Model-Based Guidelines for Automotive Electronic Systems Software Development

By

Juan Manuel Quezada Gomez

Submitted to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2015

Signature of Author Signature redacted

Juan Manuel Quezada Gomez
System Design and Management Program
February 2015

Certified by Signature redacted

Dov Dori
Visiting Professor at Engineering Systems Division
Thesis Supervisor

Accepted by Signature redacted

Patrick Hale
Director System Design and Management Fellows Program
Senior Lecturer at Engineering Systems Division

This page intentionally left blank

Model-Based Guidelines for Automotive Electronic Systems Software Development

By

Juan Manuel Quezada Gomez

Submitted on February 2015 to the System Design and Management Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

Abstract

The automobile innovation transformed the human life style ever since its introduction to the public, and for over the last one hundred years incumbent technologies have been adopted to improve its performance characteristics. Yet, we need a holistic approach to understand that automobiles shifted from being a mere assembly of mechanical parts to a multidisciplinary system that form the modern automobile.

Thanks to the increased use of electronics and software in automobiles, consumers benefit from better gas mileage, more amenities and features, such as comfort, driving assistance, and entertainment. At the same time, stability and performance of automobiles as systems have been facing deterioration, and eventually vehicle owners are finding that features and functions become inoperative over time, causing frustration, loss of time and money. Reports of problems experienced by vehicle owners have stem from casual factors of system defects that model-based systems engineering can reduce or eliminate.

This research presents a model-based systems engineering approach to an automobile electronic system design. The work is founded on a comprehensive OPM model and engineering guidelines for electronic control module software design. The purpose of the framework developed in this study is to support development of complex vehicle software that allows flexibility for changing features and creating new ones, and enables software developers to pinpoint systemic faults quicker and at earlier lifecycle phases, reducing rework, increasing safety, and providing for more effective resolution of such problems.

Thesis Supervisor: Dov Dori
Title: Visiting Professor at Engineering Systems Division,
Massachusetts Institute of Technology
Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology

Acknowledgements

To my wife Erika and my sons Jaimal and Ever, for all their comprehension, support and help getting through the master's program. For encouraging me to follow my dream and helping me see it through to completion.

To Professor Dov Dori, for providing his valuable wisdom and insights about Model Based Systems Engineering, for his support and guidance of my vision and his dedication to help make this vision a reality.

I owe a debt of gratitude to my former management and colleagues at Ford Motor Company for supporting me in my efforts to get an advanced degree. The System Design and Management program has been extremely educational and a remarkable experience. Special thanks to Armando Chacon, Pat Seashore, Ronald Brombach, Stuart Taylor, McArthur James, Dan King, Antonio Almazan, Edgar Nunez, Raciel Cruz, Guillermo Saavedra for all his advice, support and guidance.

Special thanks to my former colleagues at General Motors, Joe Lutz, Kenneth Balcom, Robert Kirchhoff, Rick Bosley, Douglas Duddles, Greg Stamm, Gabriel Rojas, Juvenal Zavala for all the great advice and support.

To Pat Hale, Bill Foley, Amal Elalam for the career advice and the constant feedback through the SDM program.

To my SDM fellows Eyemi Adepetu, Ari Liberman, Rodolfo Reyes, Shila Ray, Ken Harris, Somwang Thipphayathetthana, Daniel Camacho, Nachiket Joshi, Ankur Kumar, Alex Pina, Alex Sanchez and the rest of my cohort in the MIT-SDM 2013 who inspired me, challenging me and expanding my horizons with their shared experiences, knowledge and motivation to keep me going during the program.

My sincere gratitude goes to my parents for their unwavering support and continual encouragement.

Glossary of Acronyms

ABS/VSC	Anti-lock Braking System / Vehicle Stability Control
ALDL	Assembly Line Diagnostic Link
APQP	Advanced Product Quality Planning
BCM	Body Control Module
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAN	Controller Area Network
CHMSL	Center High Mounted Stop Lamp
DC	Direct Current
DFMEA	Design Failure Mode Effect Analysis
DRL	Daytime Running Lights
DTC	Diagnostic Trouble Code
ECM	Engine Control Unit
ECU	Electronic Control Unit
EMC	Electromagnetic Compatibility
EPS	Electric Power Steering
ESD	Electro Static Discharge
FMEA	Failure Mode and Effects Analysis
FOB	Frequency Operating Button
GM	General Motors
HVAC	Heating Ventilating and Air Conditioning
INCOSE	International Council of Systems Engineering
ISO	International Standards Organization
LF	Left Front
LR	Left Rear
MBE	Model Based Engineering
MBSE	Model Based Systems Engineering
MDE	Model Driven Engineering
MISRA	Motor Industry Software Reliability Association
MIT	Massachusetts Institute of technology
NHTSA	National Highway Traffic and Safety Association
OBD	On-Board Diagnostics
OBDI	On-Board Diagnostics version 1
OBDII	On-Board Diagnostics version 2
OEM	Original Equipment Manufacturer
OPM	Object-Process Methodology
OSI	Open Systems Interconnection
PCM	Powertrain Control Module
PD	Product Development
PWM	Pulse Width Modulation
R&D	Research and Development
RF	Right Front
RR	Right Rear
SAE	Society of Automotive Engineers
SDP	Software development process
TCM	Transmission Control Module
UART	Universal Asynchronous Receiver Transmitter
U.S.	United States
UML/SYSML	Unified Modeling Language / Systems Modeling Language
USB	Universal Serial Bus
VDP	Vehicle Development Process
VLSI	Very Large Scale Integration

List of Figures

List of Tables

Table of Contents

CHAPTER 1 INTRODUCTION

1.1 Preface

The human means of transportation have evolved since thousands of years ago, with the invention of the automobile or the "horseless carriage", in the early 1900's the mechanical systems became an incumbent technology that dominated the entire landscape, and for more than half of a century the car makers focused their efforts to design automobiles around the internal combustion engine with the support of basic electrical components such the starter, alternator and battery.

However, the gradual introduction of newer technologies such as: a) Electrical devices, switches and other passive components; b) Electro-mechanical, relays, motors, solenoids, etc.; c) Electronics, in the form of electronic control units or ECU's; d) Software, application software, calibrations and embedded software; Became key elements of modern automobile systems.

Today an automobile is comprised by components designed with the use of several engineering disciplines; furthermore the offspring of incumbent technologies mixed with mechanical systems. Newer technologies cause changes on the environment and posit challenges to company dynamics, project management, engineering, manufacturing and serviceability of this type of products.

1.2 Motivation

Electronic systems in the automotive industry face several problems related to software, ever since the introduction of solid state electronics in this industry. The automobile engineering shifted from pure mechanical systems to a collection of sub-systems where several disciplines work together to perform functions related to the transportation characteristics of those products (i.e. Safety, Comfort, Security, Infotainment, etc.).

With the increased use of electronics and software in the automobile, consumers benefited of better gas mileage, more comfort, driving assistance, etc., but at the same time faced a deterioration of the stability and performance and eventually vehicle owners found that features and functions become inoperative over time, causing frustration, loss of time and money.

During the time I worked in the industry, I have been given the privilege to participate in at least five major vehicle designs in three countries (Mexico, United States of America and South Korea), where I witnessed several defects of the vehicle performance during and/or after the design, validation and manufacturing phases.

At every opportunity that a trouble presented itself, I found that the defects were the result of engineering mistakes made during the design, and in general I assumed that it was normal business. However, after I took the Model Based Systems Engineering class, I understood that each of those mistakes were the result of the lack of understanding of the essence of systems engineering.

1.3 Objectives

The objective of this research is to investigate and develop a model-based framework for body electronic modules software design for automotive product development. This R&D effort shall explore and exploit user's experience to devise a systematic method that could help to address the existing problems associated to body electronic modules software development.

Recommendations for change in the automotive industry are developed through subject matter research and cross industry interviews. Areas for future study and development are outlined in the concluding thoughts.

1.4 Approach and Methodology

This thesis takes the approach of model-based systems engineering as a critical element in electronic modules design for the automotive industry. It assumes that many automotive firms face similar struggles in designing software, and that product development frameworks are shaped in part by the technologies produced by external vendors and in some other cases developed internally. For the most part, differences in the competitive positions within the industry can be related to the strategy followed to design and develop the software models, and from there be able to react to design constraints throughout the development process and produce robust software.

The beginning chapters of this thesis define the elements of the automobile, seen as a system, on which its design context can be characterized as a multidisciplinary engineering design process. The middle chapters address the unique nature of model-based framework for body electronic modules software design and several of its departure points from existing product development frameworks. The final chapters discuss recommendations for overcoming the inherent conflicts between the new technologies and the current way of doing business. Concluding thoughts regarding areas for future study and development are provided in the final chapter.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

The following sections outline several key elements of systems engineering related to automotive engineering that facilitate the discussion in later chapters. The intent is to provide a brief overview of important concepts, models and terminology.

The primary objective of this section is to introduce the evolution of the automobile technology, define the key system aspects of electronic controls and embedded software and the model-based systems engineering implications to create competitive performance value during the product lifecycle, to review some of the main ideas of systems thinking and systems engineering.

2.2 Evolution of automobile systems

People had dreamed of a self-propelled vehicle for centuries, this motivated innovators to create inventions such as the wheel and sled, that helped make animal transport more efficient through the introduction of vehicles prior to the Industrial Revolution (Berger, 2001).

Figure 1 Typical method of transportation before the 19th century (Davison, 1840)

During that era innovators surged all over the world ager to find new methods of transportation, by 1890's motorized vehicles were being produced for purchase, however only the wealthy could afford these hand-crafted curiosities, which were seemingly more of a novelty than a practical invention.

During the early 20th century the automobile began sweeping in the U.S. and had one of the most significant impacts on the American lifestyle, since the beginning mechanical parts dominated the whole industry. However, the automobile evolved from decade to decade as new incumbent technologies were introduced in order to keep up with consumer alluring. There are three relevant technologies that changed the paradigms in this industry: a) the starter, b) introduction of electronics, and c) introduction of software.

2.2.1 The birth of automobile and the mechanical components

In the mid 1800's the steam engines dominated, but innovators in Europe and the U.S. developed two inventions that shifted this trend:

According to (Huges, 1996), Ányos Jedlik from Hungary invented a type of electric motor in 1828. Around the same time frame Thomas Davenport was credited for inventing the first American DC electrical motor in 1834. Davenport installed his motor in a small model car which he operated on a short circular electrified track opening the door for electric propulsion for the automobile use.

In 1835, the first records of batteries used in automobiles relate to Sibrandus Stratingh who attended the Academy of Groningen in the Netherlands, and along with his assistant Christopher Becker created a small-scale electrical car powered by non-rechargeable primary cells (University of Groningen Netherlands, 2013). Thirty years later an improved battery was presented by Gaston Plante in France in 1865, (Dell, 2001) as well as his fellow countryman Camille Faure in 1881, paved the way for electric cars to flourish in Europe.

In 1891, these inventions led William Morrison of Des Moines Iowa, to develop a six-passenger electric wagon that is often considered the first practical electric vehicle in the United States (The United States Library of Congress, 2014).

Meanwhile, from 1860 to 1910, some innovators followed the path towards electric vehicles, while other inventors focused their research on gasoline-powered methods. Various forms of internal combustion engines were developed. In December 1878 a German car engineer and engine designer Karl Benz, was granted a patent for creating a reliable petrol two-stroke engine after 8 years of development (Daimler AG, 2014).

Twenty years later, in the US in 1891, Henry Ford became an engineer with the Edison Illuminating Company, and two years later he was promoted as a Chief Engineer. He had enough time and money to devote attention to his personal experiments on gasoline engines (Wikipedia, wiki/Henry_Ford, 2014) These experiments culminated in 1896 with the completion of a self-propelled vehicle which he named the Ford Quadricycle, illustrated in Figure 2 (Wikipedia, File:Henry_Ford_-_Quadricycle,_1905.jpg#filehistory, 2005).

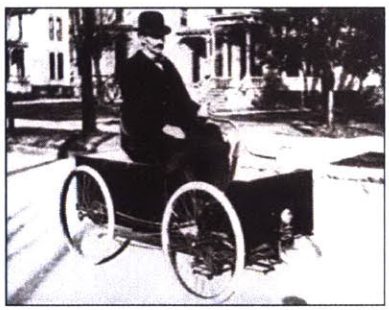


Figure 2 Ford's quadricycle.

After successfully established at the beginning of the 20th century as the leading technology behind the automobile propulsion, the electric car began to lose its position due to three major developments related to the internal combustion engine: a) Starter motor, b) Low cost thanks to improved manufacturing lines, c) the discovery of oil wells in the U.S.

By the 1920s, following these developments the improved road infrastructure was being created between American cities; in order to make use of these roads, vehicles with greater range than that offered by electric cars were needed. The discovery of large reserves of petroleum in Texas, Oklahoma, and California led to the wide availability of affordable gasoline, making gas-powered cars cheaper to operate over long distances. Electric cars were limited to urban use by their slow speed (no more than 24–32 km/h or 15–20 mph) and low range (30–40 miles or 50–65 km), and gasoline cars were now able to travel farther and faster than equivalent electrics (Wikipedia, wiki/History_of_the_electric_vehicle, 2014).

Several improvements to the automobile during the first decade of the 20th century made the vehicles more user friendly; however, the majority of the features offered complemented the internal combustion engine: convertible top, hand operated windshield, oil based headlamps, and mechanical speedometer.

2.2.2 The introduction of electromechanical devices

In 1912 Charles Kettering invented the first practical electric automobile starter (MacMahon, 2009). Kettering's invention makes gasoline-powered autos more alluring to consumers by eliminating the unwieldy hand crank starter and ultimately helps pave the way for the electric car's demise (PBS, 2010). From that moment on, the gasoline engines dominated as the preferred propulsion system accompanied by mechanical parts. One of the key parts for the engine control was the mechanical carburetor.

By the late 1960's some OEMs developed the concept of the on-board diagnostics around electronic fuel injection control. This milestone shifted the paradigms of mechanical engine control. The electronic fuel injection became an incumbent technology and remained that way until it was fully adapted in the 1980's for massive production thanks to the introduction of the microprocessor. When this industry adopted a more elaborated fuel injection control system called Engine Control Module (ECM), this technological innovation marked the beginning of a broader scale adoption of electronics in this industry (Wikipedia, /wiki/On-board_diagnostics, 2014).

2.2.3 The introduction of electronics

With the introduction of the microprocessor in the early 80's, the industry adopted a more elaborate fuel injection system, and in 1980 General Motors introduced the Assembly Line Diagnostic Link (ALDL), which was not intended to sell outside of the factory. In the same year the serial communication standards started to appear; first the UART, then in 1986 the half duplex UART (Wikipedia, /wiki/On-board_diagnostics, 2014).
By 1988, the Society of Automotive Engineers (SAE) recommends a standardized diagnostic connector and set of diagnostic test signals. In 1991 the California Air Resources Board (CARB) motivated the first government regulation, the OBD-I, to standardize the diagnostic connector (or ALDL), and by 1992 all the vehicles to be sold in California required to feature the OBDI capabilities. In 1996 the standard evolved to OBD-II and specification is made mandatory for all cars sold in the United States. This

standard made a huge impact on automobile OEMs, and the complexity of the automobile grew and spread to other disciplines (Hellestrand, 2014).

2.2.4 The introduction of software

Today's automobiles contain many complex electronic systems. Each system may incorporate a large number of electronic control units (ECUs), which communicate through layers of networks. These ECUs are becoming more numerous, complex, and interconnected. In every new vehicle generation, they handle an additional set of functions. Currently, leading-edge luxury vehicles may use as many as 100 ECUs.

As of 2005 and beyond, there are over 70 ECUs in most new designs that are in production. The ECUs may be connected by up to five buses. Accordingly, the automotive-electronics market has been growing faster than the overall electronics market and actual vehicle production. The following plot from Chip Design Magazine (Hellestrand, 2014), illustrates the rate of electronics growth in the auto industry since 1950's.

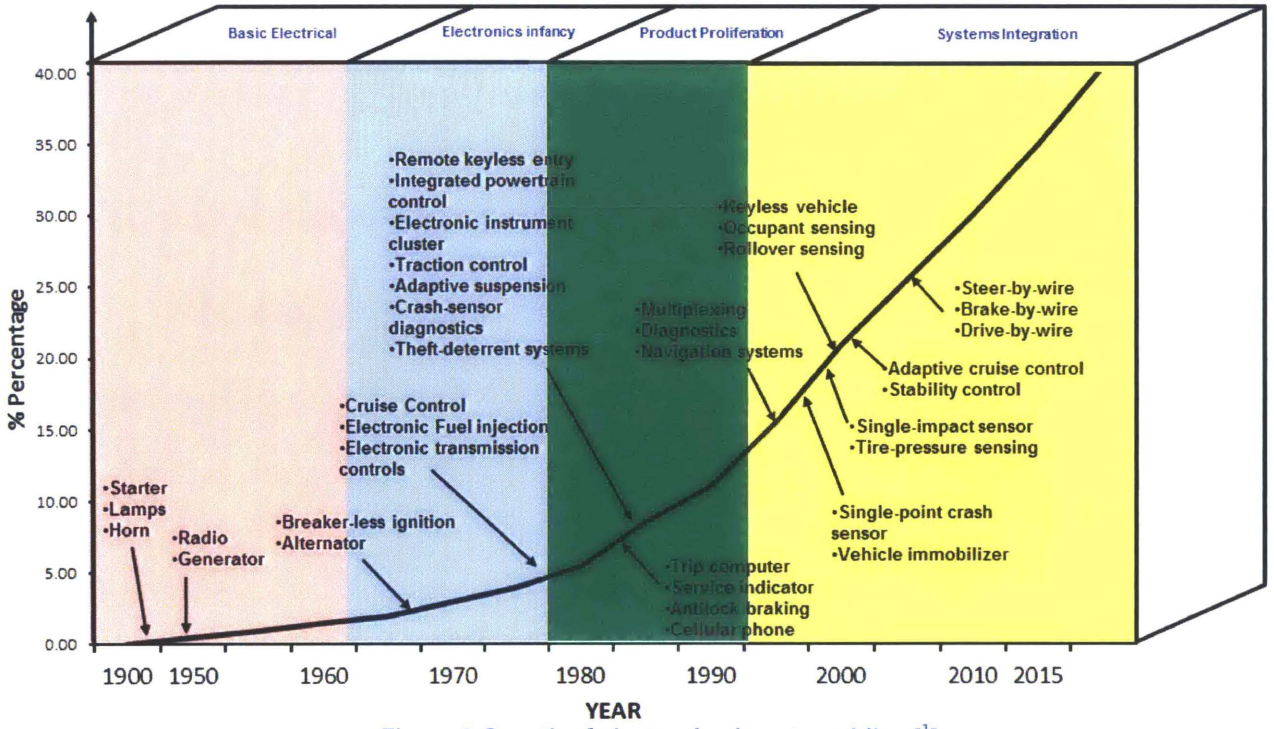


Figure 3 Growth of electronics in automobiles [1]

2.2.5 Electronic Control Unit design

The evolution of technology, after the introduction of electronics in the automotive industry, facilitated the design of complex systems. However, due to the increase of complexity, the automobiles needed control units to manage specific vehicle functions, such as powertrain, brake system, air bags, lights, immobilizer, etc. This gave birth to

[1] Figure modified from *Chip Design Magazine on line website (added years 2010 and 2015), retrieved on March 2014: http://chipdesignmag.com/display.php?articleId=57*

newer vehicle architectures featuring decoupled subsystems that communicate with each other by serial communications. Figure 4 illustrate the complexity of the first vehicles featuring networking protocols in the 1990's and a decade later in 2000's (Rince, 2012).

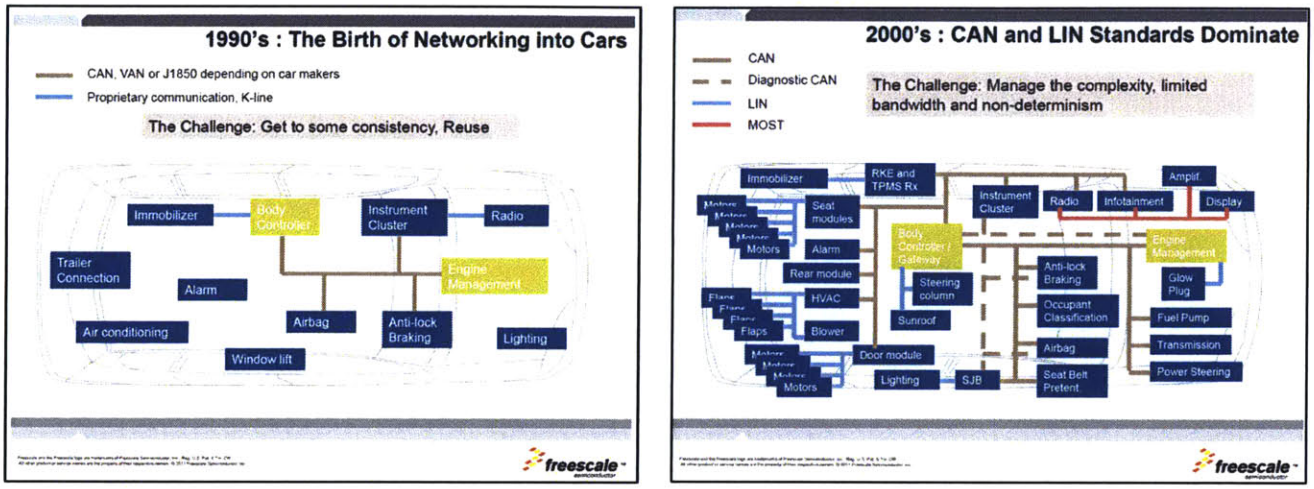


Figure 4 Automobile networking

With the introduction of serial communications, the automobiles Increased functionality, Improved reliability, allowed a "modular" approach to vehicle design, provided access to diagnostic information to improve vehicle repair and maintenance, and enabled reprogramming of modules in the vehicle.

2.2.5.1 Network Topologies

Network topologies were introduced by the automakers in the late 1980's. Later, the SAE introduced J1850, and ISO introduced the ISO/IEC 7498-1 standards. According to Wikipedia "*A network topology is the arrangement of the various elements (links, nodes, etc.) of a computer network. Essentially, it is the topological structure of a network and may be depicted physically or logically*" (Wikipedia, wiki/Network_topology, 2014).

1. Three types of topologies are used in the auto industry, as summarized in Figure 5, which illustrates a basic vehicle network.

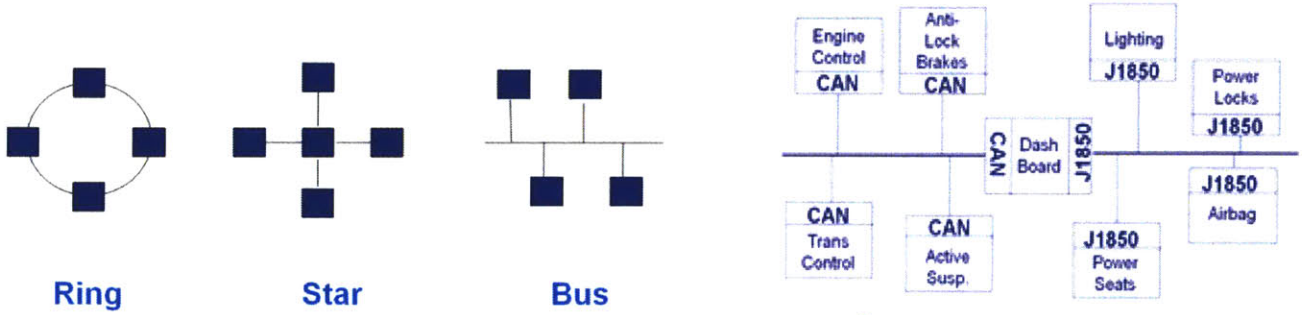


Figure 5 Network Topologies [2]

[2] Constructed from *Dearborn Group Technologies, 2001, Seminar "Introduction to in-vehicle networking- Network Topologies" pp 14,35*

2.2.5.2 Serial Communication protocols

Due to the standardization of the On Board Diagnostics (OBD) in Europe and the United States, two serial communication protocols were designed to allow microcontrollers and devices to communicate with each other within a vehicle network.

The SAE introduced the J1850 protocol in 1993, and by 1996 all the vehicles to be sold in the US were required to meet the standard. There are 2 variants: 10.4 Kbit/s (single wire, VPW) and 41.6 Kbit/s (2 wire, PWM) that were mainly used by US manufacturers (Wikipedia, /wiki/On-board_diagnostics, 2014).

In Europe, the International Organization for Standardization ISO also introduced the CAN standard. Development of the CAN bus started originally in 1983 at Robert Bosch GmbH. It was officially released in 1986 at the SAE congress in Detroit, Michigan. The first CAN controller chips, produced by Intel and Philips, appeared on the market in 1987. By 1993 the International Organization for Standardization released the CAN standard ISO 11898, which was later restructured into two parts: ISO 11898-1, which covers the data link layer, and ISO 11898-2, which covers the CAN physical layer for high-speed CAN" (Wikipedia, wiki/CAN_bus, 2014). The two protocols were widely used in this industry, but eventually the CAN protocol dominated, and today it is used worldwide.

2.2.5.3 Open Systems Interconnect (OSI) 7 Layer

Other relevant aspect of the communication protocols is the OSI layer that was introduced In order to ease the interconnection of different hardware and software communications for vehicle systems, the ISO developed the "Open Systems Interconnection Reference Model" OSI-RM. This model groups the communication tasks into logical chunks. The layers of the OSI model are intended to be standalone, self-contained entities. They cannot perform any function without the other layers, but from a programming point of view there are complete single entities (Dearborn Group Technologies, 2001).

