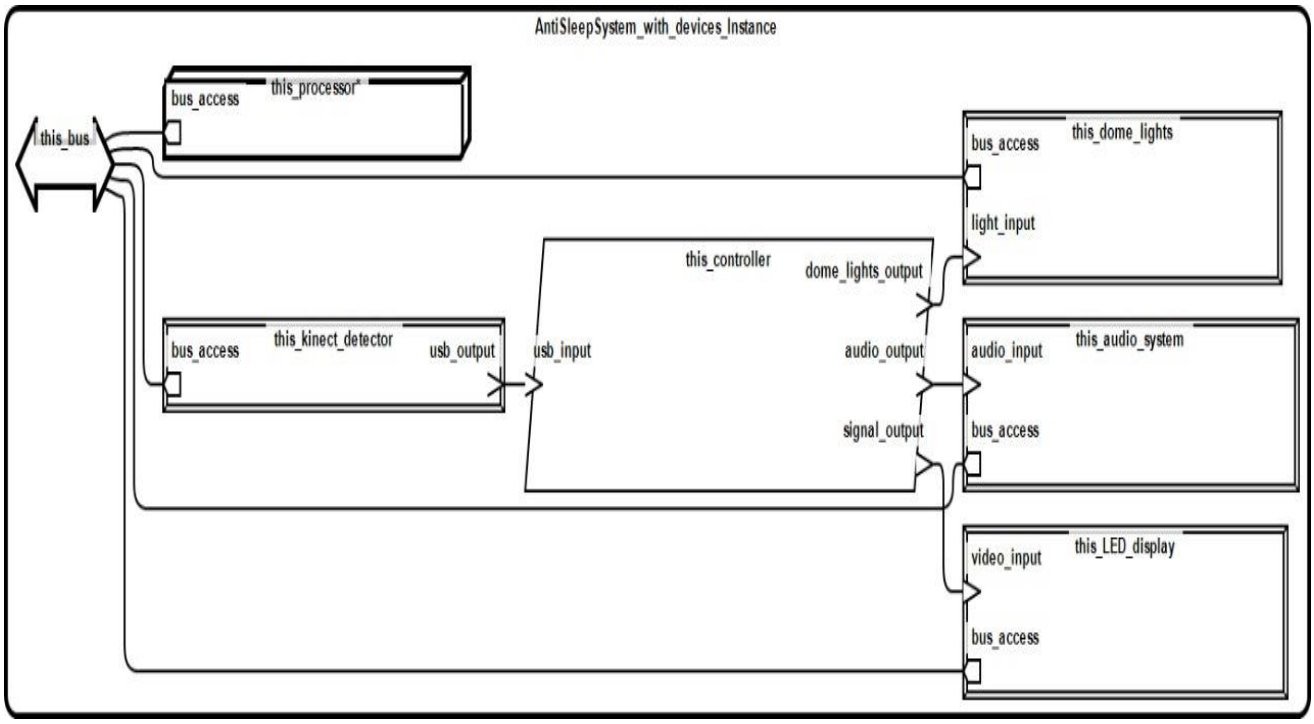

Figure 3 AADL Model Architecture

Please refer the appendix which includes the detailed aadl texture model of the system. From the given aadl model it appears that the flow in the system occurs from the flow source device “KinectDetector” whose output is provided as input to the process “Controller”. Based on the input data, the process “Controller” decides to operate the devices “DomeLights”, “AudioSystem”, “LEDDisplay” to provide alarm notification to the vehicle driver.

4 Architecture Analysis

As part of this case study, we performed several analyses supported by AADL as listed below:

- Mass and weight analysis:

  Below is the result of the mass analysis done through osate analysis plugin;

Weight totals Report
Warning! this_bus: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_kinect_detector: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_audio_system: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_LED_display: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_dome_lights: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_memory: [L] No net weight plus subcomponent weight or no gross weight
Warning! this_processor: [L] No net weight plus subcomponent weight or no gross weight
Warning! AntiSleepSystem_with_devices_Instance: [L] No net weight plus subcomponent weight or no gross weight

Figure 4 Mass and Weight Analysis

  Since the software architect of the provided model has not assigned any mass property values to system type or system implementation and its sub-systems, the mass and weight analysis report provided warnings for each of them.

- Power requirements analysis:

  The osate plugin failed to perform this analysis since the software architect of the provided model has not assigned any power property values to the system implementation and its sub systems.

- Safety Analysis[5]:

  During the case study we tried to perform Fault-Tree Analysis(FTA) to get the dependencies between error events and failure modes, and Functional Hazard Analysis(FHA)[2] to get the detailed failure inventory with their description and classification, and Fault-Impact Analysis(FIA) to check error propagations from error source to impacted component but since the model did not have any error

propagations or error state transitions, we could not proceed further with them

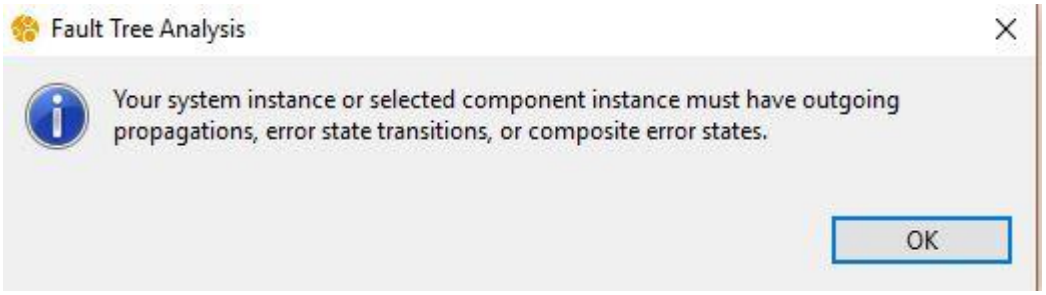

Figure 5 Fault-Tree Analysis

- Resource Budget Analysis Report:

Resource Budget Analysis Report			
Resource Summary:			
MIPS capacity 0.000 MIPS : MIPS budget 0.000 MIPS			
* 0 out of 1 with MIPS capacity			
* 0 out of 1 with MIPS budget			
Detailed Processor MIPS Capacity Report			
Component	Capacity		
processor this_processor	0.000 MIPS		
Total	0.000 MIPS		
Detailed MIPS Budget Report			
Component	Budget	Actual	Notes
device this_kinect_detector	0.000 MIPS	0.000 MIPS	
device this_audio_system	0.000 MIPS	0.000 MIPS	
device this_LED_display	0.000 MIPS	0.000 MIPS	
device this_dome_lights	0.000 MIPS	0.000 MIPS	
process this_controller	0.000 MIPS	0.000 MIPS	
Total		0.000 MIPS	

Figure 6 Resource Budgets Analysis

5 Summary

In this case study investigation, we have provided an overview of the proposed sleep detection system for vehicle drivers and explained how AADL is being used to model the proposed system and represent its architecture. We further tried to investigate and analyse the model using AADL plugins in OSATE but due to the insufficient model implementation we were unable to perform them.

Appendix

AADL textual representation

```
package syst
public

system AntiSleepSystem
end AntiSleepSystem;

system implementation AntiSleepSystem.with_devices
    subcomponents
        this_kinect_detector: device KinectDetector;
        this_audio_system: device AudioSystem;
        this_LED_display: device LEDDisplay;
        this_dome_lights: device DomeLights;
        this_controller: process Controller;
        this_bus: bus HardwareConnection.impl;
        this_processor: processor SystemProcessor.impl;
    connections
        kinect_connection: port this_kinect_detector.usb_output -> this_controller.usb_input;
        LED_display_out: port this_controller.signal_output -> this_LED_display.video_input;
        audio_connection: port this_controller.audio_output -> this_audio_system.audio_input;
        dome_light_connection: port this_controller.dome_lights_output -> this_dome_lights.light_input;

        kinect_bus_connections: bus access this_bus -> this_kinect_detector.bus_access;
        audio_bus_connections: bus access this_bus -> this_audio_system.bus_access;
        LED_bus_connection: bus access this_bus -> this_LED_display.bus_access;
        dome_light_bus_connection: bus access this_bus -> this_dome_lights.bus_access;
        processor_bus_connection: bus access this_bus -> this_processor.bus_access;
end AntiSleepSystem.with_devices;

processor SystemProcessor
    features
        bus_access: requires bus access HardwareConnection;
end SystemProcessor;
processor implementation SystemProcessor.impl
        subcomponents
            this_memory: memory Ram;
end SystemProcessor.impl;

memory Ram
end Ram;
```

Figure 7 AADL textual model of system

```
bus HardwareConnection
end HardwareConnection;

bus implementation HardwareConnection.impl
end HardwareConnection.impl;

process Controller
    features
        signal_output: out event port;
        usb_input: in event port;
        audio_output: out event port;
        dome_lights_output: out event port;
    annex behavior_specification {**
        states
            InActive : initial state;
            Active : state;
        transitions
            t1 : InActive -[active_cmd? = false]-> InActive;
            t2 : InActive -[active_cmd? = true]-> Active;
            t3 : Active -[active_cmd? = true]-> Active;
            t4 : Active -[active_cmd? = false]-> InActive;
**};
end Controller;
process implementation Controller.impl
end Controller.impl;

device KinectDetector
    features
        usb_output: out event port;
        bus_access: requires bus access HardwareConnection;
    annex behavior_specification {**
        states
            InActive : initial state;
            Active : state;
        transitions
            t1 : InActive -[active_cmd? = false]-> InActive;
            t2 : InActive -[active_cmd? = true]-> Active;
            t3 : Active -[active_cmd? = true]-> Active;
            t4 : Active -[active_cmd? = false]-> InActive;
**};
end KinectDetector;
```

Figure 8 AADL textual model of system

```
device implementation KinectDetector.impl
end KinectDetector.impl;

device AudioSystem
    features
        audio_input: in event port;
        bus_access: requires bus access HardwareConnection;
end AudioSystem;
device implementation AudioSystem.impl
end AudioSystem.impl;

device LEDDisplay
    features
        video_input: in event port;
        bus_access: requires bus access HardwareConnection;
end LEDDisplay;
device implementation LEDDisplay.impl
end LEDDisplay.impl;

device DomeLights
    features
        light_input: in event port;
        bus_access: requires bus access HardwareConnection;
end DomeLights;
device implementation DomeLights.impl
end DomeLights.impl;

end syst;
```

Figure 9 AADL textual model of system

References

[1] - https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/driver-drowsiness-detection/

[2] - https://resources.sei.cmu.edu/asset_files/Webinar/2014_018_101_424910.pdf

[3] - http://www.aadl.info/aadl/currentsite/

[4] - https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9407

[5] - https://people.cs.clemson.edu/~johnmc/courses/cpsc875/resources.html