Architecture for Safety-Critical Systems

Analysis of Safety Architectural Patterns for Adaptive Cruise Control

Research Proposal – Assignment 1

Nancy Mago (14874939)
kgz7126@autuni.ac.nz

Roger Rodrigues (15894214)
brg1789@autuni.ac.nz

Premila Pinto (15879772)
zgg2569@autuni.ac.nz

Introduction

Ubiquity of computing and technology has been growing tremendously in the past few decades. Humans have become increasingly dependent on technology and software for many critical operations. Software embedded in systems can fail and cause damage to property, injury or even loss of life. Software systems that can impact safety when it malfunctions or fails are known as safety-critical systems (Kalinsky, 2005). Initially systems that were considered safety-critical were systems used for nuclear reactors, airport control, medical equipment and others. However today the importance of safety in computer systems has expanded and now become part of daily life such as automobile electronics, powered prosthetics, online banking systems and others (Dunn, 2003).

Software architecture plays an important role in safety-critical systems by prescribing solutions to commonly faced problem in software applications. Safety-critical systems at present may be complex and distributed. However due to their nature of use, it is imperative that they are free from faults or must be able to tolerate faults and continue to function correctly (Shukla, 2009). Shortcomings in design of these systems may not be easily caught in the implementation and testing phases of development but may become evident only when these systems are used. Small errors may snowball into big ones if not detected and resolved (Kalinsky, 2005). The fixing of these bugs can be more expense if caught during or after implementation because they involve a lot of rework, higher costs and delays. Thus architectural patterns helps reduce the possibilities of these mistakes and by exploring and evaluating various design patterns for safety-critical computer systems, it is possible to reduce if not eliminate the risk in safety-critical systems (Dunn, 2003).

Cruise Control system in automobiles is one such safety-critical system whose failure or malfunction can have fatal results. There have been multiple instances of cruise control failures reported globally, leaving drivers in high risk situations. Adaptive Cruise Control (ACC) like cruise control system is a driver-assistance vehicle feature that has been recently introduced in some of the high-end automobiles. It steps ahead of the cruise control technology by enabling the vehicle to adjust its speed according to the speed of the vehicle in front of it while maintaining a safe distance using radar sensors (Eyisi et al., 2013). However any malfunction or failure in such a system can be extremely dangerous to the driver and other vehicles on the road. This research aims to investigate all the risks associated with ACC and then analyze and evaluate software architectural design models that implement safety in safety-critical systems and apply the safest to the ACC system design with the intent to reduce the possible risks associated with it. Thus improving the safety of ACC systems and avoid loss of life and property.

Literature Review

Dunn (2003) suggests to mitigate the risk in safety-critical systems, proper design patterns should be implemented. Safety-critical systems can suffer from failures such as sensor failure, effector failure, hardware failure, software failure, operator failure and others. Incorporating reliability and quality design for failures can reduce the risk for the failure of safety-critical systems (Dunn, 2003). Dunn (2003) also suggested adding warning devices and external safety devices to reduce hazard and mishap in safety-critical systems. Considering the electrical water heater, Dunn (2003) explained that an appropriate design implementation sets off the heat if the water temperature increases above the setting. Thus highlighting the importance of design patterns, in safety-critical systems which is the central subject of this research.

Kalinsky (2005) explained that minor faults in design can lead to errors, errors can cause larger faults in the system and this could have a cascading effect on the working of the system, leading to critical malfunctions and failures. While designing a safety-critical system, in addition to covering the physical and functional requirements, it is imperative to conduct a hazard and risk analysis (Kumar, Ramaiah, & Khanaa, 2011). A hazard and risk analysis helps the designer to understand the safety requirements of the system by identifying the risks and dangers that could be brought about by the system. Kalinsky (2005) suggested and explained some of the techniques used to conduct a hazard analysis namely Fault Tree Analysis (FTA) and Event Tree Analysis (ETA). Fault Tree Analysis evaluates different failures that a system encounters and helps to identify the fault tolerance in the design patterns (Wysocki & Debouk, 2007). Wysocki and Debouk (2007) analyzed design patterns for safety-critical systems on the basis of the FTA and assigned weightage to them for failure and success of the design and thus determined how safe the designs were. The next step in the process is risk analysis which is the combination of the probability of the occurrence of an undesirable situation and the severity of its consequences (Kalinsky, 2005).

Kalinsky (2005) discussed a few design patterns for implementing safety in critical systems however the research is limited only to a few design patterns. Kumar et al. (2011) also presented a variety of architectural patterns that implement safety in safety-critical systems. They also analyzed each pattern and discussed the benefits and drawbacks related to the implementation of these patterns. Kumar et al. (2011) then applied the Protected Single Channel model for actuation monitoring on a four fingered robotic hand and evaluated it. However the design patterns presented in these papers may not be exhaustive and this research will investigate the existence of other design patterns that can be used to implement safety in safety-critical systems.