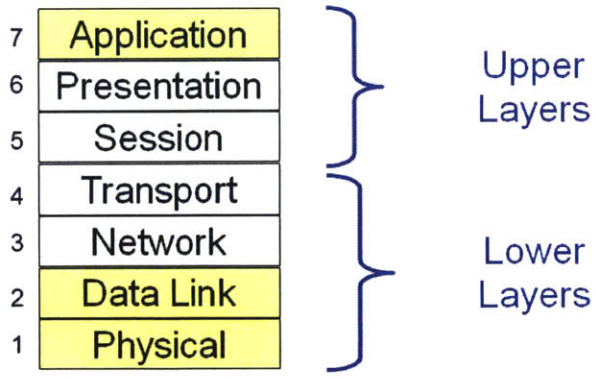


Figure 6 The OSI 7 Layer model [3]

[3] Constructed from *Dearborn Group Technologies, 2001, Seminar "Introduction to in-vehicle networking-Network Topologies" pp 17,18,19*

2.3 Characteristics of systems applicable to automobiles

After the introduction of computers in the 1980's, due to the demand for newer features and capabilities of the control systems, augmented capabilities drove increase of vehicle complexity. Systems engineering became an important element in the design of the automobiles ever since. The transformation in the design processes this industry experienced the last 60 years evolved from mechanical engineering to systems engineering. Several key characteristics can be identified in the automobiles: interaction, hierarchy, emergence, controls and communication, dynamic, interdisciplinarity.

The three key factors that play an important role in the automobile electronics and software design are interaction, dynamism, and interdisciplinary.

2.3.1 Characteristic #1 - Interaction

An automobile can be seen as a collection of subsystems (see Figure 7) that as a whole provide several functions. The elements are connected by physical, systemic, energy or mechanical exchanges. Figure 4 (left) shows a summary of features of a typical 2013 automobile. A vehicle could be seen in simplified domains on the right.

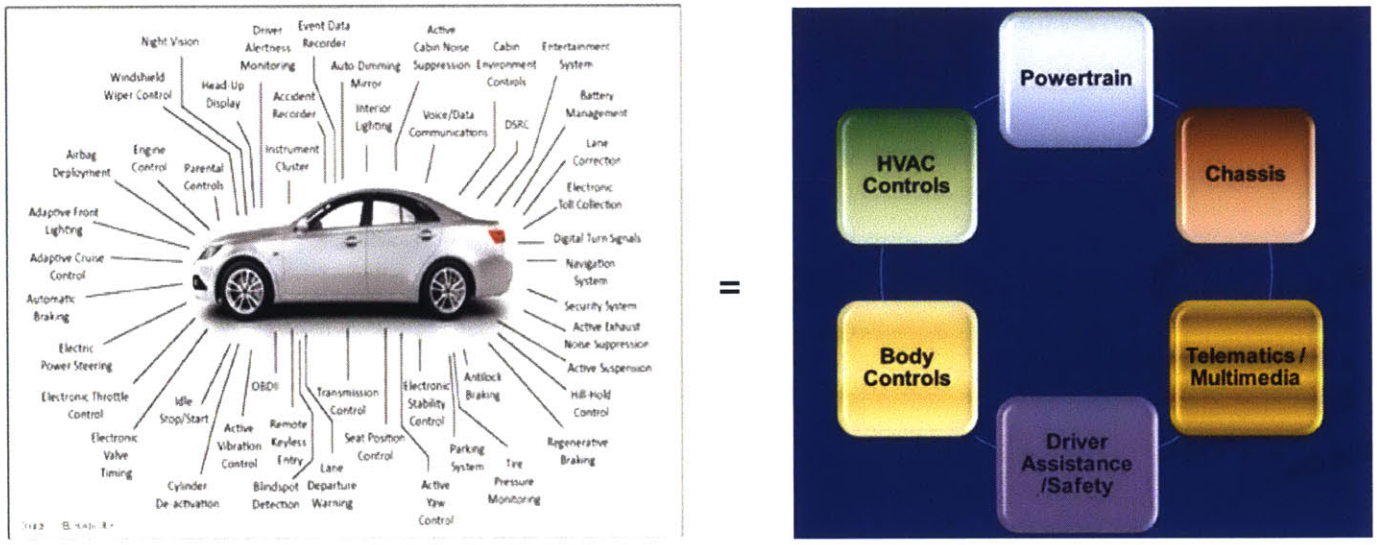


Figure 7 View of an automobile as a collection of subsystems [4]

2.3.2 Characteristic #2 - Dynamism

Every subsystem of an automobile is exposed to dynamic conditions during the entire life cycle, since the moment a vehicle is designed, manufactured, transported, stored, sold to customer, used and disposed. At each of those stages, the elements of the automobile interact under various conditions that affect the performance of each component; see Figure 8. This characteristic shall be considered when designing this type of products. Every member of an enterprise participating in the design of this type of

---

4 Constructed with image retrieved From: *http://www.cvel.clemson.edu/auto/systems/auto-systems.html* / Automotive Electronic Systems|

products has to be aware of the effects of the dynamics involved in this system and the effect to the vehicle performance.
For this type of complex systems, a challenge arises, because the automobiles performance in these dynamic conditions can affect the safety of human beings if any of the elements (mechanical, electrical, electronics, software) are not specified, developed, integrated and validated properly.

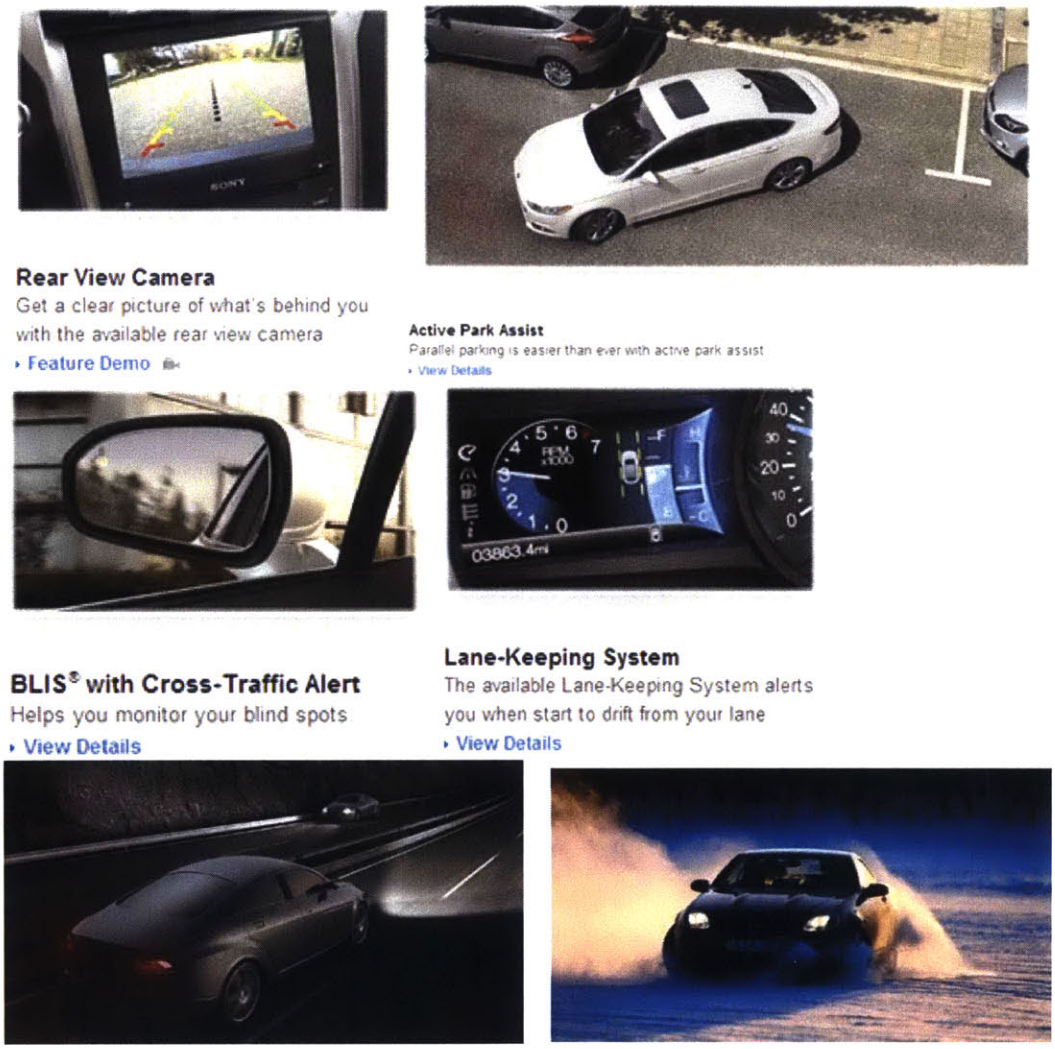


Figure 8 Dynamic environment for a vehicle to perform [5]

2.3.3 Characteristic #3 - Interdisciplinarity

To illustrate the interdisciplinarity of modern automobiles, we note that after the 1960's, the evolution of the electronics industry caused a shift to the auto industry. By 1970 few OEMs developed the concept of the on-board diagnostics with electronic fuel injection

[5] Constructed with images *retrieved from Ford Motor Company website (Top pictures):* *http://www.ford.com/cars/fusion/features/#page=FeatureCategory2*
And from Continental Automotive Group (Bottom pictures):
http://www.conti-engineering.com/www/engineering_services_de_en/themes/brakes_chassis/chassis_engineering_en.html

control (see Figure 3). With the introduction of the microprocessor in the early 80's, the industry adopted a more elaborate fuel injection system – the ECM. In addition to electronics, the introduction of microprocessors also required the use of software. That was the beginning of systems composed of more than one discipline. As we can see in Figure 5, this industry embarked on a journey to integrate sophisticated systems that over the last 3 decades became a challenge. The exponential use of electronics required a great deal of systems integration, and this is true still in 2015.

2.3.4 Characteristic #4 - Emergence

When the components of an automobile are put together, they have certain level of complexity and performance specifications. However, when these components start to interact with the rest of the vehicle, the whole system performance changes due to systemic effects of the components interaction and cause an emergent behavior. One example of this characteristic is the electrical charging system, where power fluctuations due to the consumption of energy (required by the other components in the vehicle) cause oscillations on the system battery level. Such fluctuations may be perfectly fine for all of the components in the vehicle, but a phenomenon appears on the external lamps during night time: such fluctuations cause a decrease on the light intensity delivered by the lamp for a very short period of time, all within the system defined limits. The fluctuation is a small glitch, also known as "flickering", that can be detected by the human eye.

2.4 The vehicle development process in the automotive industry

Similar to many other industries, the vehicle development process adopted by the automotive industry is the waterfall process. In general, the phases followed are: definition, design, development, validation, and launch. Nearly all activities which make up the program timing are an interdisciplinary activity, requiring contributions from stakeholders, marketing, finance, design, manufacturing, purchasing, etc. (Macias Anaya, 1999) The generic product development process essentially consists of six phases, as shown in Figure 9, lasting 18, 24 or 36 moths (Teske, 2007).

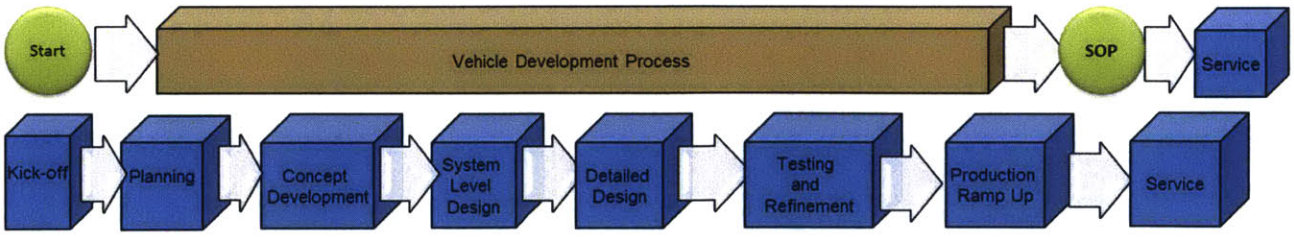


Figure 9 Generic vehicle development process in the automotive industry

There are several important factors around the product development process and the dynamics in the organizations that ultimately affect the overall vehicle design. Using a systems engineering approach, the management of a product development organization can be synthetized as a control loop, where six components—owners, controller, control, actuators and sensors—affect the overall plant outcome. Figure 10 illustrates a high level view of the control loop [6], [7].

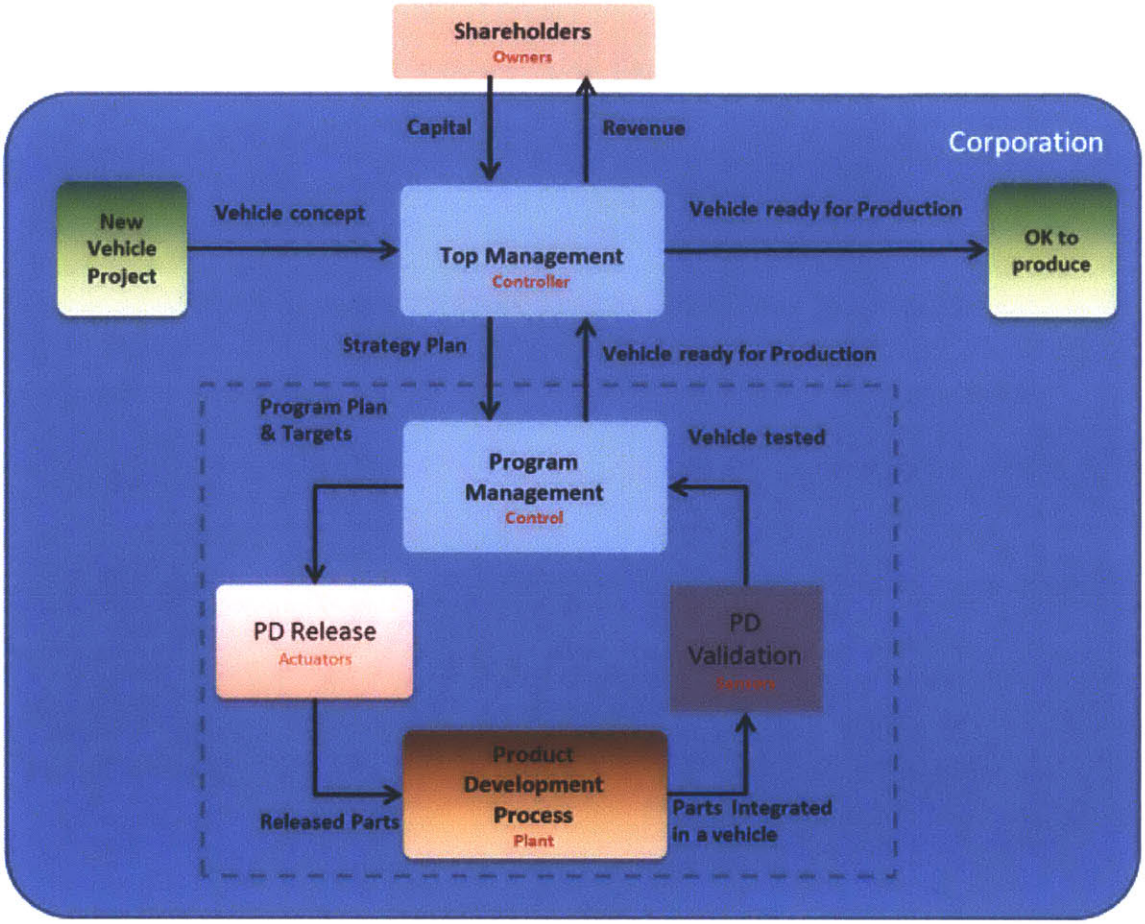


Figure 10 The control loop

In a more detailed close up view of the exchange of information during the design phases, the flow among the design phases illustrated in the Figure 11 provides the big picture of the importance of stakeholder control actions to the "plant" (the vehicle development process itself).

[6] The control loop synthetized here contain references from the Global Integration of Brands and New Product Development at General Motors article (Towsend, Cavusgil, & Baba, 2009)

[7] The control loop synthetized here contain references from the GM Ignition Switch Recall: Investigation Update article from (Energy & Commerce Committee, 2014, p. Document Binder: Document 1) and the Investigation report from Anton Valukas (Valukas, 2014, pp. Sections: IV, VII, VIII, Appendix B)

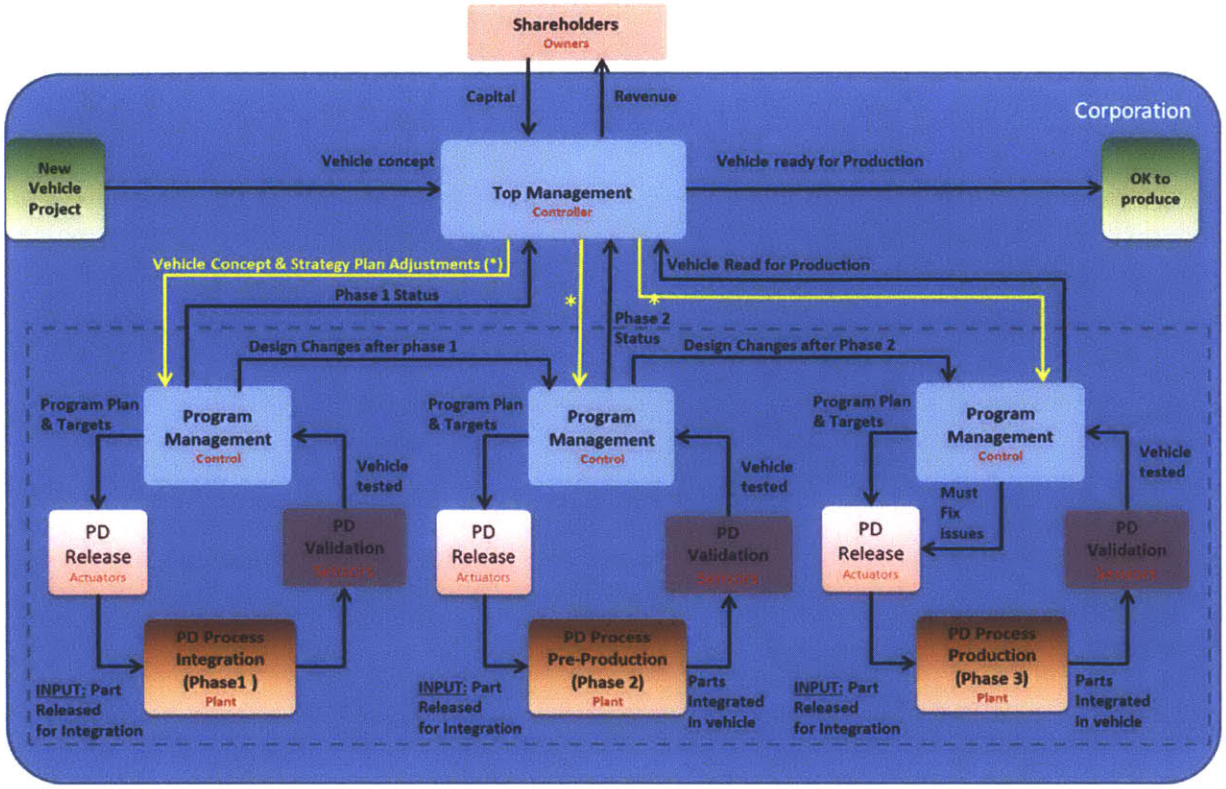


Figure 11 Expanded control loop

The overall development process is controlled by the program managers, who are the stakeholders in product development organizations. Product development release and validation are the engineering groups that design, release and validate the components used in the vehicles (actuators and sensors). Finally, the timing that governs the product development process is determined, maintained and overseen by the program management (Control).

According to INCOSE, *"Systems Engineering is a discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule."* (INCOSE, Systems Engineering Handbook, 2006). Traditionally the product development process also considers the Systems Development Process, also known as the "V" process.

2.4.1 The "V" process in the automotive industry

The use of the "V" is the most common practice in the industry to develop software for the modules used in automobiles, pretty much in any given vehicle automaker, and it was adopted to meet the requirements of the stakeholders, such as program timing deliverables, design requirements, automobile features, industry regulations, and vehicle performance requirements. The relationship of the "V" process with the project cycle in the automotive industry is very similar to the one in Figure 12.

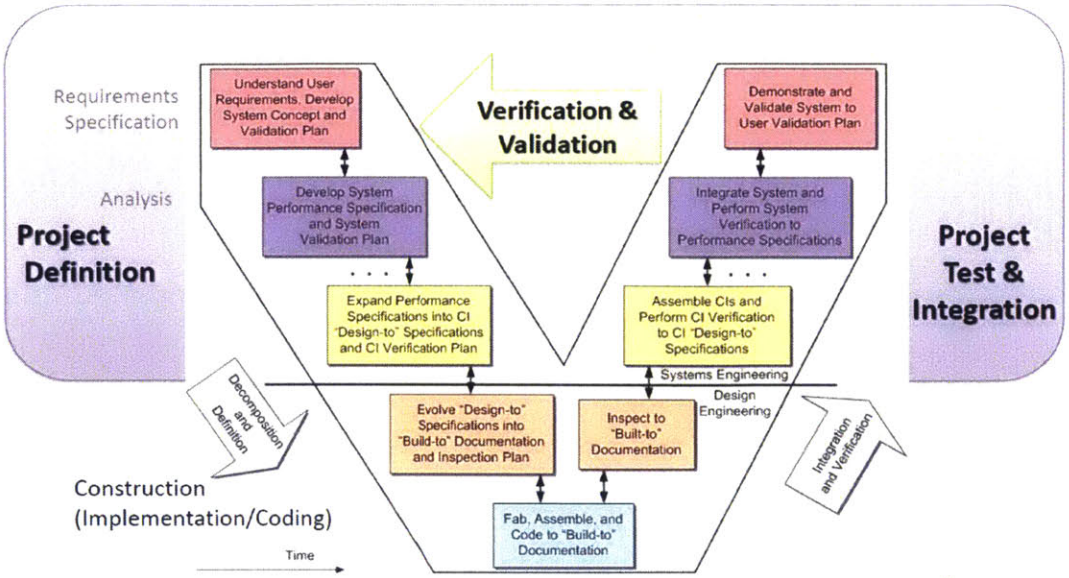


Figure 12 The relationship of Systems Engineering to the Project Cycle [8]

2.4.2 System design methodologies in the automotive industry

Several methods and processes are applied in the auto industry in addition to the V model. In 1986 Motorola developed a set of techniques and tools for process improvement in a philosophy named Six Sigma (Six sigma consulting group, 2014). "Sigma" is a statistical term that measures how far a given process deviates from perfection. The central idea behind Six Sigma is that if you can measure how many "defects" you have in a process, you can systematically figure out how to eliminate them and get as close to "zero defects" as possible. To achieve Six Sigma Quality, a process must produce no more than 3.4 defects per million opportunities. An "opportunity" is defined as a chance for nonconformance, or not meeting the required specifications. This requires nearly flawless execution of key processes (General Electric, 2014).

2.4.2.1 Six Sigma Methodology

The automotive industry targets the reduction of product defects due to design flaws experiencing over six decades of using Six Sigma (ever since it was introduced) in conjunction with two methodologies included in the SAE standard J1739: FMEA and DFMEA, later incorporated into Advanced Product Quality Planning (APQP).

2.4.2.2 Failure Mode Engineering Analysis (FMEA)

According to the SAE –J-1939 Standard: "*A Failure Mode Engineering Analysis (FMEA) can be described as a systemized group of activities intended to recognize and evaluate the potential failure of a product/process and its effects, identify actions which could eliminate or reduce the chance of the potential failure occurring, and document the process. It is complementary to the process of defining what a design or process must do to satisfy the customer*" (SAE, 1993, p. 1).

[8] Image from the *Engineering Management Journal* "The Relationship of Systems Engineering to the Project Cycle," (Forsberg & Mooz, 1992)

2.4.2.3 Failure Mode and Effects Analysis in Design (DFMEA) SAE J1739

This SAE Recommended Practice was jointly developed by DaimlerChrysler Corporation, Ford Motor Company, and General Motors Corporation. This document introduces the topic of potential Failure Mode and Effects Analysis (FMEA) and gives general guidance in the application of the technique. An FMEA's focus is on the design, whether it is of the product, the process or the machinery used to build the product.

According to the SAE J1739 *"The DFMEA is an analytical technique utilized primarily by a design responsible engineer/team as a means to assure that, to the extent possible, potential Failure Modes and their associated Causes/Mechanisms have been considered and addressed during the design of a component. End items, along with every related system, subassembly and component, should be evaluated. In its most rigorous form, an FMEA is a summary of the team's thoughts (including an analysis of items that could go wrong based on experience) as a component, subsystem, or system is designed. This systematic approach parallels, formalizes, and documents the mental disciplines that an engineer normally goes through in any design process.*

The process begins by developing a listing of what the design is expected to do, and what it is expected not to do (i.e., the design intent). Customer wants and needs should be incorporated, which may be determined from sources such as Quality Function Deployment (QFD), Vehicle Requirements Documents, known product requirements, and/or manufacturing/assembly/service/ recycling requirements. The better the definition of the desired characteristics, the easier it is to identify potential Failure Modes for preventive/corrective action." (SAE, 1993, pp. 5,7).

2.4.2.4 Advanced Product Quality Planning (APQP)

Following SAE J1739, Advanced Product Quality Planning (APQP) has been used in the automotive industry broadly; the graphic shown on Figure 13 [[9]], illustrate the six steps proposed for this method.

[9] The Figure 13 was constructed with information from the GM APQP Reference manual (General Motors Corporation, 2005, pp. III, IV), this Advanced Product Quality Planning (APQP) process was designed in conjunction with Ford, Chrysler, General Motors and all the suppliers of automotive parts and components in the U.S. it was designed a quality planning tool. It features 18 tasks to execute during the product development process; it consist of 6 steps that shall control the development lifecycle.

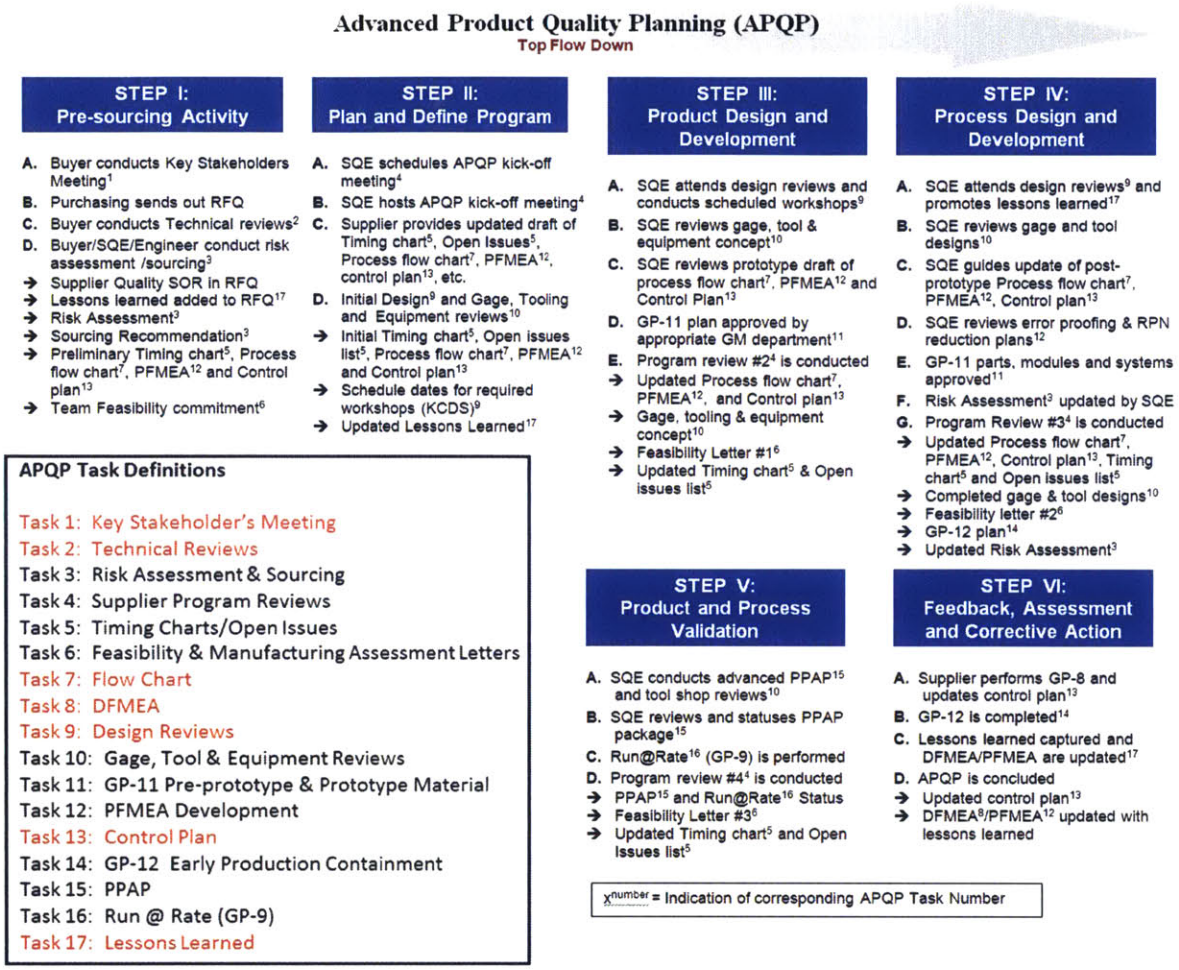


Figure 13 The APQP elements

2.4.3 The stakeholders of automotive product development

A key aspect of the automobile design process discussed in section 2.4 is the people involved on the design process itself. Based on the definition of a stakeholder from (Rozanski & Woods, 2005) *"Stakeholder in the architecture of a system is an individual, team, organization, or classes thereof, having an interest in the realization of the system.*

Due to the importance of software design for electronic components used in the automobiles, the relationship of the stakeholders and the employees that are part of the development process in the auto industry is a critical element of a successful design. Traditional software development has been driven by the need of the delivered software to meet the requirements of users.

Although the definition of the term user varies, all software development methods are based around this principle in one way or another. However, the people affected by a software system are not limited to those who use it. Software systems are not just used: They have to be built and tested, they have to be operated, and they may have to be repaired, There are usually enhanced, and of course they have to be paid for. Each of

these activities involves a number – possibly a significant number – of people in addition to the users."

We can assume that those individuals are the *"Stakeholders"*, who work for companies, either managing, overseeing and/or executing the business plans to achieve the designs the enterprises produce for sale in the markets worldwide.

2.4.3.1 Stakeholder categorization

Rozanski & Woods included a further description about the categories of stakeholders *"The great majority of system development projects include representatives from most if not all of these stakeholder groups, although their relative importance will obviously vary from project to project. However, if you do not at least consider each class, you will have problems in the future. You need to balance and prioritize the needs of the different stakeholder groups, so that when conflicts occur, you can make sound, well-reasoned decisions"* (Rozanski & Woods, 2005).

The question that arises is how do the stakeholders relate to the automotive development process? Considering the overall development process presented on section 2.4 (see figures 9, 10, 11), the stakeholder participation in the product development process is illustrated in Figure 14, where the share of responsibility in the development process can be concluded.

The "Control" performed by management, "Actuator" action by the design engineers, "Sensors" by the validation, manufacturing, service, purchasing engineers, and finally the "Plant" is the Vehicle Development Process that is controlled, actuated and sensed by the above stakeholders entirely.

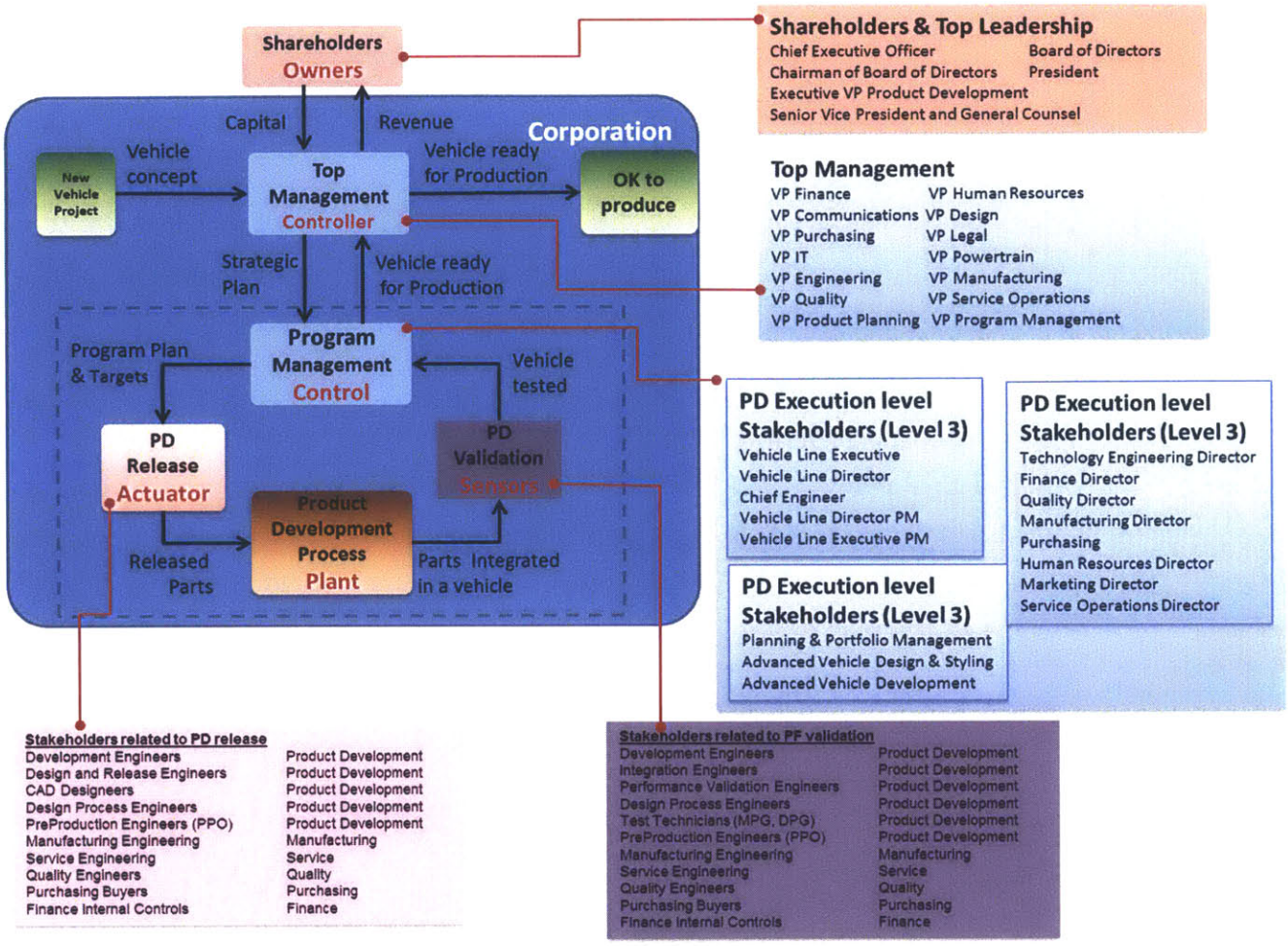


Figure 14 Stakeholder participation in the automotive development process [10]

2.5 Relevance of systems thinking and systems engineering for automotive electronic systems software design

The vast majority of engineering organizations that design modern systems, such as the automobile, face many challenges which vary in nature from organizational, managerial, project execution constraints to technical. During the last 14 years, I have personally observed a particular characteristic: stakeholders at all levels have a limited view that is often linked to the engineering group and organization to which they belong due to the specialization

In a presentation by Heidi Davidz in 2006, provided a view of the mechanisms that enable systems thinking: *"The participants are influenced by their unique system of*

[10] This figure of the stakeholder participation in the automotive development process was created with information from the following sources: (Macias Anaya, 1999) , (Teske, 2007), (Towsend, Cavusgil, & Baba, 2009), (Energy & Commerce Committee, 2014), and the Investigation report from Anton Valukas (Valukas, 2014, pp. Sections: IV, VII, VIII, Appendix B), with reference to the development process followed by General Motors.

interest. Additionally, the articulation of a systems thinking definition is not necessarily a direct measure of the understanding of that concept in a person's mind. The articulation of a definition is limited by a person's verbal skill, the limitations of language, and the maturity of terms in the field. These factors contribute to the divergence of definitions. Nonetheless, there are underlying themes that weave through the definitions. In particular, two key definitions of interest are:

a) Functions and behaviors at the contextual edge – though the system context may change, one aspect of systems thinking is dealing with that contextual edge

b) Interactions and how elements relate – the specific interactions and elements may change with system context, but the ability to consider interactions is developed by similar mechanisms. The primary mechanisms cited enable and encourage: Translation across contextual edges, Consideration of interactions, Higher impact learning". (Davidz, 2006)

Considering the background information in the previous sections of this chapter, a question that arises is: *Can we understand the complexity of an automobile using the systems engineering context?*

The answer is yes, and to do so the first thing to do is to understand the system architecture, the relationships and the interdisciplinary nature of a vehicle. Using systems thinking, we can visualize how things are connected, how they affect each other, estimate the effects using common sense and engineering background to design components knowing the environment they will interact with to achieve the intended function.

2.5.1 System Architecture

Understanding the system architecture, components, interconnections and relationship is a very important step that is often ignored in the industry. When the essential understanding of the vehicle architecture is not fully understood during the requirement and the design phases, several defects infiltrate through the design process, resulting in problems that are found way too late by customers in the field. Two important aspects, Form and Function, are needed to understand the system architecture.

2.5.2 System Architecture – Form

The physical domain is the first abstraction where we can identify the "Form" of the components of a vehicle. The anatomy of a vehicle is illustrated in Figure 15, where we see the components broken down by mechanical domains.

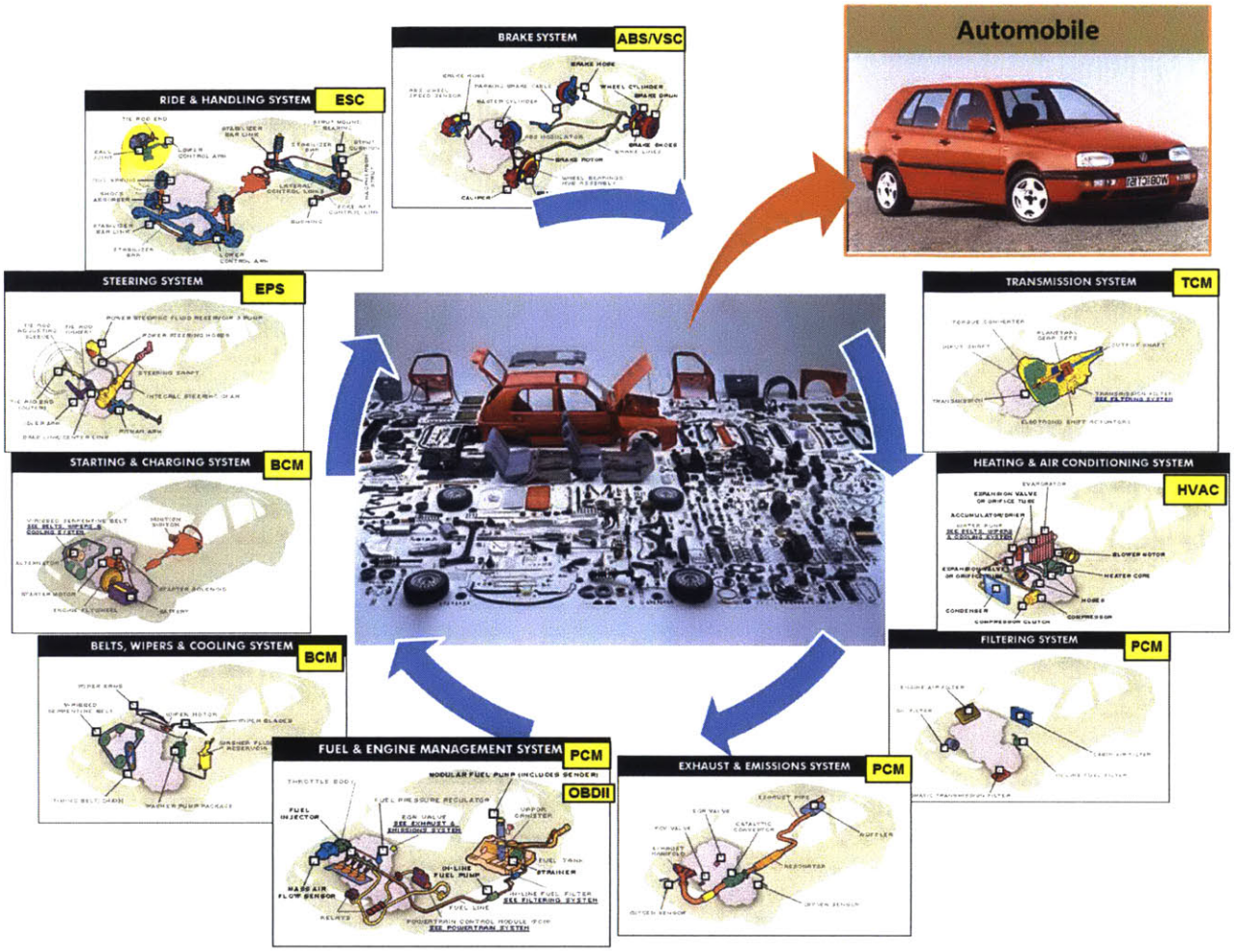


Figure 15 The physical / mechanical view of an automobile [11]

2.5.3 System Architecture – Function

Considering a second abstraction that can help understand the architecture, we can use the "Function" view of the automobile. Given that the physical components must perform certain functions as part of the overall system purpose, and the fact that electronic controls are used to control the whole system, we can relate the "Form" to the "Function" in Figure 15, where we can see an electronic module assigned to each domain.

2.5.4 Systems Engineering

With the view introduced in the previous sections we have a broader picture of the automobile taxonomy and its context, which should be similar among the different automakers worldwide, using holistic thinking we can understand the components and

[11] The physical / mechanical view of an automobile illustration on this figure was constructed with graphics from the following sources:
http://socutecrafts.blogspot.com/2013/01/street-racing-import-tuning-gallery.html
http://www.economymufflerandbrake.com/services.html
http://www.autoplenum.de/Auto/VW/Golf+3/Bild-id343193.html

the interconnections using System Architecture and Model Based Systems Engineering (MBSE) and with the OPM model , that will be discussed in chapter 4, we will be able to see the "Form" (components) and the "Function" at the highest level in a complete view.

Two more elements of systems engineering play a critical role in automobile design: Project Management and System Dynamics.

2.5.5 Project Management

During the execution of a program, the decisions from the control stakeholders affect the entire vehicle product development process, to understand more about the implications. As shown in Figure 16 and according to O. De Weck and J. Lyneis *"One of the most fundamental concepts in project management is the 'iron triangle'. This refers to the fact the big three considerations in projects: Scope, Cost, Schedule are in tension with each other. Achieving more, usually takes more time and/or cost. Reducing costs for a project often means having to de-scope and so forth. There is debate in the project management community whether risk ought to be shown as a separate category, or whether it is simply a qualifier on Scope, Cost and Schedule. Regardless, risk is essentially the likelihood that scope will not be achieved according to plan. This captures the potential downside that budgeted costs will be overrun or that the scheduled finish date and intermediate milestones will slip"* (De Weck & Lynesis, Successfully Designing and Managing Complex Projects, 2013, p. 102).

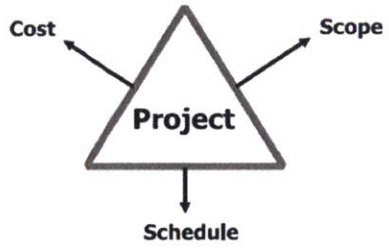


Figure 16 Iron triangle of Project Management: Scope, Cost and Schedule [12]

"The iron triangle deserves its name because the tension is generally inescapable. Risk is increased sharply if all three dimensions are constrained together. One should not accept to lead a project that is over-constrained in all three dimensions (over-scoped, under budget, deadlines too tight) and therefore appears infeasible because this is a recipe for failure. This issue will be discussed extensively in later chapters. Sometimes firms propose infeasible projects in order to win contracts. While understandable from a business perspective, this is not good Systems Engineering"

The effects of lack of balance in the iron triangle severely impact the cost of a project, as illustrated in Figure17, where we see the increase in cost of defects discovered at different stages of the development process.

[12] Image extracted from the book "Successfully Designing and Managing Complex Projects", Fig. 8.1. Iron triangle of Project Management: Scope, Cost and Schedule , (De Weck & Lynesis, Successfully Designing and Managing Complex Projects, 2013, p. 102)

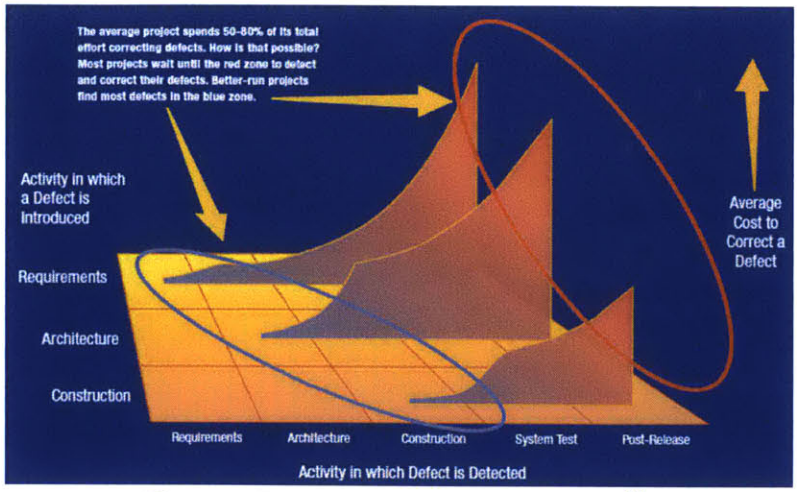


Figure 17 The cost of defects in systems [13]

This phenomenon of project management results in fire-fighting, which is a reaction often followed by companies to fix problems faced by the organizations, and requires analysis of the interaction between the stakeholders. Using system dynamics view will help to understand further.

2.5.6 System Dynamics

In the ideal world, the design of an automobile is a very straight forward process; however, there are other factors to consider. While external events are a fact of life on projects, project performance problems are fundamentally dynamic problems that result from attempts to manage in the face of change and uncertainty.

According to (De Weck, Lynesis, & Moser, 2013) *"We often find poorly defined project objectives, shifting system requirements, manager's mental models, etc. Typical tools (computer models) are not helpful in understanding dynamics.* A common mistake in project management is that teams attribute problems to external factors, view a project statically (no interaction, no feedback), projects are treated as unique."

This view correlates with the practices I experienced while working in this industry. Project management relies on having a balance between cost, scope and schedule. The ultimate phenomena observed because of a shift on the iron triangle are unexpected troubles.

To understand causes and effects of iron triangle shifting, System Dynamics can be used. Jay Forrester developed the Systems Dynamics approach at MIT in the 1960's. For several decades Forrester applied Systems Thinking to business management, society and politics, maintaining throughout, that system dynamics is the necessary foundation essential for effective thinking about systems. According to (Forrester, 2010, p. 1), *"Understanding systems is crucial to improving the organization of schools and to modernizing material that students learn. But how is one to think about systems? Our*

[13] Image source: http://www.construx.com/Resources/Posters/Software_Development%E2%80%99s_Defect_Cost_Increase/

educational, social, and economic systems are far more complex than the technological systems faced by engineers. Even with the simpler systems of chemical refineries and space flight, an engineer would never try to design by simply thinking and depending on intuition. The engineer would use computer simulations to anticipate the behavior of a design, and would build prototype systems to demonstrate performance."

System dynamics is a methodology for understanding behavior over time, consider that all dynamics are driven by accumulation processes and feedback processes that are represented by stocks and flows, as illustrated in Figure 18.

Figure 18 The Stocks and Flows representation used in System Dynamics [14]

The problems faced in project management can be analyzed with the help of feedback loops. Consider that for a given problem, certain actions are taken as part of the solution, furthermore the solution itself led to additional situations and decisions *"a problem presents itself as a discrepancy between an important goal and the current situation. Those responsible for achieving the goal arrive at a solution in the form of a decision leading to action and results that change the current situation"* (Reynolds & Holwell, 2010, p. 30), this is illustrated in Figure 19, where the change to the current situation creates other situations that require other decisions, that ultimately will affect the original situation and therefore the original discrepancy.

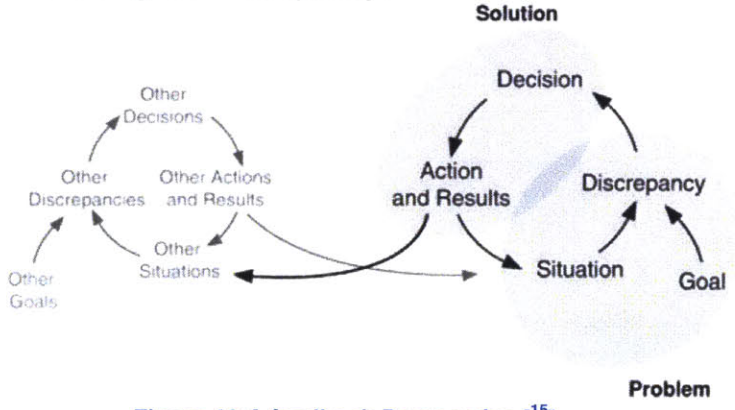


Figure 19 A feedback Perspective [15]

Using the System Dynamics methodology can help understand the discrepancies that present themselves during execution of a project, De Weck and Lyneis introduced a study applicable to project management where they discussed undiscovered re-work in projects.

In addition to schedule and budget over-runs, Dr. Lyneis presented further analysis applied to the major phases of an automobile design project, where the dynamics of the

[14] Image source (De Weck & Lynesis, Lecture 6 Introduction to Project Dynamics, 2013, pp. 31,32,33,34)
[15] Image source Figure 2.4 A feedback perspective (Reynolds & Holwell, 2010, p. 31)

engineering tasks for the Requirements, Design and Build/Test phases were analyzed (see Figure 20).