Eyisi et al. (2013) described the working of the Adaptive Cruise Control system and classified the system controls into upper level and lower level controllers. This paper stated that the upper level controller has two modes the velocity mode and the spacing control mode. When the radar does not detect a vehicle in front of it, the system uses the velocity mode. However when a vehicle is detected, it uses the space control

mode. The lower level controls use switching logic to determine if it should apply braking control or throttle control based on the inputs received from the upper level controller. This paper provided this research with valuable insights on the working of ACC systems.

This research will draw together all the proposed architectural patterns to implement safety in safety-critical systems and apply these patterns to the component design of Adaptive Cruise Control and evaluate its effect on the safety of Adaptive Cruise Control.

Research Question

This research will investigate the architectural patterns that can be used to implement the safety requirement in critical systems. Each of these architectural models will then be analyzed and applied to ACC software design. The application of the models on the ACC design components may provide insight into solving some of the safety issues associated with ACC.

This research aims to investigate the most appropriate software architectural model that can be used to implement maximum safety in adaptive cruise control systems.

Research Methodology and Design

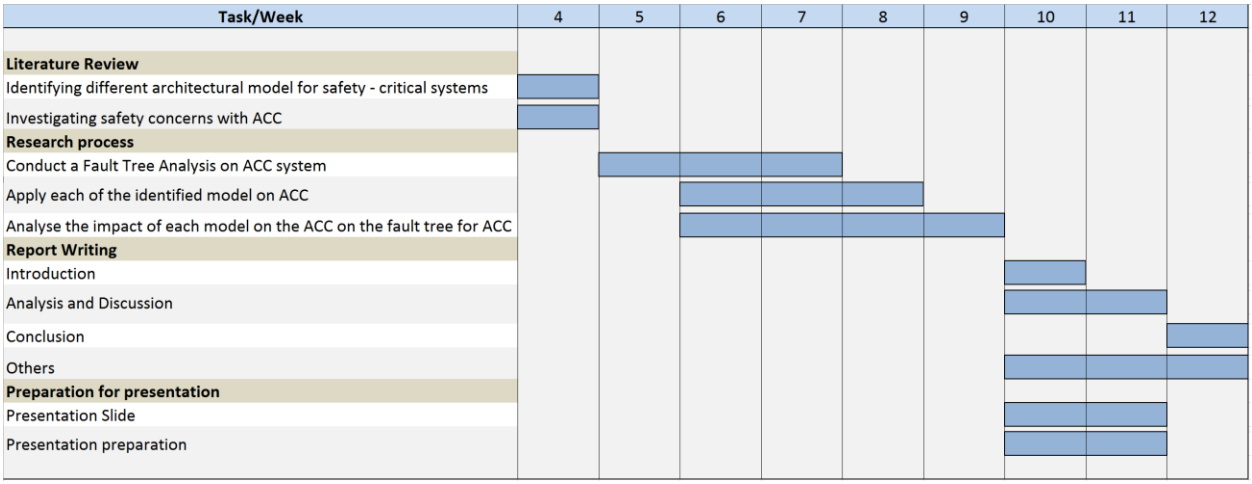
A literature review will help identify all the possible models that can be used to implement safety in safety-critical system. A detailed understanding the working of these models, will provide an understanding on how they can be applied to the Adaptive Cruise Control design components. A hazard analysis of ACC will be conducted as part of this research using Fault Tree Analysis. Application of individual safety models will improve safety and may eliminate some of the faults identified in the fault tree.

The research will be conducted as follows:

- Identify architectural models that implement safety in critical systems
- Conduct a Fault Tree Analysis on ACC system.
- Apply each architectural model that has been identified to implement safety on ACC software components.
- Conduct a critical analysis of the applied safety model and its effect on the fault tree for ACC.

Timeline

Each researcher will participate in the literature review and collaboratively conduct the critical analysis. The following chart will provide the timelines that will be followed in this research and the division of efforts related to this research objective.



References

Dunn, W. R. (2003). Designing safety-critical computer systems. *Computer, 36*(11), 40-46. doi: 10.1109/MC.2003.1244533

Eyisi, E., Zhang, Z., Koutsoukos, X., Porter, J., Karsai, G., & Sztipanovits, J. (2013). Model-Based Control Design and Integration of Cyberphysical Systems: An Adaptive Cruise Control Case Study. *Journal of Control Science and Engineering, 2013*, 1-15. doi: 10.1155/2013/678016

Kalinsky, D. (2005). Architecture of Safety-Critical Systems -- It's one thing to know your system is safety-critical; it's another to know how to deal with it. The author explains how to evaluate errors, categorize them, and safely handle them when they happen. *Embedded Systems Programming*, 14.

Kumar, S., Ramaiah, P., & Khanaa, V. (2011, 2011). *Architectural patterns to design software safety based safety-critical systems*.

Shukla, S. K. (2009). Model-Driven Engineering and Safety-Critical Embedded Software. *Computer, 42*(9), 93-95. doi: 10.1109/MC.2009.294

Wysocki, J., & Debouk, R. (2007). METHODOLOGY FOR ASSESSING SAFETY-CRITICAL SYSTEMS. *International Journal of Modelling & Simulation, 27*(2), 99. doi: 10.2316/Journal.205.2007.2.205-4232