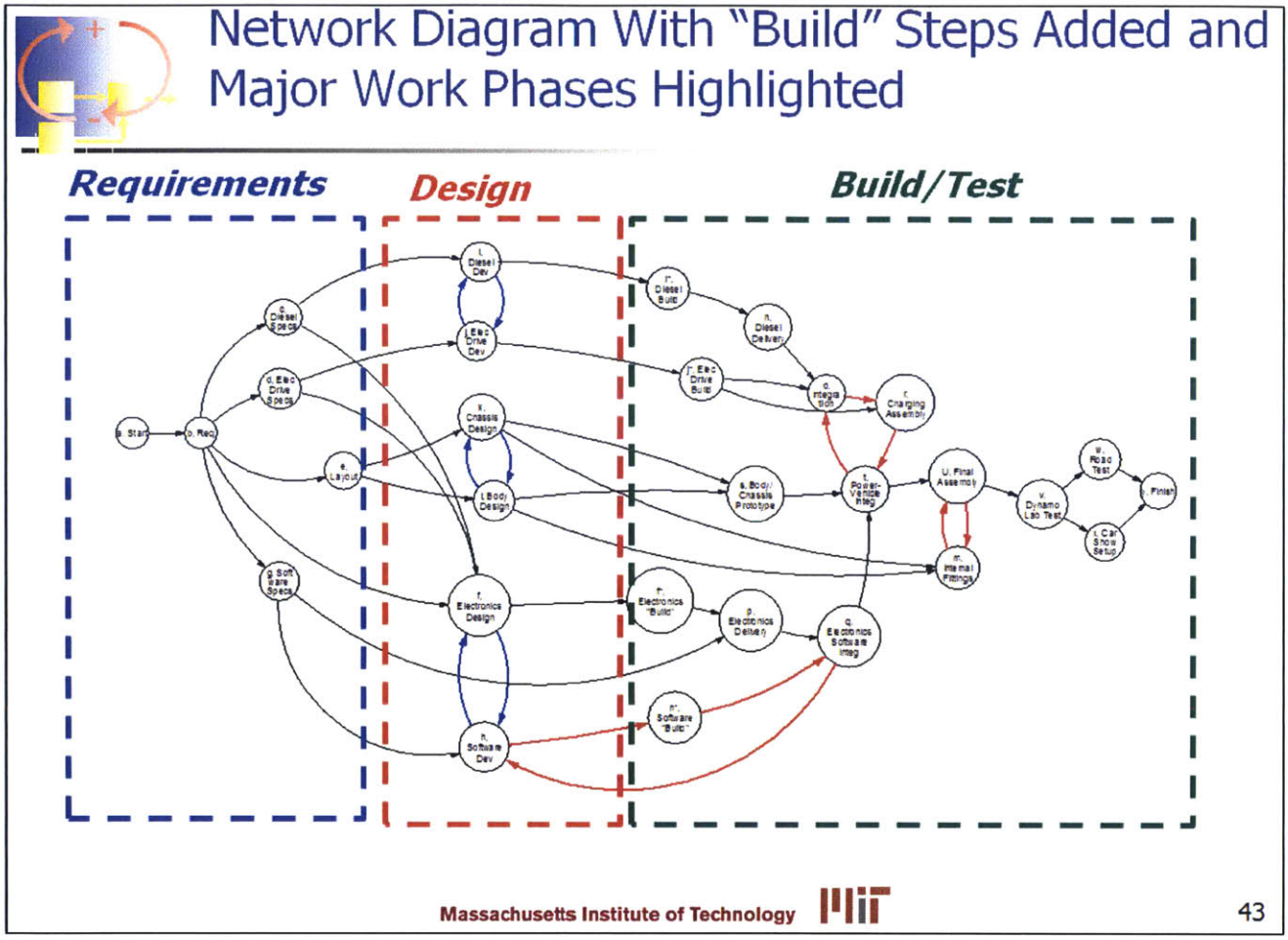


Figure 20 The dynamics in a vehicle project [16]

Consider that the re-work cycle, illustrated in Figure 21, is present at every stage during the execution of these major phases of the product development.

[16] Image source: (De Weck & Lynesis, Lecture 7 The re-work Cycle, 2013, p. 43)

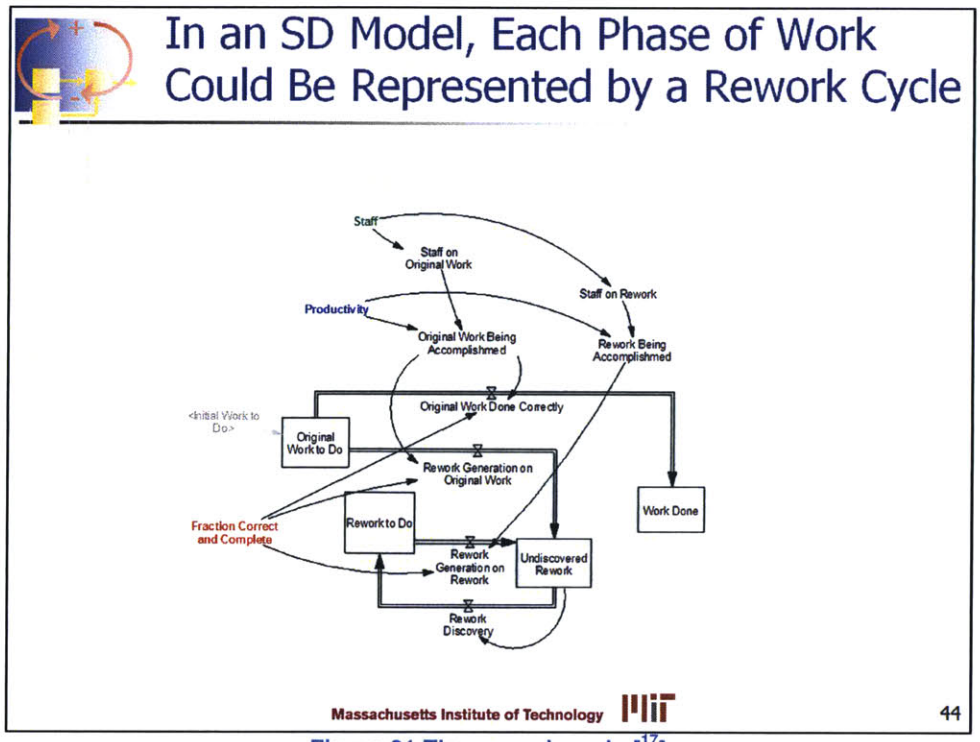


Figure 21 The re-work cycle [17]

Behind the dynamics for the major phases there are re-work cycles that affect the project performance, as illustrated on Figure 22.

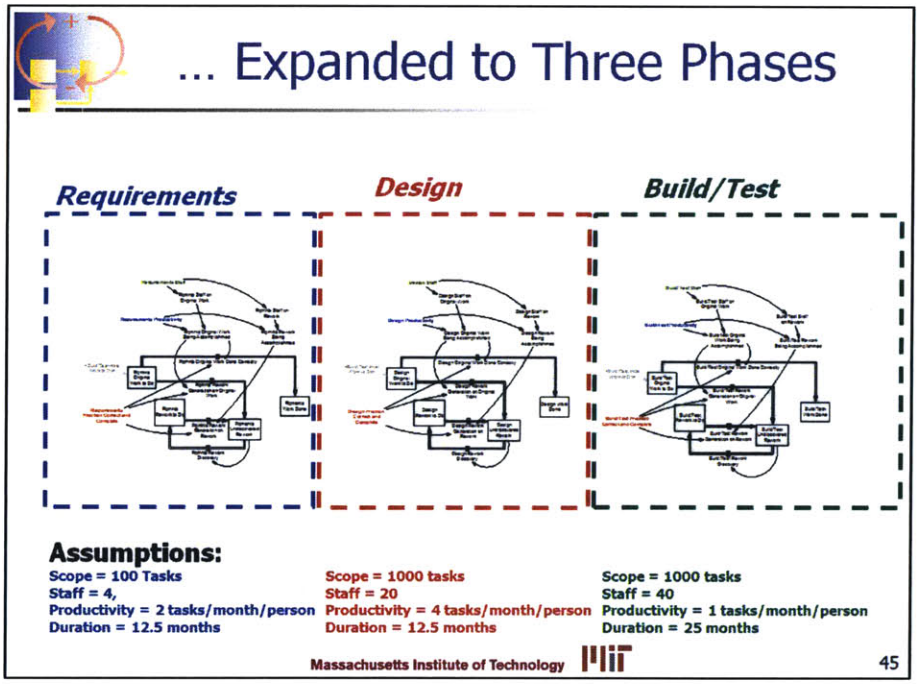


Figure 22 The expanded view of the re-work cycle in the major project phases [18]

[17] Image source: (De Weck & Lynesis, Lecture 7 The re-work Cycle, 2013, p. 44)
[18] Image source: (De Weck & Lynesis, Lecture 7 The re-work Cycle, 2013, p. 45)

A holistic view for the automobile design shall consider that the dynamics involving the product development process and the rework cycles affect the overall outcome; hence these are critical for the system design.

2.6 Conclusion of the Automobile Systems Historic Background

Every man-made system required innovation to generate new ideas and production of tools that stimulated technological advancements. The evolution of products due to these advancements has been a signature of the human ingenuity for centuries.

In this chapter we analyzed the automobile and how the product changed to adapt incumbent technologies to improve the performance. We still need a holistic view to understand that automobiles shifted from sole discipline parts to multi-disciplinary groups of parts forming subsystems that today make up an automobile. This view of the automobile shall consider all the elements discussed in this chapter, the historic and technological background, the product development process and the systems engineering elements that are required to design this products.

Learning and understanding the historic and technological background helps determine where we were, where we are and where we need to be. The Product development process, stakeholders involved on the automobile business, project management and dynamics of the rework cycles are critical components that normally pose threats to any project execution.

System engineering elements of system architecture, complexity and relationships between components, need to be understood in order to establish the intended functionality and establishment of system boundaries. The design of an automobile shall be done using Systems Engineering, as the intrinsic nature of the product and the amount of disciplines required to collaborate in this business are clear indicators that mandate transforming our traditional views.

Model Based Systems Engineering (MBSE) methodology for software development is discussed in the next two chapters.

CHAPTER 3 MODEL BASED SYSTEMS ENGINEERING IN THE AUTOMOTIVE INDUSTRY

3.1 Introduction

The historic background discussed in Chapter 2 provided insights about the automobile industry in the U.S. and the transition from pure mechanical systems to complex systems. Another important technological innovation that contributed to the evolution in this industry was the invention of the personal computer.

Beyond the use electronics for components and electronic control units (ECU's), the automotive OEM's introduced computers to aid in the design activities. This incumbent technology affected the design process itself. This industry witnessed a dramatic shift over the last 60 years, and today the design of a new car requires the use of computers at every stage of the Product Development Process. The use of computers as the main design tool caused companies to face greater challenges to succeed in their mission to stay in business.

Behind all of the computerized tools used to design the components of an automobile, the engineers and designers use models to incorporate all of the parametric characteristics of a part/component and evaluate the performance in the virtual world, hence the importance of models posit a great value added in automotive design. This chapter will be dedicated to discuss the importance of Model Based Systems Engineering (MBSE) in automotive design.

3.2 The automotive engineering transition to computerized tools

Ever since the birth of the automotive industry in the U.S., mechanical engineering was employed to design parts, following a traditional manual process for about 60 years. However, after the introduction of the computers, things began to change. One of the early adopters of computerized tools was General Motors [19], who by the middle of 1950's implemented analog computers at the Milford Proving grounds ([45]), marking the beginning of a new product development era that changed the way an automobile was designed. By the 70's, GM's drafting rooms were replaced by CAD terminals (see Appendices A, B, C, E).

By 1985, the design capabilities at GM covered a wide spectrum of engineering analysis techniques (refer to Appendix B - The growth of math based simulation methods in GM). Therefore, the use of the computer tools for CAD, CAM and CAE resulted in the use of computerized representations, or *"models"*, which added flexibility, reliability and robustness to the automotive design (refer to Appendix E), and by the end of 1990's computerized tools transformed the way an automobile was designed.

[19] It is important highlight that at the same time this automaker implemented the use of computer based tools, other OEM's had similar strategies not only in the U.S. but in other countries in Europe and Asia to implement computer based tools to aid on the engineering of automobile parts.

3.2.1 What is a Model

There are several definitions of a model, for example the Merriman Webster dictionary definition is: *"A description or analogy used to help visualize something that cannot be directly observed".* A more refined definition is: *"A model is an abstraction of a system aimed at understanding, communicating, explaining or designing aspects of interest of that system"* (Dori, Lecture #2 MBSE Introduction, 2014, pp. 42,45). According to Embley and Thalheim, *"Models are created to achieve different purposes and we build them to increase our understanding of something complex, for example: a) Analysis of an application domain, b) Constructing of a system, c) Communicating about an application, d) Assessment, e) Governance"* (Embley & Thalheim, 2011, p. 543).

There are numerous approaches and methods for the use of models, Dori (Dori, Lecture #2 MBSE Introduction, 2014, p. 45), the categorization of a model can be made based on its type: physical, graphical, mathematical, or natural language.

Leveson highlighted a constraint embedded within a model design: *"... models simplify the thing being modeled by abstracting away what are assumed to be irrelevant details and focusing on the features of the phenomenon that are judged to be the most relevant. Selecting some factors as relevant and others as irrelevant is, in most cases, arbitrary and entirely the choice of the modeler. That choice, however, is critical in determining the usefulness and accuracy of the model in predicting future events"* (Leveson, 2011, p. 15)

After the introduction of computers, the use of the four types of models to aid in engineering design has been a common practice in the auto industry, and this is known as Model-Based Engineering (MBE) or Model-Driven Engineering (MDE).

3.2.2 MBSE

According to INCOSE, *"Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. MBSE is part of a long-term trend toward model-centric approaches adopted by other engineering disciplines, including mechanical, electrical and software. In particular, MBSE is expected to replace the document-centric approach that has been practiced by systems engineers in the past and to influence the future practice of systems engineering by being fully integrated into the definition of systems engineering processes"* (INCOSE, SYSTEMS ENGINEERING VISION 2020, 2007, p. 15).

While electronic control modules follow the software development process (SDP), the application of the MSBE methodology for the automobile development design cycle shall accommodate the SDP and the vehicle development process (VDP). Figure 23 illustrates the lifecycle stages in the automobile design domain (a zoomed in view can be found in Appendix F).

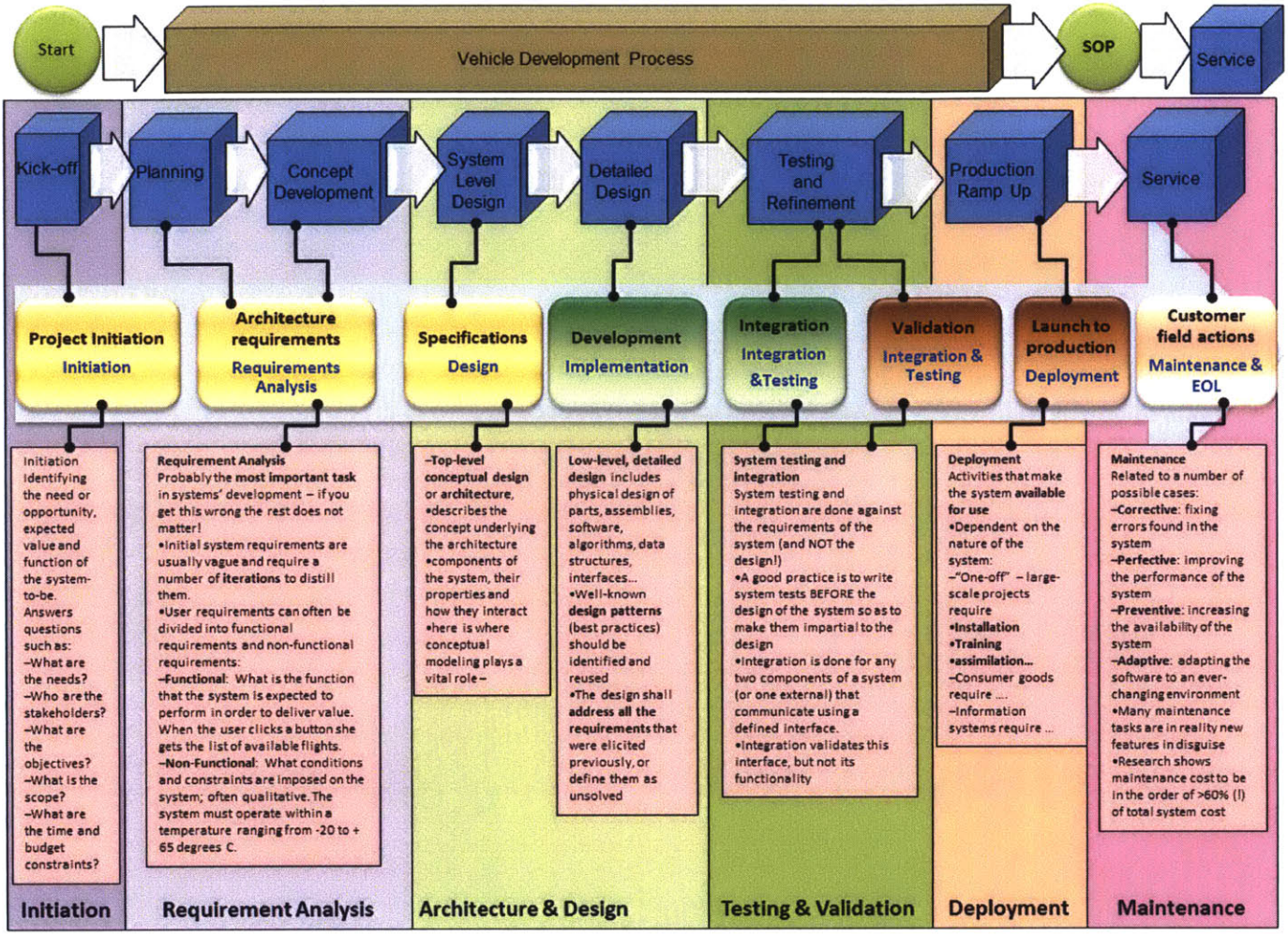


Figure 23 The lifecycle stages in the automobile domain [20]

3.2.3 MBSE design philosophy in the auto industry

There are several examples of MBSE applied to the automotive engineering. A remarkable one is the accelerated engineering capabilities that GM obtained as a result of a project named "Trilby" (see Appendix E and D). This example of early application of model based design and the usefulness and accuracy of models played a very important role in the quality of the automobiles produced by this automaker.

While working for three major automakers, I collected experiences from different MBE methodologies used for ECU's software design, illustrated in Table 1. The inclination to use a specific software modeling tool is driven by intellectual property ownership, requirements management, and the vehicle domain.

[20] This illustration was constructed based on the MBSE class Lecture 2 (Dori, Lecture #2 MBSE Introduction, 2014) and the Vehicle Development Process discussed on section 2.4.

Software Model / Design Environment	Engine Control		Transmission Control		Brake Control		Body Control		Air Bag Control		Instrument Cluster		HVAC Controls		Power Steering		Audio & Telematics	
	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned	OEM owned	Supplier owned
MatLab Simulink	✓		✓			✓		✓		✓		✓		✓		✓		✓
UML & SYSML							✓				✓						✓	
Rational Rhapsody							✓				✓						✓	
Rational StateMate	✓		✓				✓				✓						✓	

Table 1 MBSE Design Tool use by ECU domain & IP ownership

Traditionally, software design for ECU's used for engine control and transmission controls are designed with IP owned by the automaker and document centric requirements. The preferred tool for the software modeling behavior is Simulink. For the rest of the control modules, traditionally the supplier owns the intellectual property, however, the MBSE philosophy introduced with UML/SYSML led to a shift in the modeling tools and the intellectual property ownership. Today several ECU's are designed using a model centric requirements management philosophy, and the use of UML/SYSML expanded the capabilities while maintaining compatibility with Simulink.

3.2.4 Measurement of systemic faults from MBE and MBSE designed vehicles

The trace of systemic issues discovered in a project that followed MBE methodology by one automaker, referred to as "OEM #A", during the vehicle design lifecycle is illustrated in Figure 24. The vehicle architecture contained 19 different ECU's with a medium level of complexity.

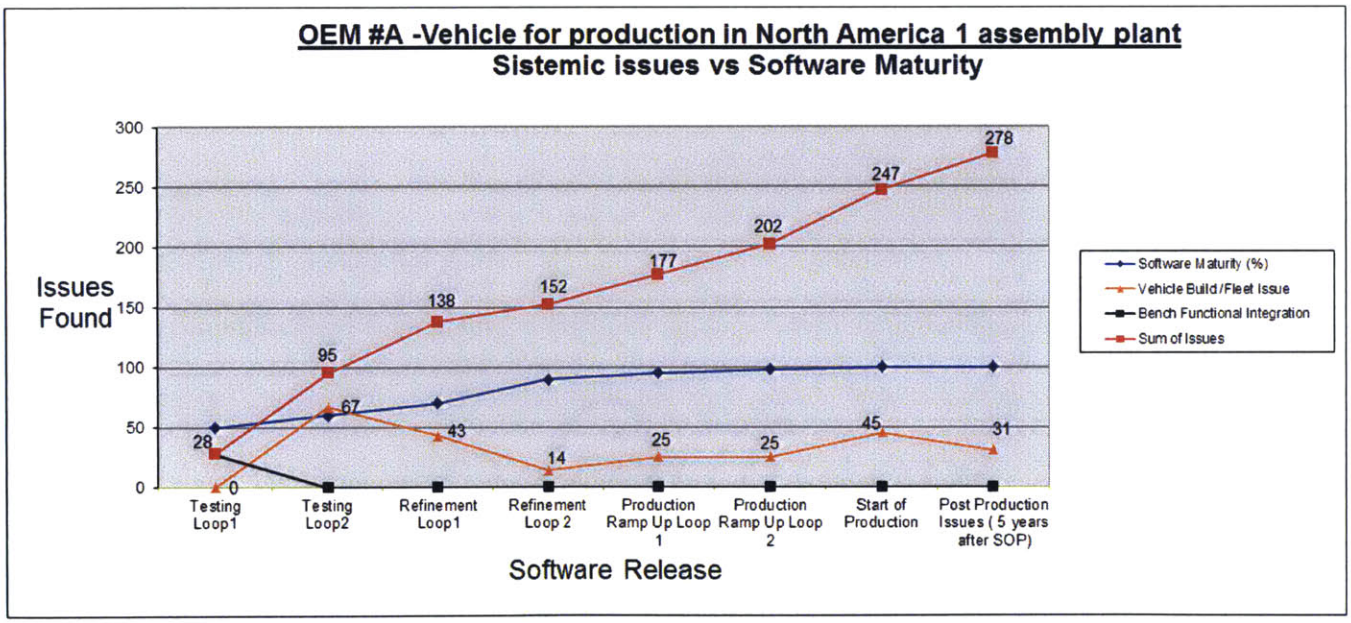


Figure 24 Results of using traditional MBE[21]

The majority,(90%) of problems were found in physical vehicles, and the rest (10%) were found in test benches or other simulation tools. However, after 5 years of production

[21] The results shown on this picture illustrate the trend of undiscovered problems while following the MBE methods, the information for the post production was extracted from the NHTSA data base from the (NHTSA, 2014)

there were 31 undiscovered problems, illustrated in Figure 25, which resulted on a 12.6% increase

Software Release	Testing Loop1	Testing Loop2	Refinement Loop1	Refinement Loop 2	Production Ramp Up Loop 1	Production Ramp Up Loop 2	Start of Production	Post Production Issues (5 years
Bench Functional Integration	28	0	0	0	0	0	0	0
Vehicle Build /Fleet Issue	0	67	43	14	25	25	45	31
Sum of Issues	28	95	138	152	177	202	**247**	278
Ideal non-vehicle tested	0	95	43	14	25	25	1	1
Software Maturity (%)	50	60	70	90	95	98	100	100

Issues found during testing and refinement phase		
Found in:	Issues found during design	Participation
Test bench	28	10.07%
Vehicle	219	78.78%
Subtotal	**247**	

Post Production Issues (5 years after SOP)	
Undiscovered issues	Delta
31	**12.6%**

Figure 25 Vehicle System Problems during lifecycle - MBE method applied

Similarly, figures 26 and 27 illustrate the results of using MBSE methodology in a global program featuring 34 ECU's since the early conception of the project. In this case 71% of the problems were found using test bench along with a systematic test plan, while 23% of the problems were found in the vehicles. Furthermore, after 5 years in the market, there were 29 undiscovered problems or 5.8% increase.

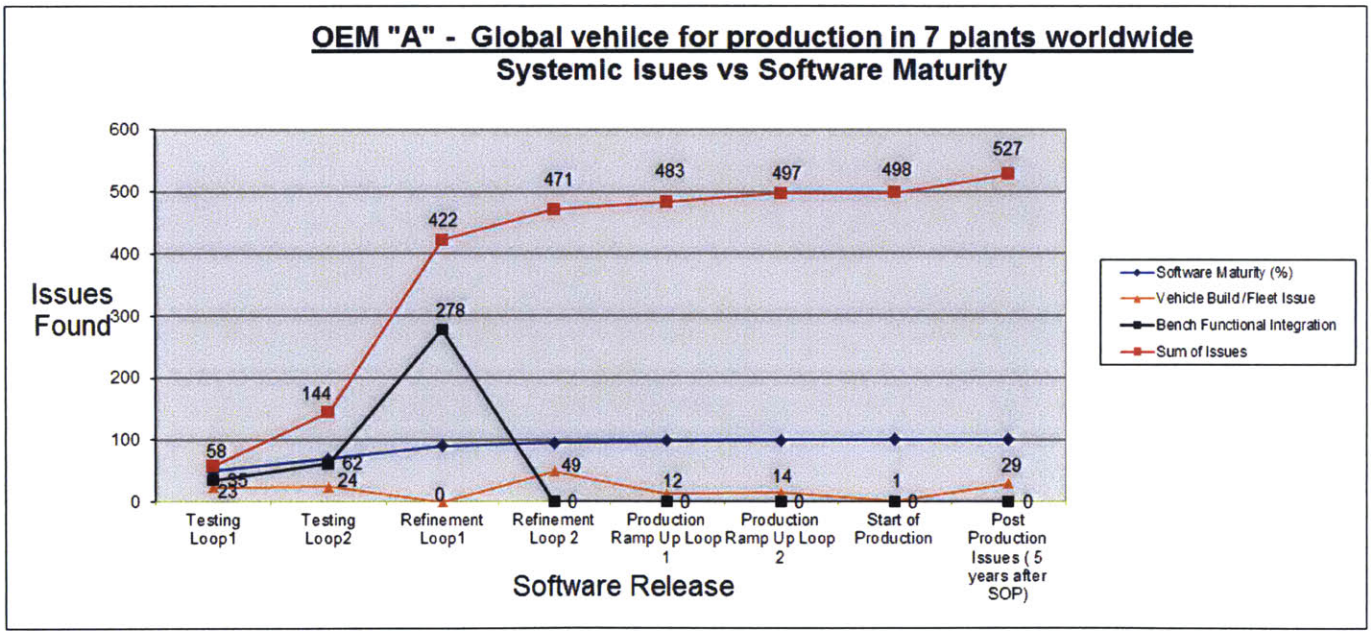


Figure 26 Results of the use of MBSE methodology [22]

[22] The results shown on this picture illustrate the trend of undiscovered problems while following the MBE methods, the information for the post production was extracted from the NHTSA data base from the (NHTSA, 2014)

Software Release	Testing Loop1	Testing Loop2	Refinement Loop1	Refinement Loop 2	Production Ramp Up Loop 1	Production Ramp Up Loop 2	Start of Production	Post Production Issues (5 years
Bench Functional Integration	35	62	278	n/a	n/a	n/a	0	0
Vehicle Build /Fleet Issue	23	24	n/a	49	12	14	1	29
Sum of Issues	58	144	422	471	483	497	**498**	527
Ideal non-vehicle tested	58	86	278	49	12	14	1	1
Software Maturity (%)	50	70	90	95	98	99	100	100

Issues found during testing and refinement phase		
Found in:	Issues found	Participation
Test bench	375	71.16%
Vehicle	123	23.34%
Subtotal	**498**	

Post Production Issues (5 years after SOP)	
Undiscovered issues	Delta
29	**5.8%**

Figure 27 Vehicle System Problems during lifecycle - MBSE method applied

The previous data is indicative of the benefits of using MBSE during the earliest phases of product design, and the improvements obtained. This data correlates with the six sigma philosophy followed in the auto industry. Furthermore, the 5 year data from the NHTSA data base (NHTSA, 2014) also indicated a specific breakdown of faults illustrated in Figure 28.

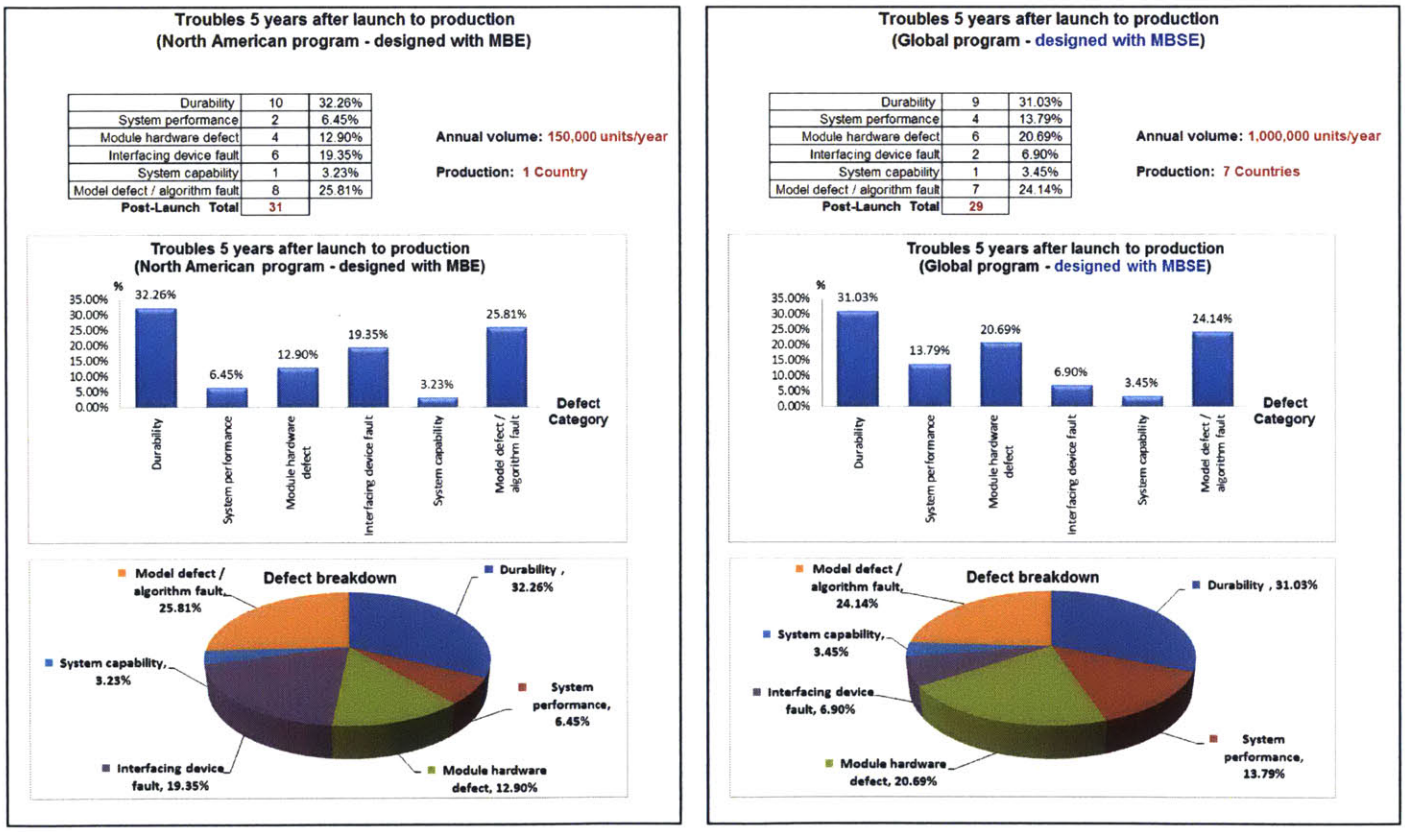


Figure 28 NHTSA 5 year data comparisions between a program designed with MBE & MBSE

The faults reported by vehicle owners have a relationship to casual factors of system defects. In the following section, the analysis of system casual factors will help understand systemic faults of automobiles.

3.2.5 Casual Factors Defects of ECU's in the automotive industry

The computers used in every vehicle subsystem in the automotive industry feature a control system that uses software. According to Leveson, systems follow a standard control loop like the one illustrated on Figure 29.

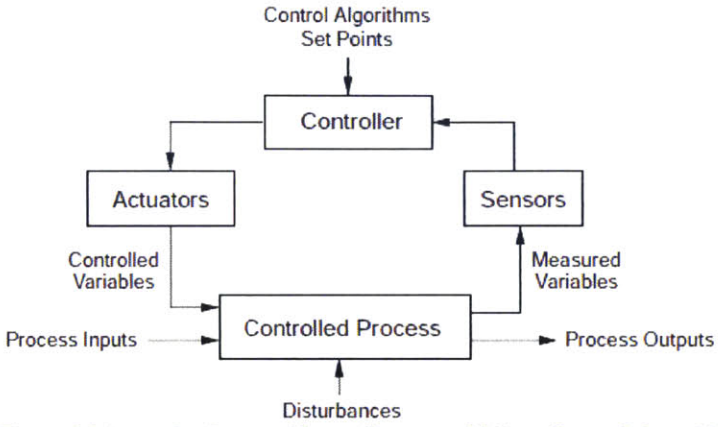


Figure 29 A standard control loop, (Leveson, 2011, p. figure 3.2 pp 66)

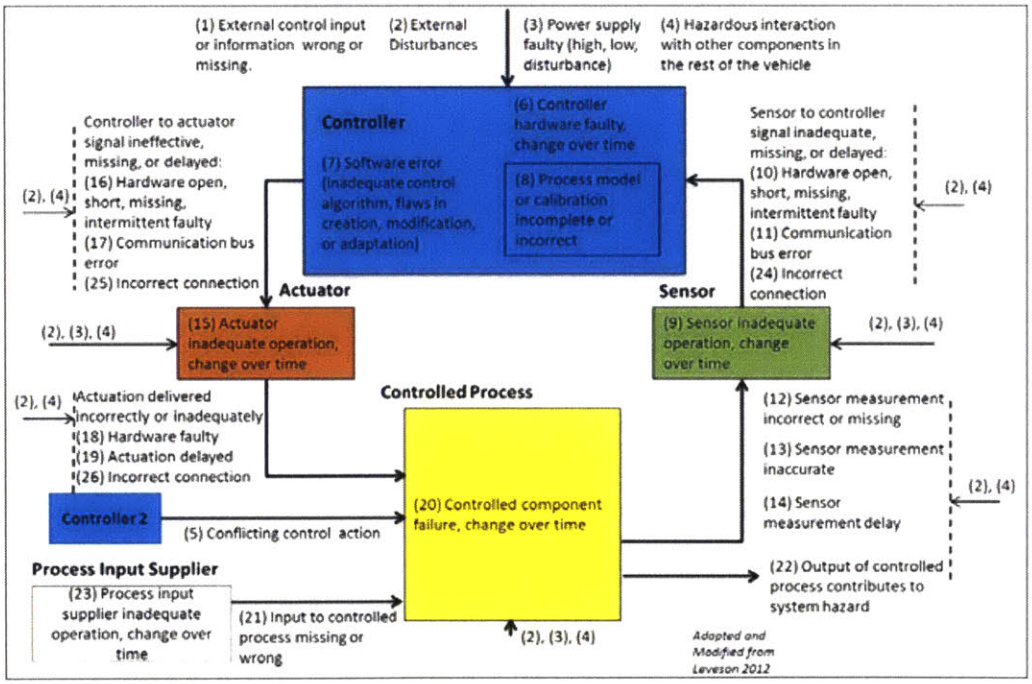


Figure 30 Causal Factor Diagram suggested in the Safety-HAT[23]

[23] Picture extracted from the Transportation Systems Safety –HAT user's guide (U.S. Department of Transportation, 2014, pp. 61, figure 57),

Furthermore, the problems found by vehicle owners also relate to the concept of software-related accidents. Hence, since the ECU's used in the automotive industry are designed with control system principles, these components feature similar faults as suggested in the Transportation Systems Safety-HAT work (U.S. Department of Transportation, 2014, pp. 61, figure 57). Figure 30 contains a summary of the potential faults of a system.

The overall design process requires analysis of the constraints that affect all levels of the vehicle design and ultimately the performance. The next chapter includes design guidelines that address those constraints while using the OPM graphical MBSE tool.

CHAPTER 4 PROPOSED MBSE STRATEGY FOR BODY CONTROL MODULES SOFTWARE DEVELOPMENT

4.1 Introduction - Object Process Methodology (OPM)

According to (Estefan, 2008, p. 43), there are six leading MBSE methodologies that are commercially available [[24]], with unique characteristics and features. In this section we focus on guidelines to design a model of a Body Control Module (BCM) for the automotive domain using Object Process Methodology (OPM). Among the six MBSE methodologies, the modeling language used during this research is Object Process Methodology. According to (Dori, 2002) *"Object-Process Methodology (OPM) is a holistic, integrated approach to the design and development of systems in general and complex dynamic systems. OPM is a formal yet intuitive paradigm for systems architecting, engineering, development, lifecycle support, and evolution. It has been used for modeling complex systems, both natural and artificial, where artificial ones might comprise humans, physical objects, hardware, software, regulations, and information. As its name suggests, the two basic building blocks in OPM are (possibly stateful) objects, i.e.,things that exist (possibly at some state), and processes, i.e., things that transform objects by creating or destroying them, or by changing their state. Objects and processes are of equal importance, as they complement each other in the single model specification of the system. Links, which are the OPM elements that connect entities, are of two types: structural (connecting objects to objects or processes to processes) and procedural (connecting objects to processes). The generic definition of these elements makes OPM suitable for modeling complex systems that comprise technology and humans. This is the type of systems that aim to deliver complex products via executing large-scale projects. OPM notation supports conceptual modeling of systems using a single type of diagram to describe the functional, structural and behavioral aspects of a system. An OPM model consists of a set of hierarchically-organized Object-Process Diagrams (OPDs) that alleviate systems' complexity. Each OPD is obtained by refining (via in-zooming or unfolding) a thing (object or process) in its ancestor OPD."*

4.2 Application of OPM to a project lifecycle in the automotive industry.

MBSE methodologies advocate a holistic view of a system, and when applied to a project with a wide systems engineering perspective such as a complex automobile architecture, they are extremely helpful to aid on the management of the system interactions on a high level.

Constraints of an automobile system present themselves as malfunctions detected by the customer, vehicle driver(s), engineers, technicians or any other person related to the

[24] The paper from Jeff A. Estefan contains a survey of the six leading MBSE methodologies used in the industry :
1-IBM Telelogic Harmony-SE,
2-INCOSE Object-Oriented Systems Engineering Method (OOSEM),
3-IBM Rational Unified Process for Systems Engineering (RUP SE) for Model-Driven Systems Development (MDSD)
4-Vitech Model-Based System Engineering (MBSE) Methodology,
5-JPL State Analysis (SA) ,
6-Object-Process Methodology (OPM)

functional elements of this systems that in some cases resulted in tragic accidents. We can argue that these constraints can be associated to errors introduced during vehicle development process, that remain undiscovered until the final product reaches the hands of the customer and are indeed produced and executed by different stakeholders involved in the design process.

Furthermore, design errors can be linked to the causal factors identified in sections 3.2.4 and 3.2.5. Problems experienced by an automobile owner can be classified into the following categories: a) System performance error, b) System capability limitations, c) Software algorithm defects, d) System components durability and e) Hardware defect. These failures present an opportunity for automobile systems design improvements.

This research is intended to help develop a context for automotive body electronic modules software design by providing a model-based framework based on Object-Process Methodology (Dori, 2002). These guidelines can help address existing problems associated with body electronic modules software development.

4.3 OPM model-based framework for body electronic modules software design

To start to lay out the framework, consider the eight steps of MBSE, illustrated in Figure 29 along with the vehicle development process followed in the automotive industry. The first two steps—project Initiation and requirements analysis, posit a high level of importance to the system design. The model-based framework is divided on three stages: 1) Definition of System Purpose and Scope, 2) System Requirements, 3) Requirement Analysis.

4.3.1 Definition of the System Purpose & High Level Requirements

The body electronics controlling system functional requirements are derived from the automobile functions and depend also on the system architecture. Figure 31 shows a simplified block diagram with the main functions of the body electronics domain. There are nine sub-functions that make it up: Vehicle Network Controlling, Energy & Charging Managing, Time and Date Managing, Vehicle Access Controlling, Starting Controlling, Safety Controlling, Security Controlling, Inputs Controlling, and Lightning Controlling.

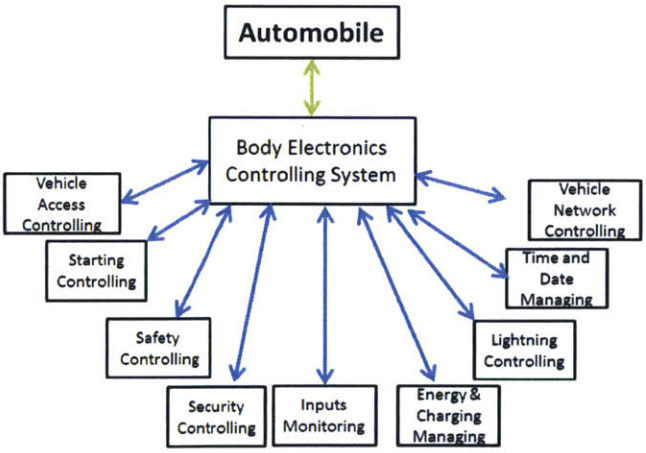


Figure 31 The Basic functions of a body electronics controlling system in an automobile

System Name: Body Electronics Controlling System

Purpose:
The purpose of the body electronics system is to monitor and control the features and functions of the body electrical and electronics domain in an automobile system. The system manages the following functional sections: Vehicle Power Mode, Energy & Charging, Access, Starting, Safety, Security, Lightning, and Network Management.

High Level functional requirements:

1. The system shall provide Vehicle Power Mode control to monitor and synchronize the vehicle network communications with the ignition key position and serve as a means to manage the vehicle operational state.

2. The system shall manage the Power Modes that depend on the ignition switch position controlled by the driver by monitoring the electrical signals that are considered state encoded values or "modes" that a driver can select by turning the ignition switch.

The desired state is communicated to all the electronic modules connected to the vehicle's serial communications network by means of a power mode message. There are four states in the Power Mode:

a) Power Mode "Off", which indicates key position off and network bus sleep mode; the sleep mode is also divided in two possible states off-sleep: a) no network activity and most of the vehicle functions are unpowered and b) off-awake, when the vehicle is remotely unlocked and any of the doors are opened waking up the network.

b) Power Mode "Accessory", indicates a transition to the accessory position where the electrical accessories are powered and operational.

b) Power Mode "Run", indicates that the driver is ready to start the engine, so the related components—fuel pump, starter, Engine Controller, etc., become prepared to crank the engine.

c) Power Mode "Crank", this mode indicates the transition of the ignition switch to crank position; it starts the process of engine cranking. During this mode, most of the electrical and electronic systems are not operational to reduce battery voltage drop and facilitate the battery power to achieve the engine cranking. Once the key is released from the crank power mode, the ignition switch goes back to the "Run" position and so is the power mode.

2. The system shall provide Energy & Charging control. This functional section monitors the energy power from the automobile's battery. It controls the charging by commanding the alternator to adjust the amount of energy supplied to keep the energy flow at the optimum level. Additionally, this section also controls the electrical load shedding to disable loads when the energy level becomes critically low to preserve the energy and extend the battery life.

3. The system shall provide Vehicle Access control: This functional section monitors and controls the components related to the vehicle access. These can be Door latches, Remote Key Fob, Key, and others. The feature facilitates the vehicle entry, securing of the doors locks, and setting the alarm on. This vehicle features can be enabled or disabled by the driver by mean of the door lock command during the power mode "Off".

4. The system shall provide Starting control. This functional section monitors and controls the components related to the engine start. These include Fuel Pump Relay, Starter Motor Relay or Crank Relay and Run Relay. The feature commands the components during the engine cranking in coordination with the power mode.

5. The system shall provide Safety control. This functional section monitors and control the components related to the vehicle safety. These include Wipers and Horn. This vehicle's features can be enabled or disabled by the driver at will during power modes "Accessory" and "Run".

6. The system shall provide Security control. This functional section monitors and controls the components related to the vehicle security, also known as the alarm system. This vehicle feature can be enabled or disabled by the driver by mean of the door lock command during the power mode "Off".

7. The system shall provide Lightning control. This functional section monitors and controls the components related to the vehicle lightning: Interior or Exterior Lights. This vehicle's features can be enabled or disabled at will by the driver during power modes "Accessory" and "Run".
Network Management: This functional section monitors and controls the body control module network activity for any of the power modes selected by the driver.

8. The system shall communicate the functional information to the modules participating in the vehicle network, by CAN functional messages following the CAN protocol, the functional messages include all the operational data for the sub-domains.

9. The system shall support network diagnostics services following the ISO 14229 standard.

10. Each sub-function shall monitor and determine the status of the units or processes it is responsible for, monitor incoming messages from the vehicle via the serial data network, update the operational data set, command the actuators related to the function and finally process any serial data updates to be communicated to the rest of the vehicle.

Unmet Needs

The following are the requirements that will allow the body electronics controlling system to feature flexibility for design, maintenance, software updates, troubleshooting, etc.

For the vehicle network environment:

1 The body electronics controlling system shall be designed considering that the vehicle requires to be compliant to the ISO 14229 standard, refer to (ISO-14229, 2014),which features a communication port (OBDII) that provides access to the vehicle's network, allowing communication with any ECU connected to it.

2 The body electronics controlling system shall be diagnosable using the diagnostic services defined in the ISO14229 standard, which includes Monitoring of Data Trouble Codes (DTC), System functionality upgrades using a software download service, Configuration of specific vehicle features that can be enabled or disabled via software by a memory write service [[25]]. These services facilitate the diagnosing of the vehicle during maintenance and provide the benefit of software updates outside of the manufacturing facility.

3 The body electronics controlling software shall be divided into separate modules following the ISO/OSI Seven Layer model.

   a. The body electronics controlling software shall be designed using the CAN solutions developed by Vector CAN Tech [[26]], which provides the drivers and software templates to develop the application software.
   b. The body electronics controlling software shall implement the CAN communication messaging data base (CAN DBC) defined by the OEM in order to ensure compatibility with the vehicle network.
   c. The interfaces to the vehicle components shall be clearly defined and consider standardized input and output circuit designs followed by the OEM to allow the compatibility with the existing company standards.
   d. The body electronics controlling system shall feature diagnostics for the interfaces following the OEM company strategies.

For the software design

4 The body electronics controlling software modules shall be coded using a model-driven approach to allow the modelers to modify the related function in the model without affecting other modules.

5 The body electronics controlling software design shall be performed using coding standards defined in MISRA [[27]]

---

[25] The list of all the services is can be found at https://www.iso.org/obp/ui/#iso:std:iso:14229:-5:ed-1:v1:en
[26] The Vector CAN Tech solutions can be located at : http://vector.com/vi_can_solutions_en.html
[27] The MISRA standards can be located at http://www.misra.org.uk/MISRAHome/WhatisMISRA/tabid/66/Default.aspx

6 The body electronics controlling system shall support upgrades and changes in the hardware and/or in the operating system.

a. The body electronics controlling software shall be coded with platform- and operating-system-independent programming languages.
b. The update of body electronics controlling software due to hardware change shall affect the hardware related modules only and not propagate to other modules and/or processes.
c. The body electronics controlling software shall be backwards compatible. In case a software update does not perform as expected, the previous version could be reprogrammed back to previous software versions.

4.3.2 System Model

The system requirements are divided into three groups. The first group is **the automobile context**. The second groups is the **Body Electronics Control System Requirements - Physical Context**, which includes the ECU hardware general requirements that are derived from the vehicle architecture. These provide the conventional body control module hardware scope, which is a representation of the "Form". The third group is the **Functional Requirements**, which include the features and functions of the body electronics domain in the automobile, or a representation of the architecture "Functions".

4.3.2.1 The automobile context

At the highest level shown in Figure 38, the purpose of the system is to allow a Driver to handle the automobile's Body Electronics Features and Functions. The Automobile System shall consist of two high level components, Physical Components that form the system, and the Functional Domains, which consist of the Body Electronics Control System that monitors and controls the vehicle's body electronic features and functions.

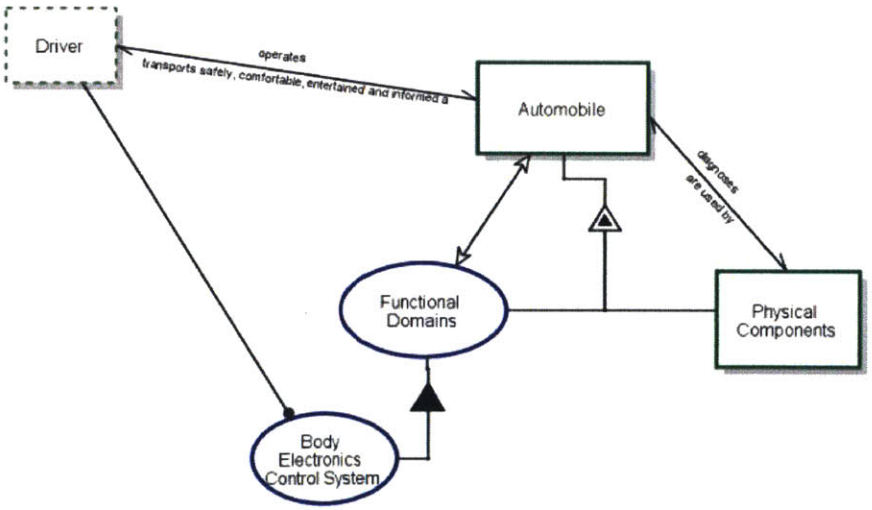


Figure 32 The Automobile context view.

Automobile is physical.
Automobile exhibits Physical Components, as well as Functional Domains.
Physical Components is physical.
Physical Components are used by Automobile.
Functional Domains consists of Body Electronics Control System.
Functional Domains affects Automobile.
Automobile diagnoses Physical Components.
Automobile transports safely, comfortable, entertained and informed a Driver.
Driver is environmental and physical.
Driver operates Automobile.
Driver handles Body Electronics Control System.

4.3.2.2 Automobile Physical Components

The physical components that form the automobile are depicted in Figure 33, and the groups of parts that form the different vehicle subsystems are indicated as well.

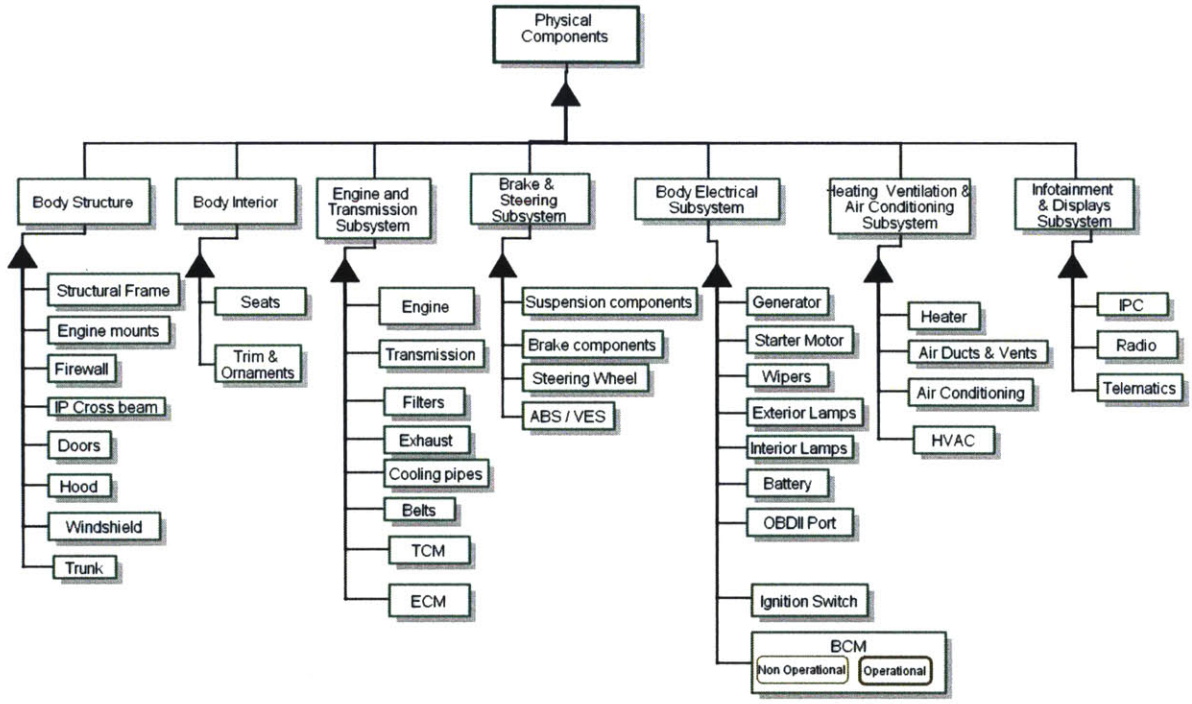


Figure 33 Automobile Physical components context

Physical Components is physical.
Physical Components consists of Body Structure, Body Interior, Engine and Transmission Subsystem, Brake & Steering Subsystem, Body Electrical Subsystem, Heating Ventilation & Air Conditioning Subsystem, and Infotainment & Displays Subsystem.
Body Structure is physical.
Body Structure consists of Structural Frame, Doors, Windshield, Hood, Trunk, Firewall, Engine mounts, and IP Cross beam.
Structural Frame is physical.
Doors is physical.
Windshield is physical.
Hood is physical.
Trunk is physical.
Firewall is physical.
Engine mounts is physical.
IP Cross beam is physical.
Body Interior is physical.
Body Interior consists of Seats and Trim & Ornaments.
Seats is physical.
Trim & Ornaments is physical.
Engine and Transmission Subsystem is physical.
Engine and Transmission Subsystem consists of Engine, Transmission, Filters, Exhaust, Cooling pipes, Belts, ECM, and TCM.
Engine is physical.
Transmission is physical.

Filters is physical.
Exhaust is physical.
Cooling pipes is physical.
Belts is physical.
ECM is physical.
TCM is physical.
Brake & Steering Subsystem is physical.
Brake & Steering Subsystem consists of Brake components, Suspension components, Steering Wheel, and ABS / VES.
Brake components is physical.
Suspension components is physical.
Steering Wheel is physical.
ABS / VES is physical.
Body Electrical Subsystem is physical.
Body Electrical Subsystem consists of Generator, Starter Motor, Wipers, Exterior Lamps, Battery, Interior Lamps, BCM, OBDII Port, and Ignition Switch.
Generator is physical.
Starter Motor is physical.
Wipers is physical.
Exterior Lamps is physical.
Battery is physical.
Interior Lamps is physical.
BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.
OBDII Port is physical.
Ignition Switch is physical.
Heating Ventilation & Air Conditioning Subsystem is physical.
Heating Ventilation & Air Conditioning Subsystem consists of Heater, Air Ducts & Vents, Air Conditioning, and HVAC.
Heater is physical.
Air Ducts & Vents is physical.
Air Conditioning is physical.
HVAC is physical.
Infotainment & Displays Subsystem is physical.
Infotainment & Displays Subsystem consists of IPC, Radio, and Telematics.
IPC is physical.
Radio is physical.
Telematics is physical.

4.3.2.2.1 Body Electronics Control System Requirements - Physical Context

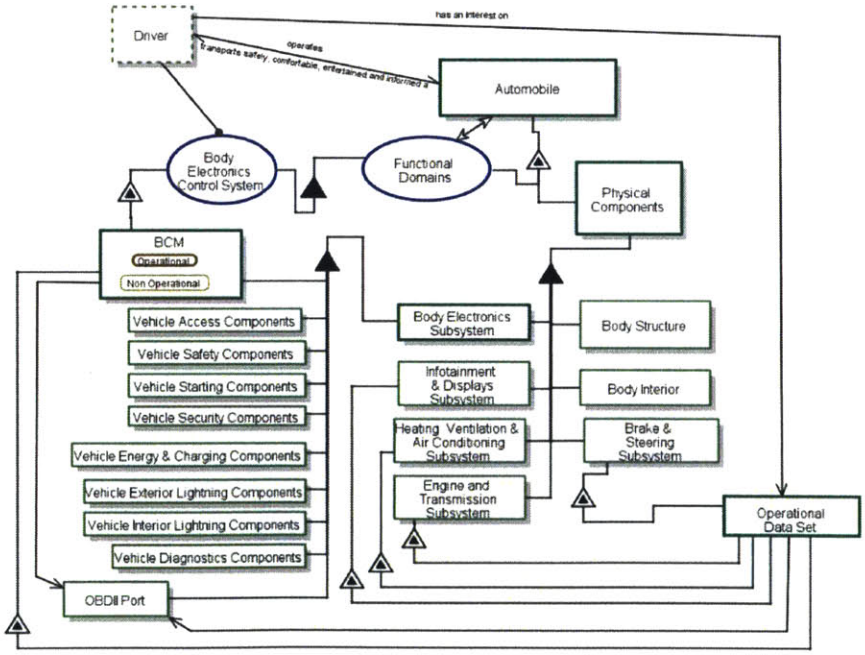


Figure 34 The Body electronics physical domain.

Driver is environmental and physical.
Driver has an interest on Operational Data Set.
Driver operates Automobile.
Driver handles Body Electronics Control System.
Automobile is physical.
Automobile exhibits Physical Components, as well as Functional Domains.
Physical Components is physical.
Physical Components consists of Body Structure, Body Interior, Engine and Transmission Subsystem, Brake & Steering Subsystem, Body Electrical Subsystem, Heating Ventilation & Air Conditioning Subsystem, and Infotainment & Displays Subsystem.
Body Structure is physical.
Body Interior is physical.
Engine and Transmission Subsystem is physical.
Engine and Transmission Subsystem exhibits Operational Data Set.
Operational Data Set relates to OBDII Port.
Brake & Steering Subsystem is physical.
Brake & Steering Subsystem exhibits Operational Data Set.
Body Electrical Subsystem is physical.
Body Electrical Subsystem consists of BCM, OBDII Port, Vehicle Energy & Charging Components, Vehicle Exterior Lightning Components, Vehicle Interior Lightning Components, Vehicle Safety Components, Vehicle Access Components, Vehicle Starting Components, Vehicle Security Components, and Vehicle Diagnostics Components.
BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.
BCM exhibits Operational Data Set.
BCM relates to OBDII Port.
OBDII Port is physical.
Vehicle Energy & Charging Components is physical.
Vehicle Exterior Lightning Components is physical.
Vehicle Interior Lightning Components is physical.
Vehicle Safety Components is physical.
Vehicle Access Components is physical.
Vehicle Starting Components is physical.
Vehicle Security Components is physical.
Vehicle Diagnostics Components is physical.
Heating Ventilation & Air Conditioning Subsystem is physical.
Heating Ventilation & Air Conditioning Subsystem exhibits Operational Data Set.
Infotainment & Displays Subsystem is physical.
Infotainment & Displays Subsystem exhibits Operational Data Set.
Functional Domains consists of Body Electronics Control System.
Body Electronics Control System exhibits BCM.
Functional Domains affects Automobile.
Automobile transports safely, comfortable, entertained and informed a Driver.

Figure 34 is a simplified view of the automobile's body electronics subsystem, which includes the physical components grouped by the functional domain to which they belong. These components interact with the Body Control Module (BCM) at the hardware level.

4.3.2.2.2 Body Electronics Control System Requirements - Interfacing

The interfacing of the body electronics control system with the vehicle components at the physical level is illustrated in Figure 35. There are eight functional domain groups that interface with the cluster of components that are part of the domain.

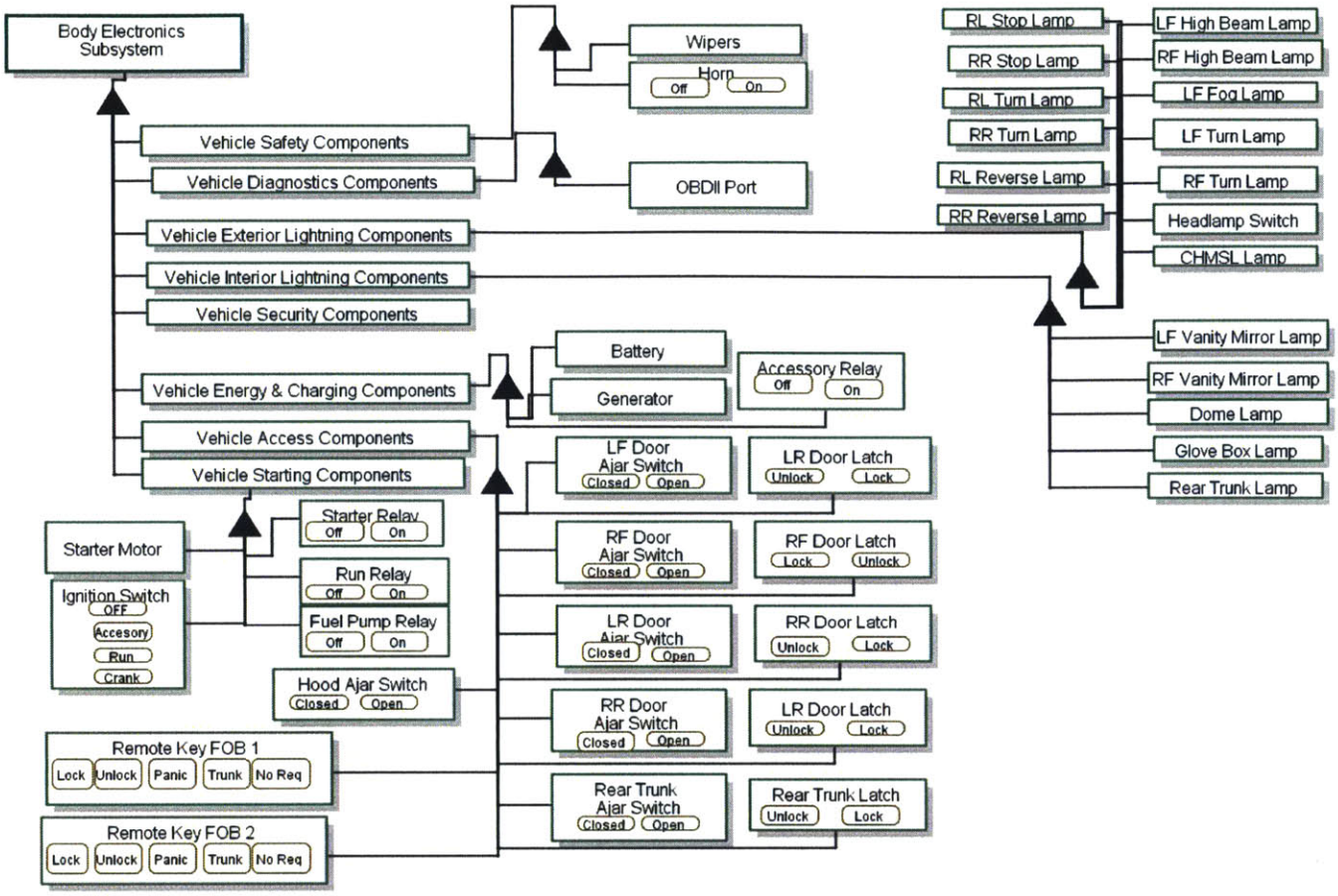


Figure 35 Automobile's Body Electronics domain interfacing.

Body Electronics Subsystem is physical.
Body Electronics Subsystem consists of Vehicle Energy & Charging Components, Vehicle Exterior Lightning Components, Vehicle Interior Lightning Components, Vehicle Safety Components, Vehicle Access Components, Vehicle Starting Components, Vehicle Security Components, and Vehicle Diagnostics Components.

Vehicle Energy & Charging Components is physical.
Vehicle Energy & Charging Components consists of Generator, Battery, and Accessory Relay.

Generator is physical.
Battery is physical.
Accessory Relay can be On or Off.

Vehicle Exterior Lightning Components is physical.
Vehicle Exterior Lightning Components consists of Headlamp Switch, LF High Beam Lamp, RF High Beam Lamp, LF Fog Lamp, LF Turn Lamp, RF Turn Lamp, CHMSL Lamp, RL Stop Lamp, RR Stop Lamp, RL Turn Lamp, RR Turn Lamp, RL Reverse Lamp, and RR Reverse Lamp.

Headlamp Switch is physical.
LF High Beam Lamp is physical.
RF High Beam Lamp is physical.
LF Fog Lamp is physical.
LF Turn Lamp is physical.
RF Turn Lamp is physical.
CHMSL Lamp is physical.
RL Stop Lamp is physical.
RR Stop Lamp is physical.
RL Turn Lamp is physical.
RR Turn Lamp is physical.
RL Reverse Lamp is physical.
RR Reverse Lamp is physical.

Vehicle Interior Lightning Components is physical.
Vehicle Interior Lightning Components consists of LF Vanity Mirror Lamp, RF Vanity Mirror Lamp, Dome Lamp, Glove Box Lamp, and Rear Trunk Lamp.

LF Vanity Mirror Lamp is physical.
RF Vanity Mirror Lamp is physical.

Dome Lamp is physical.
Glove Box Lamp is physical.
Rear Trunk Lamp is physical.
Vehicle Safety Components is physical.
Vehicle Safety Components consists of Wipers and Horn.
Wipers is physical.
Horn is physical.
Horn can be Off or On.
Vehicle Access Components is physical.
Vehicle Access Components consists of LF Door Ajar Switch, RF Door Ajar Switch, LR Door Ajar Switch, RR Door Ajar Switch, Rear Trunk Ajar Switch, Hood Ajar Switch, Remote Key FOB 1, Remote Key FOB 2, RF Door Latch, LR Door Latch, RR Door Latch, LR Door Latch, and Rear Trunk Latch.
LF Door Ajar Switch is physical.
LF Door Ajar Switch can be Open or Closed.
RF Door Ajar Switch is physical.
RF Door Ajar Switch can be Open or Closed.
LR Door Ajar Switch is physical.
LR Door Ajar Switch can be Open or Closed.
RR Door Ajar Switch is physical.
RR Door Ajar Switch can be Open or Closed.
Rear Trunk Ajar Switch is physical.
Rear Trunk Ajar Switch can be Open or Closed.
Hood Ajar Switch is physical.
Hood Ajar Switch can be Closed or Open.
Remote Key FOB 1 is physical.
Remote Key FOB 1 can be Panic, No Req, Lock, Unlock, or Trunk.
Remote Key FOB 2 is physical.
Remote Key FOB 2 can be Lock, Unlock, Panic, Trunk, or No Req.
RF Door Latch is physical.
RF Door Latch can be Lock or Unlock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RR Door Latch is physical.
RR Door Latch can be Unlock or Lock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
Rear Trunk Latch is physical.
Rear Trunk Latch can be Unlock or Lock.
Vehicle Starting Components is physical.
Vehicle Starting Components consists of Starter Motor, Ignition Switch, Starter Relay, Run Relay, and Fuel Pump Relay.
Starter Motor is physical.
Ignition Switch is physical.
Ignition Switch can be OFF, Accessory, Run, or Crank.
Starter Relay can be On or Off.
Run Relay can be On or Off.
Fuel Pump Relay can be On or Off.
Vehicle Security Components is physical.
Vehicle Diagnostics Components is physical.
Vehicle Diagnostics Components consists of OBDII Port.
OBDII Port is physical.

4.3.2.2.3 The Body Control Module (BCM) Hardware

The body electronics control system features an electronic control module called Body Control Module (BCM); the BCM architecture is illustrated in Figure 36. The electronic control module at the physical level contains Circuit Board and Enclosure and Mounting Bracket. The Circuit Board features 6 major functional groups: 1) Power Control Unit, 2) Main Processing Unit, 3) Serial Communication Unit, 4) Outputs Control Unit, 5) Inputs Monitoring unit and 6) Electrical Connectors.

The Main Processing Unit features four elements: Operational Software, Application Software, Configuration and Operational Data Set, which allow the flexibility to update the BCM functionality.

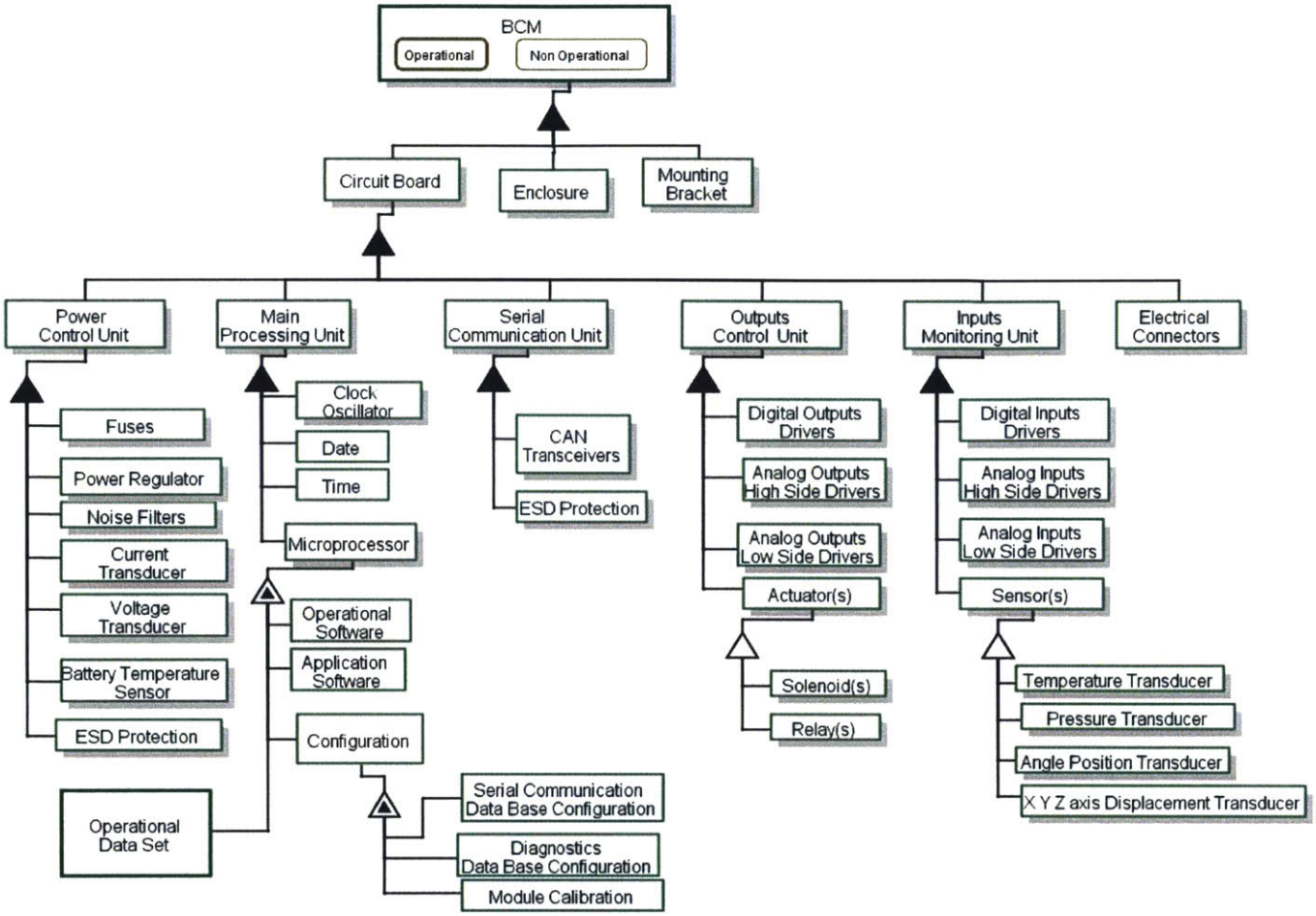


Figure 36 The automobile's Body Control Module

BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.
BCM consists of Circuit Board, Enclosure, and Mounting Bracket.
Circuit Board is physical.
Circuit Board consists of Power Control Unit, Serial Communication Unit, Inputs Monitoring Unit, Outputs Control Unit, Main Processing Unit, and Electrical Connectors.
Power Control Unit is physical.
Power Control Unit consists of Power Regulator, Fuses, Noise Filters, Current Transducer, Voltage Transducer, ESD Protection, and Battery Temperature Sensor.
Power Regulator is physical.
Fuses is physical.
Noise Filters is physical.
Current Transducer is physical.
Voltage Transducer is physical.
ESD Protection is physical.
Battery Temperature Sensor is physical.
Serial Communication Unit is physical.
Serial Communication Unit consists of CAN Transceivers and ESD Protection.
CAN Transceivers is physical.
ESD Protection is physical.
Inputs Monitoring Unit is physical.
Inputs Monitoring Unit consists of Digital Inputs Drivers, Analog Inputs High Side Drivers, Analog Inputs Low Side Drivers, and Sensor(s).
Digital Inputs Drivers is physical.
Analog Inputs High Side Drivers is physical.
Analog Inputs Low Side Drivers is physical.
Sensor(s) is physical.

Outputs Control Unit is physical.
Outputs Control Unit consists of Digital Outputs Drivers, Analog Outputs High Side Drivers, Analog Outputs Low Side Drivers, and Actuator(s).
Digital Outputs Drivers is physical.
Analog Outputs High Side Drivers is physical.
Analog Outputs Low Side Drivers is physical.
Actuator(s) is physical.
Main Processing Unit is physical.
Main Processing Unit consists of Microprocessor, Clock Oscillator, Date, and Time.
Microprocessor is physical.
Microprocessor exhibits Operational Software, Application Software, Configuration, and Operational Data Set.
Configuration exhibits Serial Communication Data Base Configuration, Diagnostics Data Base Configuration, and Module Calibration.
Clock Oscillator is physical.
Electrical Connectors is physical.
Enclosure is physical.
Mounting Bracket is physical.
Solenoid(s) is physical.
Solenoid(s) is an Actuator(s).
Relay(s) is physical.
Relay(s) is an Actuator(s).
Temperature Transducer is physical.
Temperature Transducer is a Sensor(s).
Pressure Transducer is physical.
Pressure Transducer is a Sensor(s).
Angle Position Transducer is physical.
Angle Position Transducer is a Sensor(s).
X Y Z axis Displacement Transducer is physical.
X Y Z axis Displacement Transducer is a Sensor(s).

4.3.2.3 Functional requirements

The functional domains of an automobile are illustrated in Figure 37. Each of the vehicle domains features an electronic control unit (ECU), which is part of the "Form" on the physical domain and serves as a linkage to connect to the "Function" of that given domain with the overall system. The purpose of this illustration is to help identify where the Body Electronics Control System fits in the functional domain structure.

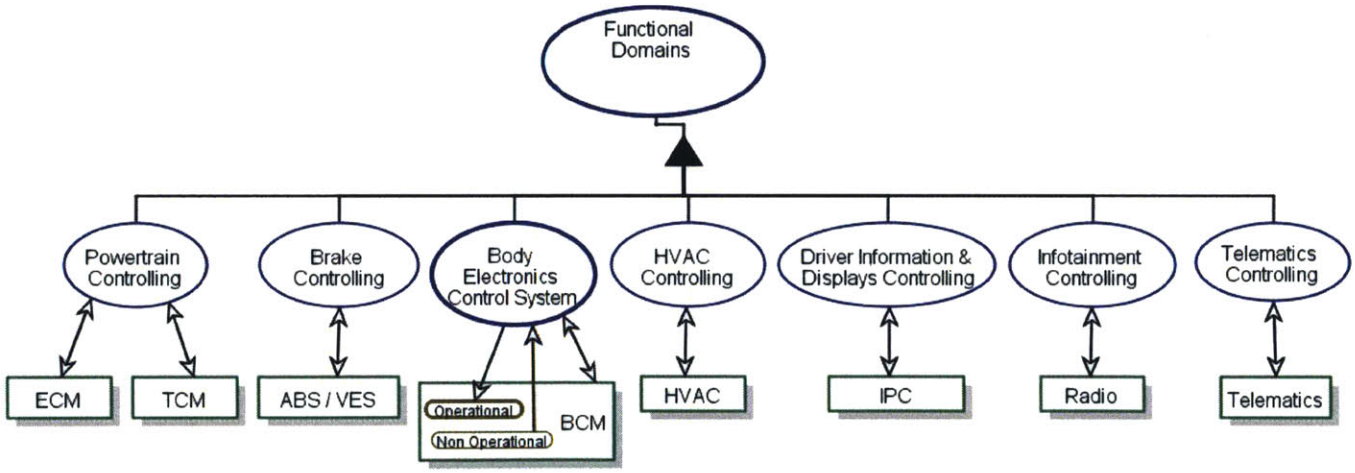


Figure 37 Automobile Functional Domains.

ECM is physical.
TCM is physical.
ABS / VES is physical.
HVAC is physical.
IPC is physical.
Radio is physical.

Telematics is physical.
BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.
Functional Domains consists of Body Electronics Control System, Powertrain Controlling, Brake Controlling, Driver Information & Displays Controlling, Infotainment Controlling, Telematics Controlling, and HVAC Controlling.
Body Electronics Control System affects BCM.
Body Electronics Control System changes BCM from Non Operational to Operational.
Powertrain Controlling affects TCM and ECM.
Brake Controlling affects ABS / VES.
Driver Information & Displays Controlling affects IPC.
Infotainment Controlling affects Radio.
Telematics Controlling affects Telematics.
HVAC Controlling affects HVAC.

4.3.2.3.1 Body Electronics Control System

The body Electronics Control System manages the following functional sections: Vehicle Power Mode, Energy & Charging, Access, Starting, Safety, Security, Lightning, and Network Management. Figure 38 illustrates the system arrangement at a high level and the interactions with the Body Control Module (BCM) component units.

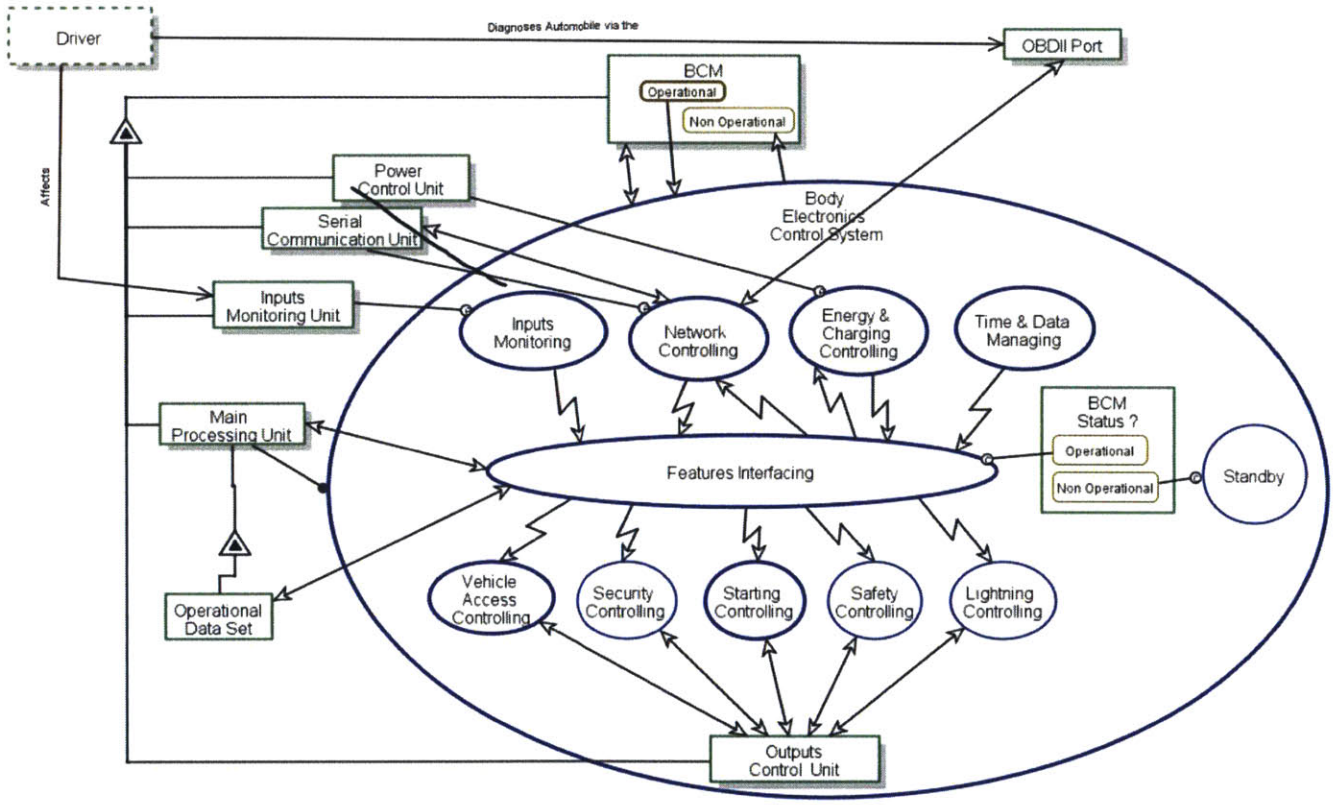


Figure 38 Body Electronics Control System

Driver is environmental and physical.
Driver Affects Inputs Monitoring Unit.
Driver Diagnoses Automobile via the OBDII Port.
BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.

BCM exhibits Power Control Unit, Serial Communication Unit, Inputs Monitoring Unit, Outputs Control Unit, and Main Processing Unit.
Power Control Unit is physical.
Power Control Unit triggers Energy & Charging Controlling.
Serial Communication Unit is physical.
Serial Communication Unit triggers Network Controlling.
Inputs Monitoring Unit is physical.
Inputs Monitoring Unit triggers Inputs Monitoring.
Outputs Control Unit is physical.
Main Processing Unit is physical.
Main Processing Unit exhibits Operational Data Set.
Main Processing Unit handles Body Electronics Control System.
OBDII Port is physical.
Body Electronics Control System affects BCM.
Body Electronics Control System changes BCM from Operational to Non Operational.
Body Electronics Control System zooms into Time & Data Managing, Energy & Charging Controlling, Inputs Monitoring, Network Controlling, Standby, Features Interfacing, Lightning Controlling, Safety Controlling, Starting Controlling, Security Controlling, and Vehicle Access Controlling, as well as BCM Status ?.
BCM Status ? can be Non Operational or Operational.
Time & Data Managing invokes Features Interfacing.
Energy & Charging Controlling requires Power Control Unit.
Energy & Charging Controlling invokes Features Interfacing.
Inputs Monitoring requires Inputs Monitoring Unit.
Inputs Monitoring invokes Features Interfacing.
Network Controlling requires Serial Communication Unit.
Network Controlling affects Serial Communication Unit and OBDII Port.
Network Controlling invokes Features Interfacing.
Standby occurs if BCM Status ? is Non Operational.
Features Interfacing occurs if BCM Status ? is Operational.
Features Interfacing affects Operational Data Set and Main Processing Unit.
Features Interfacing invokes Energy & Charging Controlling, Network Controlling, Lightning Controlling, Safety Controlling, Starting Controlling, Security Controlling, and Vehicle Access Controlling.
Lightning Controlling affects Outputs Control Unit.
Safety Controlling affects Outputs Control Unit.
Starting Controlling affects Outputs Control Unit.
Security Controlling affects Outputs Control Unit.
Vehicle Access Controlling affects Outputs Control Unit.

4.3.2.3.1.1 Network Controlling in-zoomed

The system core function relies on the Network Controlling Unit, which also provides Vehicle Power Mode control. This functional section monitors and synchronizes the vehicle network communications with the ignition key position and the vehicle's operational state. The Power Mode is a translation of the ignition switch position, which is controlled by the driver, to electrical signals that are considered state encoded values or "modes" that a driver can select by turning the ignition switch. The high level view of the Network controlling is illustrated in Figure 39.

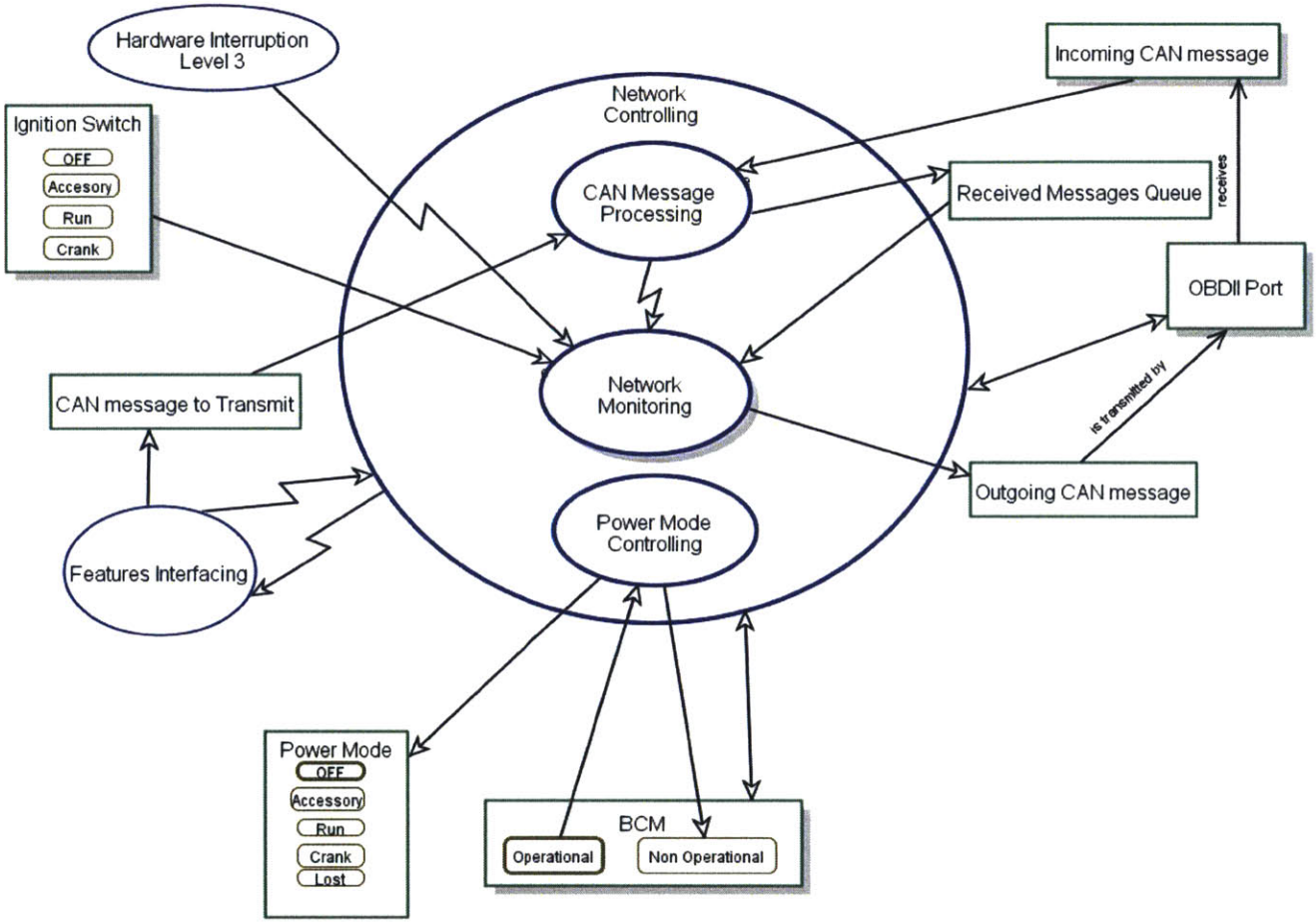


Figure 39 Network Controlling in-zoomed

BCM is physical.
BCM can be Operational or Non Operational.
    Operational is initial.
OBDII Port is physical.
OBDII Port receives Incoming CAN message.
Power Mode can be OFF, Accessory, Run, Crank, or Lost.
    OFF is initial.
Ignition Switch is physical.
Ignition Switch can be OFF, Accessory, Run, or Crank.
Ignition Switch triggers Network Monitoring when its state changes.
Incoming CAN message is physical.
Incoming CAN message triggers CAN Message Processing.
Outgoing CAN message is transmitted by OBDII Port.
Features Interfacing yields CAN message to Transmit.
Features Interfacing invokes Network Controlling.
Hardware Interruption Level 3 invokes Network Monitoring.
Network Controlling affects BCM and OBDII Port.
Network Controlling invokes Features Interfacing.
Network Controlling zooms into CAN Message Processing, Network Monitoring, and Power Mode Controlling.
    CAN Message Processing consumes CAN message to Transmit and Incoming CAN message.
    CAN Message Processing yields Received Messages Queue.
    CAN Message Processing invokes Network Monitoring.
    Network Monitoring is physical.
    Network Monitoring consumes Received Messages Queue and Ignition Switch.
    Network Monitoring yields Outgoing CAN message.
    Power Mode Controlling changes BCM from Operational to Non Operational.
    Power Mode Controlling yields Power Mode.

4.3.2.3.1.1.1 CAN Message Processing in-zoomed

This unit is responsible to communicate the functional information to the modules participating in the vehicle network by CAN functional messages, following the CAN protocols and diagnostics messages. The function shall monitor and determine the status of incoming messages from the vehicle via the serial data network and process any serial data updates to be communicated to the rest of the vehicle. Figure 40 illustrates the building blocks of this function.

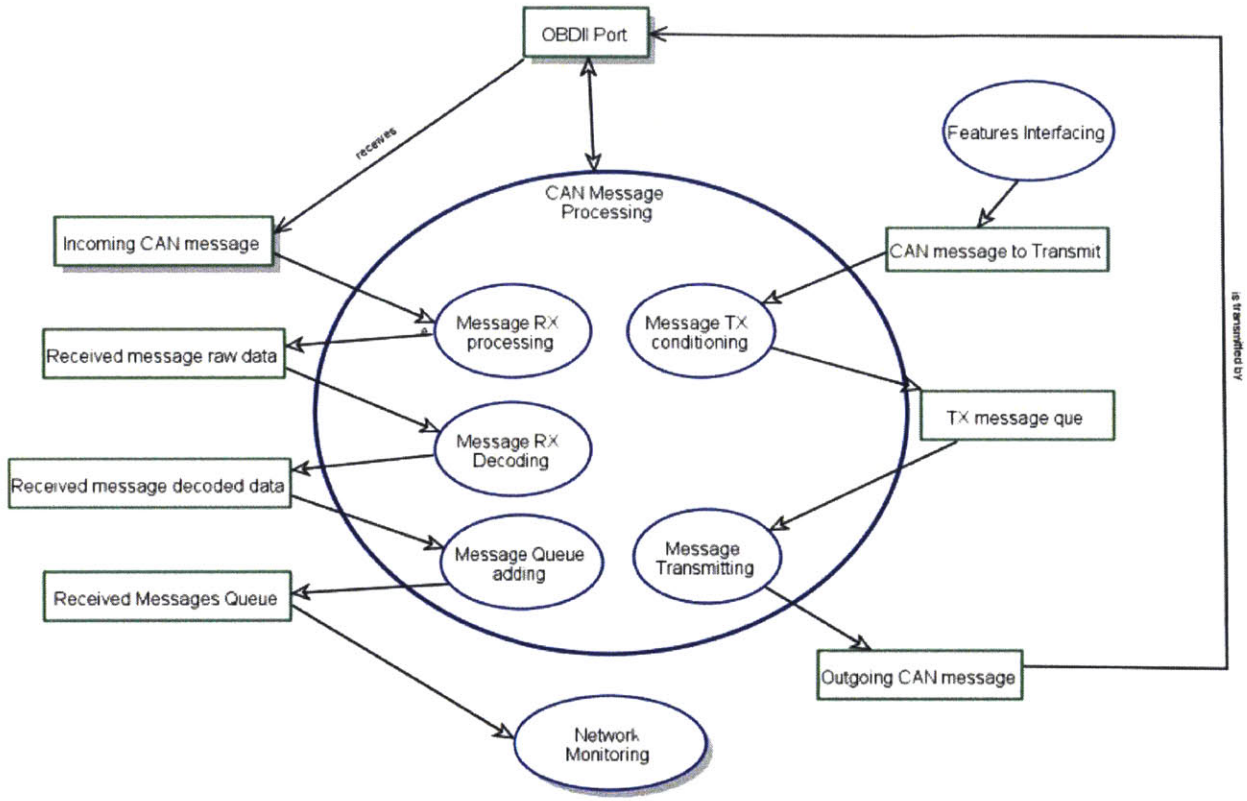


Figure 40 CAN Message Processing in-zoomed

OBDII Port is physical.
OBDII Port receives Incoming CAN message.
Incoming CAN message is physical.
Incoming CAN message triggers Message RX processing.
Outgoing CAN message is transmitted by OBDII Port.
Features Interfacing yields CAN message to Transmit.
Network Monitoring is physical.
Network Monitoring consumes Received Messages Queue.
CAN Message Processing affects OBDII Port.
CAN Message Processing zooms into Message RX processing, Message TX conditioning, Message RX Decoding, Message Transmitting, and Message Queue adding.

- Message RX processing consumes Incoming CAN message.
- Message RX processing yields Received message raw data.
- Message TX conditioning consumes CAN message to Transmit.
- Message TX conditioning yields TX message queue.
- Message RX Decoding consumes Received message raw data.
- Message RX Decoding yields Received message decoded data.
- Message Transmitting consumes TX message queue.
- Message Transmitting yields Outgoing CAN message.
- Message Queue adding consumes Received message decoded data.
- Message Queue adding yields Received Messages Queue.

4.3.2.3.1.1.2 Network Monitoring in-zoomed

This unit manages the Vehicle Network operational states that can be: a) Off Sleep, b) Off-Awake, and c) Awake. Since the BCM is the network master module, the network operational states are managed by the BCM in this section. The BCM monitors two types of inputs that enable the vehicle network: Hardware inputs and Network wake up messages (CAN messages). Figure 41 illustrates this monitoring process.

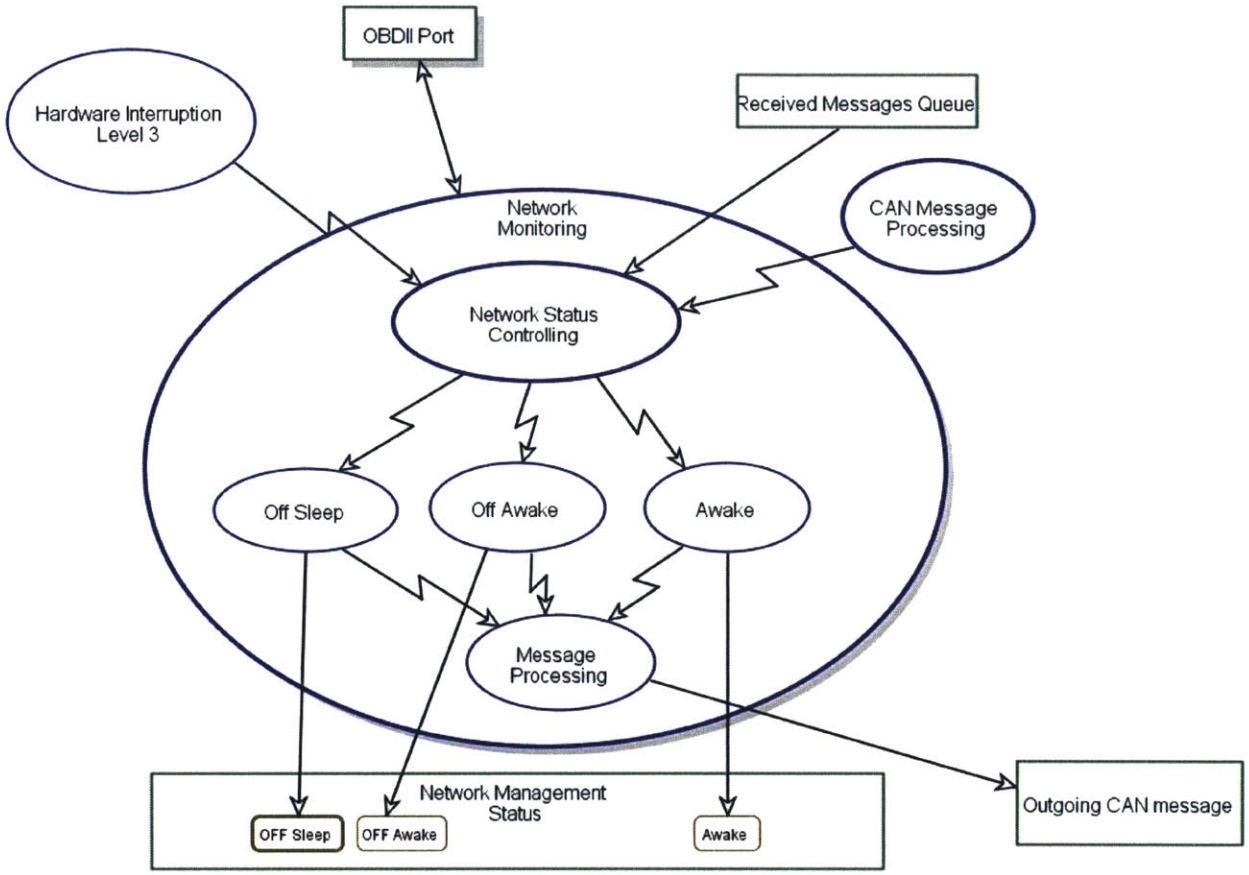


Figure 41 Network Monitoring in-zoomed

OBDII Port is physical.
Network Management Status can be Awake, OFF Awake, or OFF Sleep.
OFF Sleep is initial.
Hardware Interruption Level 3 invokes Network Status Controlling.
CAN Message Processing invokes Network Status Controlling.
Network Monitoring is physical.
Network Monitoring affects OBDII Port.
Network Monitoring zooms into Network Status Controlling, Awake, Off Awake, Off Sleep, and Message Processing.
Network Status Controlling consumes Received Messages Queue.
Network Status Controlling invokes Off Sleep, Off Awake, and Awake.
Awake yields Awake Network Management Status.
Awake invokes Message Processing.
Off Awake yields OFF Awake Network Management Status.
Off Awake invokes Message Processing.
Off Sleep yields OFF Sleep Network Management Status.
Off Sleep invokes Message Processing.
Message Processing yields Outgoing CAN message.

4.3.2.3.1.1.3 Network Status Controlling in-zoomed

As shown in Figure 42, this section controls the network status. By monitoring the wake up inputs (hardware or CAN messages), the power mode the vehicle is in, and the current network status, it determines the required transitions between the Off Sleep, Off-Awake and Awake states and yields the new network status.

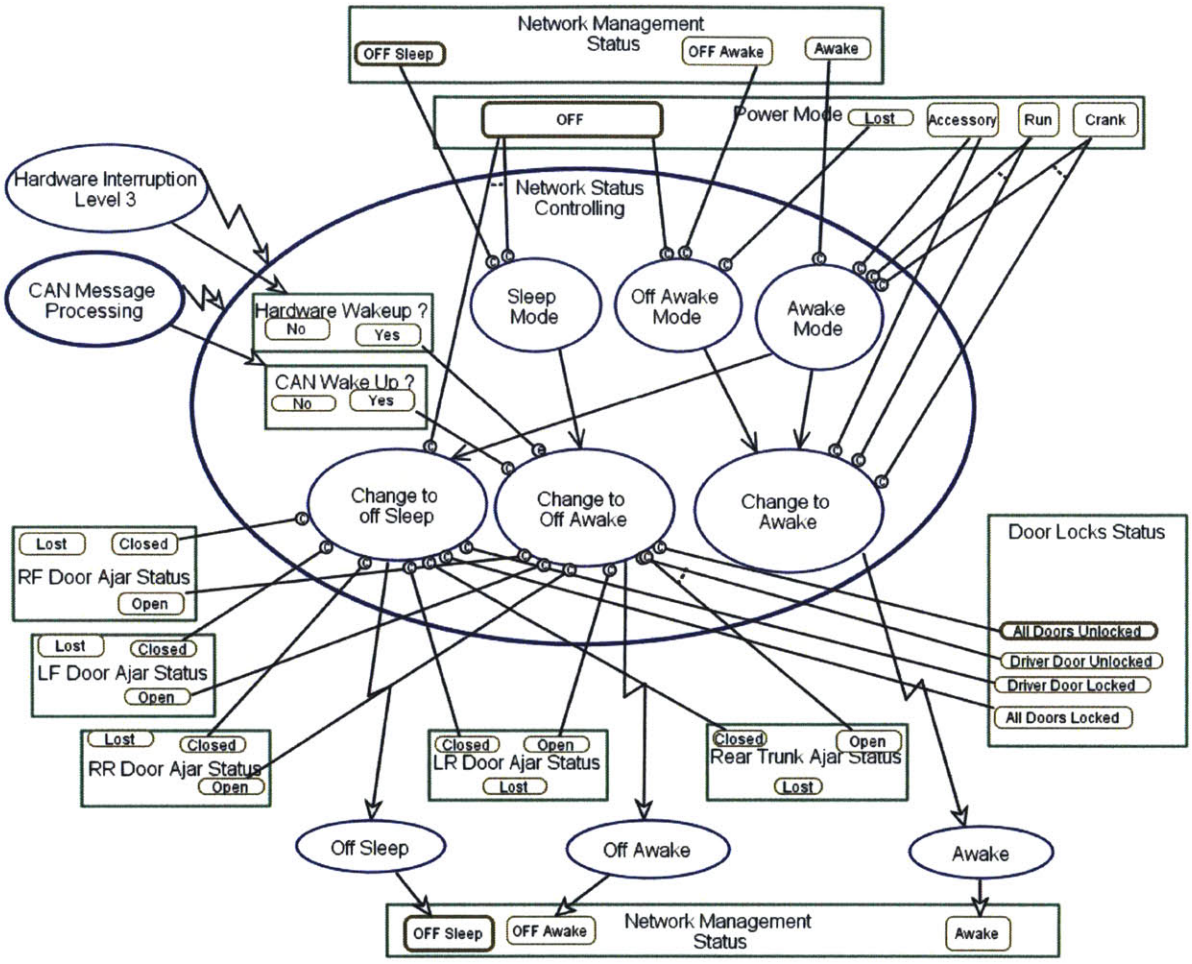


Figure 42 Network Status Controlling in-zoomed

Power Mode can be OFF, Accessory, Run, Crank, or Lost.
OFF is initial.
Door Locks Status can be All Doors Locked, Driver Door Locked, All Doors Unlocked, or Driver Door Unlocked.
All Doors Unlocked is initial.
Network Management Status can be Awake, OFF Awake, or OFF Sleep.
OFF Sleep is initial.
LF Door Ajar Status can be Open, Closed, or Lost.
LR Door Ajar Status can be Open, Closed, or Lost.
RF Door Ajar Status can be Open, Closed, or Lost.
RR Door Ajar Status can be Closed, Open, or Lost.
Rear Trunk Ajar Status can be Closed, Open, or Lost.
Hardware Interruption Level 3 yields Hardware Wakeup ?.
Hardware Interruption Level 3 invokes Network Status Controlling.
CAN Message Processing yields CAN Wake Up ?.
CAN Message Processing invokes Network Status Controlling.
Off Awake yields OFF Awake Network Management Status.
Off Sleep yields OFF Sleep Network Management Status.
Awake yields Awake Network Management Status.
Network Status Controlling zooms into Off Awake Mode, Sleep Mode, Awake Mode, Change to off Sleep, Change to Awake, and

Change to Off Awake, as well as CAN Wake Up ? and Hardware Wakeup ?.

CAN Wake Up ? can be Yes or No.

Hardware Wakeup ? can be Yes or No.

Hardware Wakeup ? triggers Change to Off Awake when it enters Yes.

Off Awake Mode Change to Awake.

Off Awake Mode occurs if Power Mode is OFF, Network Management Status is OFF Awake, and Power Mode is Lost.

Sleep Mode Change to Off Awake.

Sleep Mode occurs if Network Management Status is OFF Sleep and Power Mode is OFF.

Awake Mode Change to Awake.

Awake Mode Change to off Sleep.

Awake Mode occurs if Power Mode is Run, Power Mode is Accessory, Power Mode is Crank, and Network Management Status is Awake.

Change to off Sleep occurs if Power Mode is OFF, Door Locks Status is All Doors Locked, Door Locks Status is Driver Door Locked, RF Door Ajar Status is Closed, LF Door Ajar Status is Closed, RR Door Ajar Status is Closed, LR Door Ajar Status is Closed, and Rear Trunk Ajar Status is Closed.

Change to off Sleep invokes Off Sleep.

Change to Awake occurs if Power Mode is Accessory, Power Mode is Run, and Power Mode is Crank.

Change to Awake invokes Awake.

Change to Off Awake occurs if RF Door Ajar Status is Open, LF Door Ajar Status is Open, RR Door Ajar Status is Open, LR Door Ajar Status is Open, Door Locks Status is All Doors Unlocked, and CAN Wake Up ? is Yes.

Change to Off Awake occurs if either Rear Trunk Ajar Status is Open or Door Locks Status is Driver Door Unlocked.

Change to Off Awake requires Yes Hardware Wakeup ?.

Change to Off Awake invokes Off Awake.

4.3.2.3.1.1.4 Power Mode Controlling in-zoomed

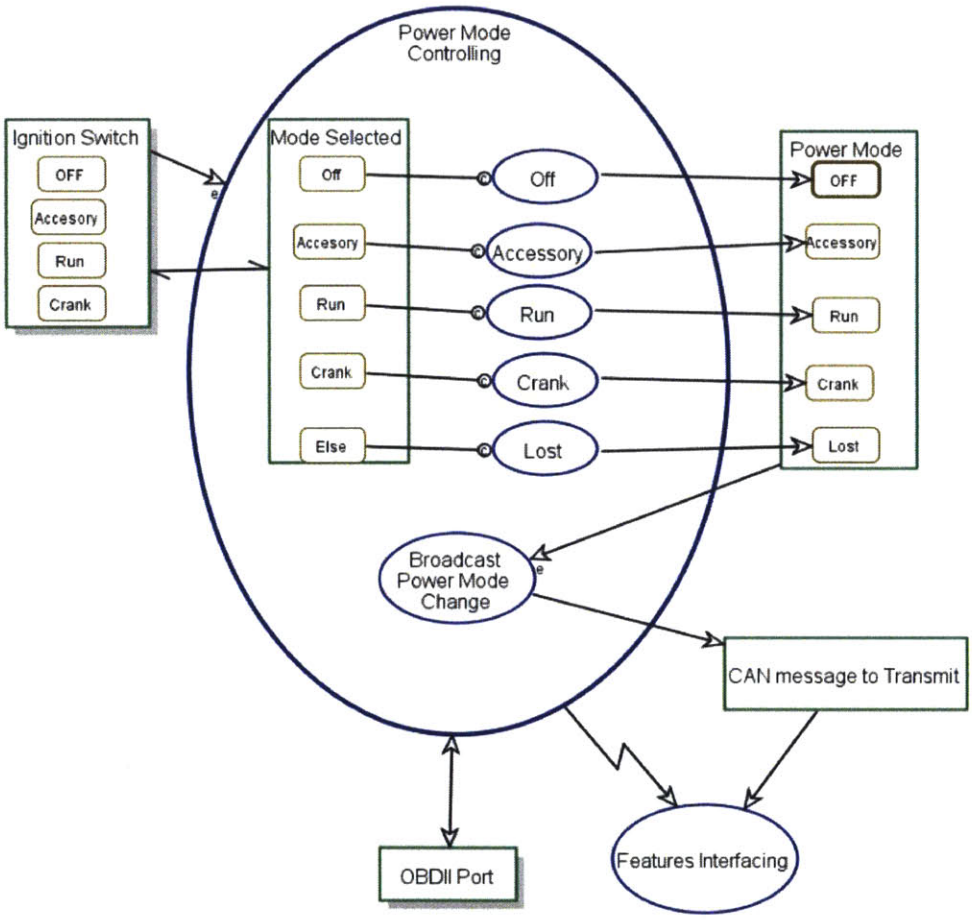


Figure 43 Power Mode Controlling in-zoomed

OBDII Port is physical.

Power Mode can be OFF, Accessory, Run, Crank, or Lost.

OFF is initial.
Power Mode triggers Broadcast Power Mode Change when its state changes.
Ignition Switch is physical.
Ignition Switch can be OFF, Accessory, Run, or Crank.
Ignition Switch triggers Power Mode Controlling when its state changes.
Features Interfacing consumes CAN message to Transmit.
Power Mode Controlling affects OBDII Port.
Power Mode Controlling consumes Ignition Switch.
Power Mode Controlling invokes Features Interfacing.
Power Mode Controlling zooms into Off, Accessory, Run, Crank, Lost, and Broadcast Power Mode Change, as well as Mode Selected.
Mode Selected can be Off, Accessory, Run, Crank, or Else.
Mode Selected and Ignition Switch are equivalent.
Off occurs if Mode Selected is Off.
Off yields OFF Power Mode.
Accessory occurs if Mode Selected is Accessory.
Accessory yields Accessory Power Mode.
Run occurs if Mode Selected is Run.
Run yields Run Power Mode.
Crank occurs if Mode Selected is Crank.
Crank yields Crank Power Mode.
Lost occurs if Mode Selected is Else.
Lost yields Lost Power Mode.
Broadcast Power Mode Change consumes Power Mode.
Broadcast Power Mode Change yields CAN message to Transmit.

The vehicle power mode determination depends on the Ignition Switch position, selected by the driver. As shown in Figure 43, a change in the ignition switch triggers this sub-function to determine the new power mode, and the Broadcast Power Mode subsequently invokes Features Interfacing by providing a new CAN message updated with the desired power mode to be communicated to the vehicle network.

4.3.2.3.1.2 Energy & Charging Controlling – in zoomed

As shown in Figure 44, the Energy & Charging control section monitors the available energy power stored in the battery. Additionally, it controls battery charging by commanding the alternator to adjust the amount of energy to be generated to keep the energy flow at the optimum level (any value from 0% to 100% of the generator capacity). This section also controls the electrical load shedding to disable loads when the energy level becomes critically low to preserve the energy and extend the battery life.

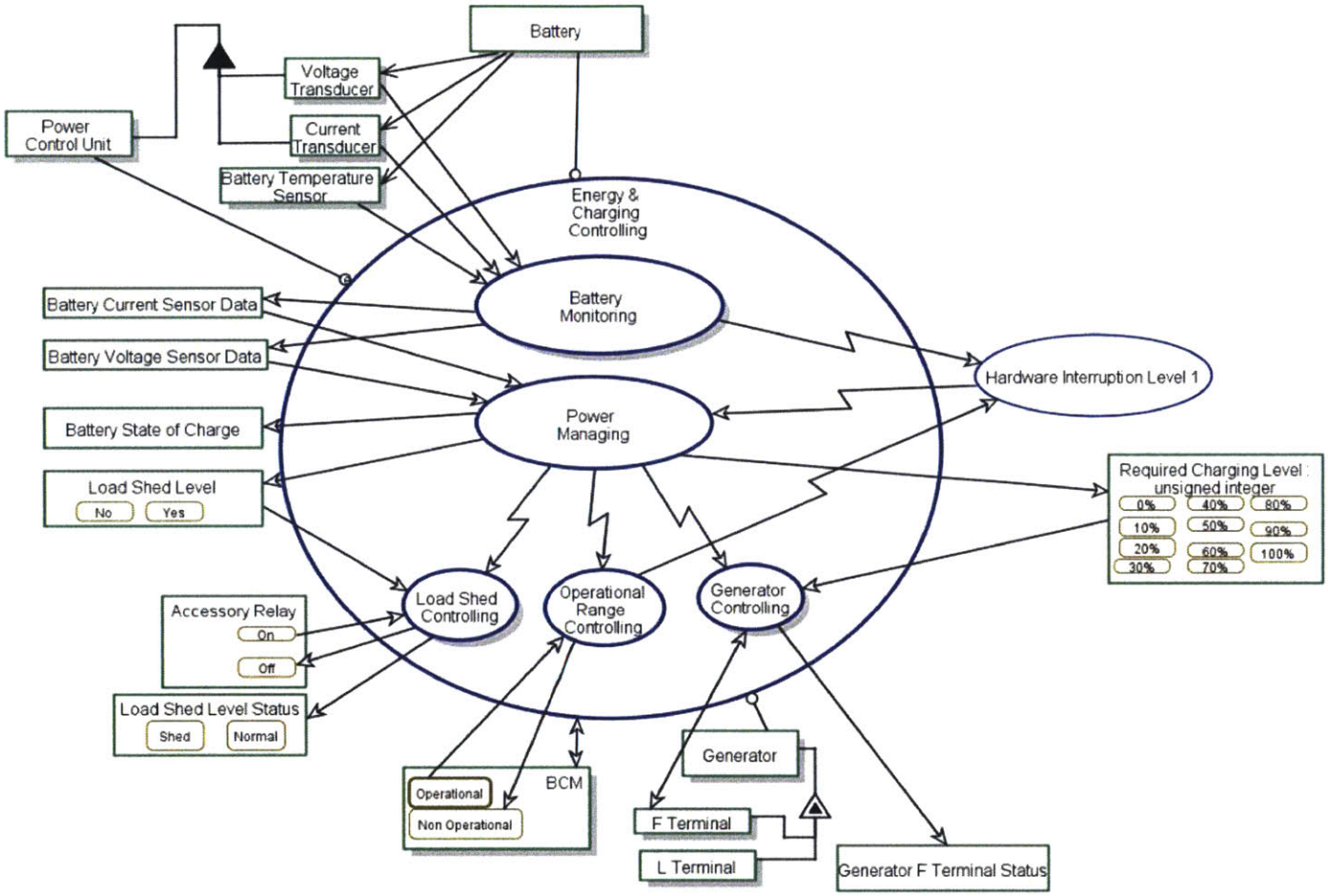


Figure 44 Energy & Charging Controlling – in zoomed

Generator is physical.
Generator exhibits L Terminal and F Terminal.
    L Terminal is physical.
    F Terminal is physical.
Battery is physical.
Battery relates to Battery Temperature Sensor.
Battery relates to Current Transducer.
Battery relates to Voltage Transducer.
Power Control Unit is physical.
Power Control Unit consists of Current Transducer and Voltage Transducer.
    Current Transducer is physical.
    Voltage Transducer is physical.
Power Control Unit triggers Energy & Charging Controlling.
BCM is physical.
BCM can be Operational or Non Operational.
    Operational is initial.
Load Shed Level Status can be Normal or Shed.
Required Charging Level is of type unsigned integer.
Load Shed Level can be Yes or No.
Accessory Relay can be On or Off.
Battery Temperature Sensor is physical.
Hardware Interruption Level 1 invokes Power Managing.
Energy & Charging Controlling requires Battery, Generator, and Power Control Unit.
Energy & Charging Controlling affects BCM.
Energy & Charging Controlling zooms into Battery Monitoring, Power Managing, Generator Controlling, Operational Range Controlling, and Load Shed Controlling.
    Battery Monitoring is physical.
    Battery Monitoring consumes Voltage Transducer, Current Transducer, and Battery Temperature Sensor.
    Battery Monitoring yields Battery Current Sensor Data and Battery Voltage Sensor Data.

Battery Monitoring invokes Hardware Interruption Level 1.
Power Managing consumes Battery Current Sensor Data and Battery Voltage Sensor Data.
Power Managing yields Required Charging Level, Load Shed Level, and Battery State of Charge.
Power Managing invokes Generator Controlling, Load Shed Controlling, and Operational Range Controlling.
Generator Controlling is physical.
Generator Controlling affects F Terminal.
Generator Controlling consumes Required Charging Level.
Generator Controlling yields Generator F Terminal Status.
Operational Range Controlling changes BCM from Operational to Non Operational.
Operational Range Controlling invokes Hardware Interruption Level 1.
Load Shed Controlling is physical.
Load Shed Controlling changes Accessory Relay from On to Off.
Load Shed Controlling consumes Load Shed Level.
Load Shed Controlling yields Load Shed Level Status.

4.3.2.3.1.2.1 Battery Monitoring in-zoomed

This section is responsible to monitor battery voltage, current and temperature. Figure 45 illustrates the serial process that calls in sequence the Battery Parameters Sampling, Data Filtering, Analog to Digital Conversion, and Data Encoding. The output from this process is a Hardware interruption level 1 that invokes the Power Managing sub-process with updated Battery Voltage Sensor Data, Battery Current Sensor Data, and Battery Temperature Sensor Data.

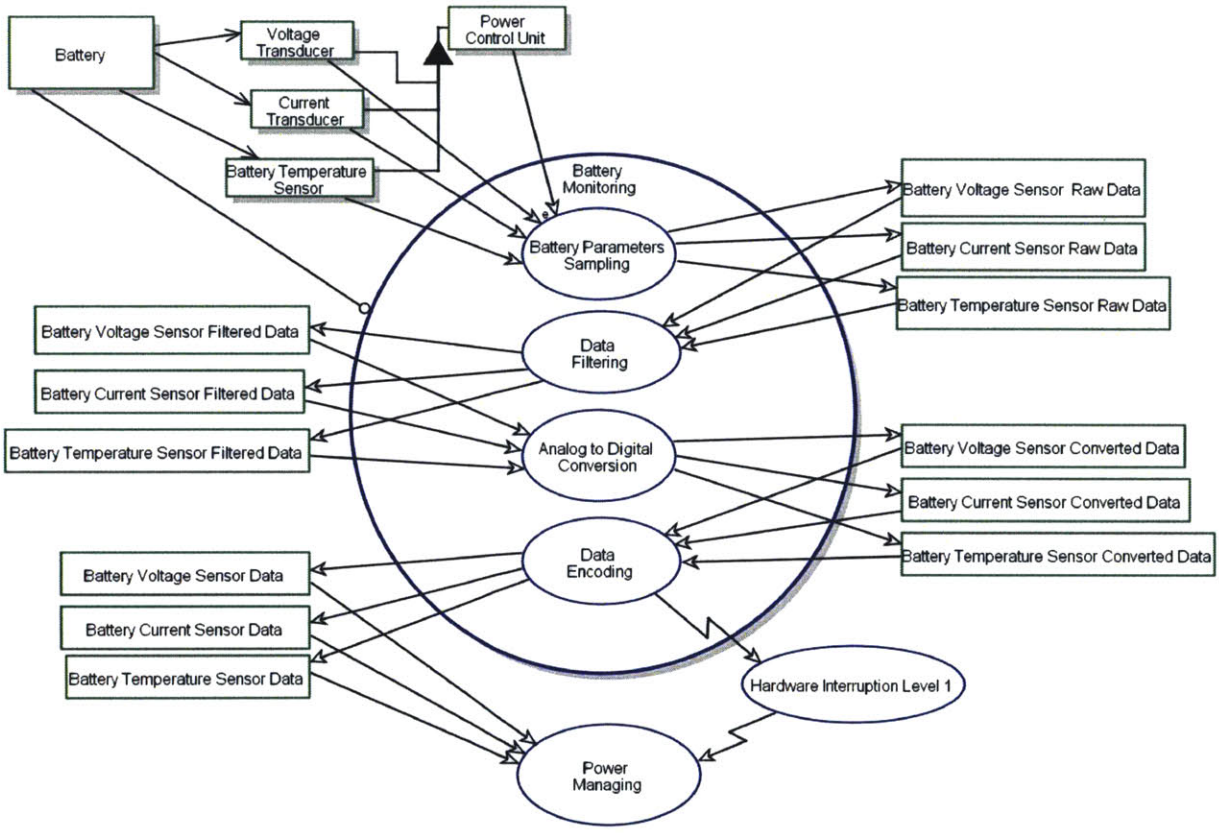


Figure 45 Battery Monitoring in-zoomed

Battery is physical.
Battery relates to Battery Temperature Sensor.
Battery relates to Current Transducer.
Battery relates to Voltage Transducer.
Power Control Unit is physical.
Power Control Unit consists of Current Transducer, Voltage Transducer, and Battery Temperature Sensor.
Current Transducer is physical.
Voltage Transducer is physical.
Battery Temperature Sensor is physical.
Power Control Unit triggers Battery Parameters Sampling.
Power Managing consumes Battery Temperature Sensor Data, Battery Current Sensor Data, and Battery Voltage Sensor Data.
Hardware Interruption Level 1 invokes Power Managing.
Battery Monitoring is physical.
Battery Monitoring requires Battery.
Battery Monitoring zooms into Battery Parameters Sampling, Data Filtering, Analog to Digital Conversion, and Data Encoding.
Battery Parameters Sampling consumes Voltage Transducer, Current Transducer, Battery Temperature Sensor, and Power Control Unit.
Battery Parameters Sampling yields Battery Current Sensor Raw Data, Battery Voltage Sensor Raw Data, and Battery Temperature Sensor Raw Data.
Data Filtering consumes Battery Current Sensor Raw Data, Battery Voltage Sensor Raw Data, and Battery Temperature Sensor Raw Data.
Data Filtering yields Battery Current Sensor Filtered Data, Battery Voltage Sensor Filtered Data, and Battery Temperature Sensor Filtered Data.
Analog to Digital Conversion consumes Battery Current Sensor Filtered Data, Battery Voltage Sensor Filtered Data, and Battery Temperature Sensor Filtered Data.
Analog to Digital Conversion yields Battery Current Sensor Converted Data, Battery Voltage Sensor Converted Data, and Battery Temperature Sensor Converted Data.
Data Encoding consumes Battery Current Sensor Converted Data, Battery Voltage Sensor Converted Data, and Battery Temperature Sensor Converted Data.
Data Encoding yields Battery Current Sensor Data, Battery Voltage Sensor Data, and Battery Temperature Sensor Data.
Data Encoding invokes Hardware Interruption Level 1.

4.3.2.3.1.2.2 Generator Controlling in-zoomed

As sown in Figure 46, Generator Controlling sub-process is also a serial process that provides a command to control the electric generator L-Terminal, by providing a PWM pulse that sets the desired charging control or duty cycle. The sub-process is invoked by the Power Managing process. It receives the Required Charging level by the Charging Level Command Receiving, thus the command is translated by Conversion to PWM that yields to the F Terminal physical output and the Generator F Terminal Status update to adjust the amount of energy to be supplied to the battery and keep the energy flow at the optimum level. The desired PWM is also communicated to the rest of the vehicle via the PWM message and the Hardware interruption level 4 which is called at the end of this process.

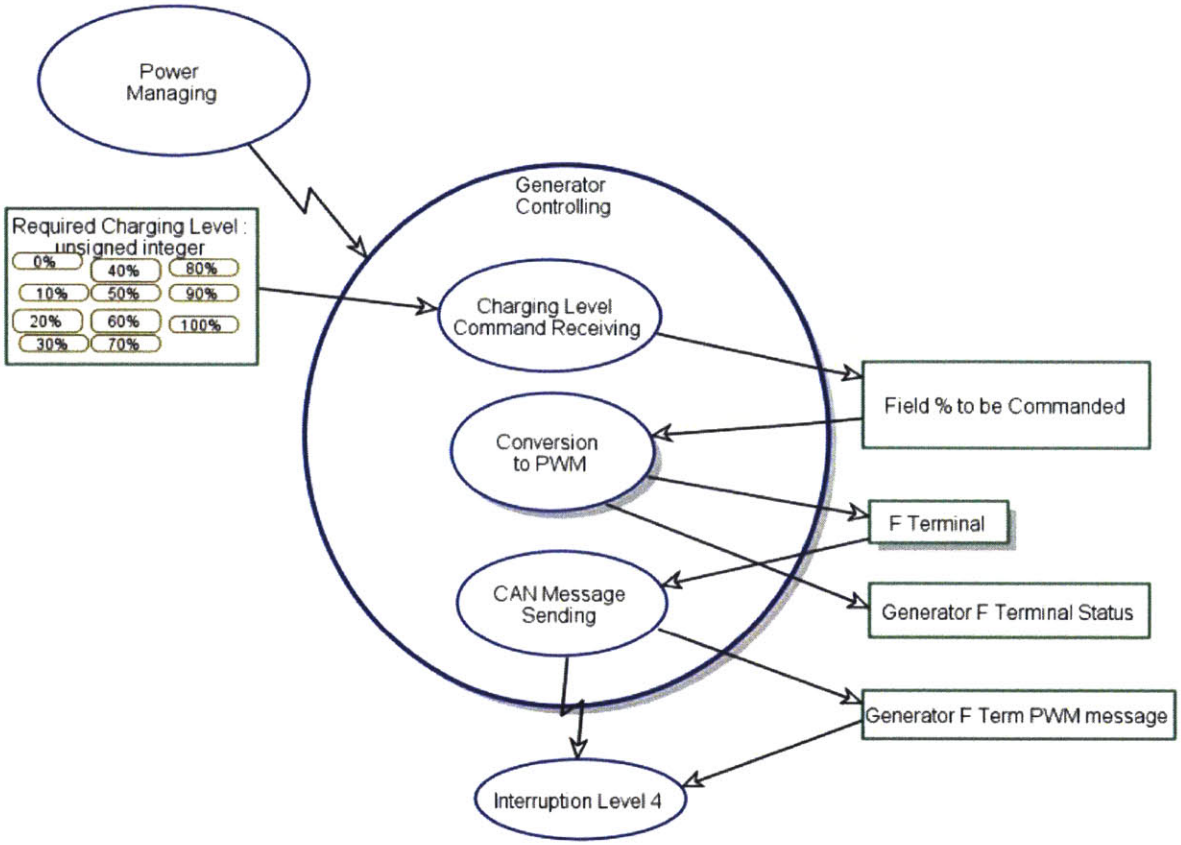


Figure 46 Generator Controlling in-zoomed

F Terminal is physical.
Required Charging Level is of type unsigned integer.
Power Managing invokes Generator Controlling.
Interruption Level 4 consumes Generator F Term PWM message.
Generator Controlling is physical.
Generator Controlling zooms into Charging Level Command Receiving, Conversion to PWM, and CAN Message Sending.
- Charging Level Command Receiving consumes Required Charging Level.
- Charging Level Command Receiving yields Field % to be Commanded.
- Conversion to PWM is physical.
- Conversion to PWM consumes Field % to be Commanded.
- Conversion to PWM yields F Terminal and Generator F Terminal Status.
- CAN Message Sending consumes F Terminal.
- CAN Message Sending yields Generator F Term PWM message.
- CAN Message Sending invokes Interruption Level 4.

4.3.2.3.1.2.3 Load Shed Controlling in-zoomed

As illustrated in Figure 47, this section monitors the status of the Load Shed flag, which is updated by the Operational Range Controlling section, and manages the accessory relay to disable the energy consumption to preserve the energy that is required to operate the vehicle. This process also transmits the CAN messages to display on the instrument cluster when Load Shed condition occurs.

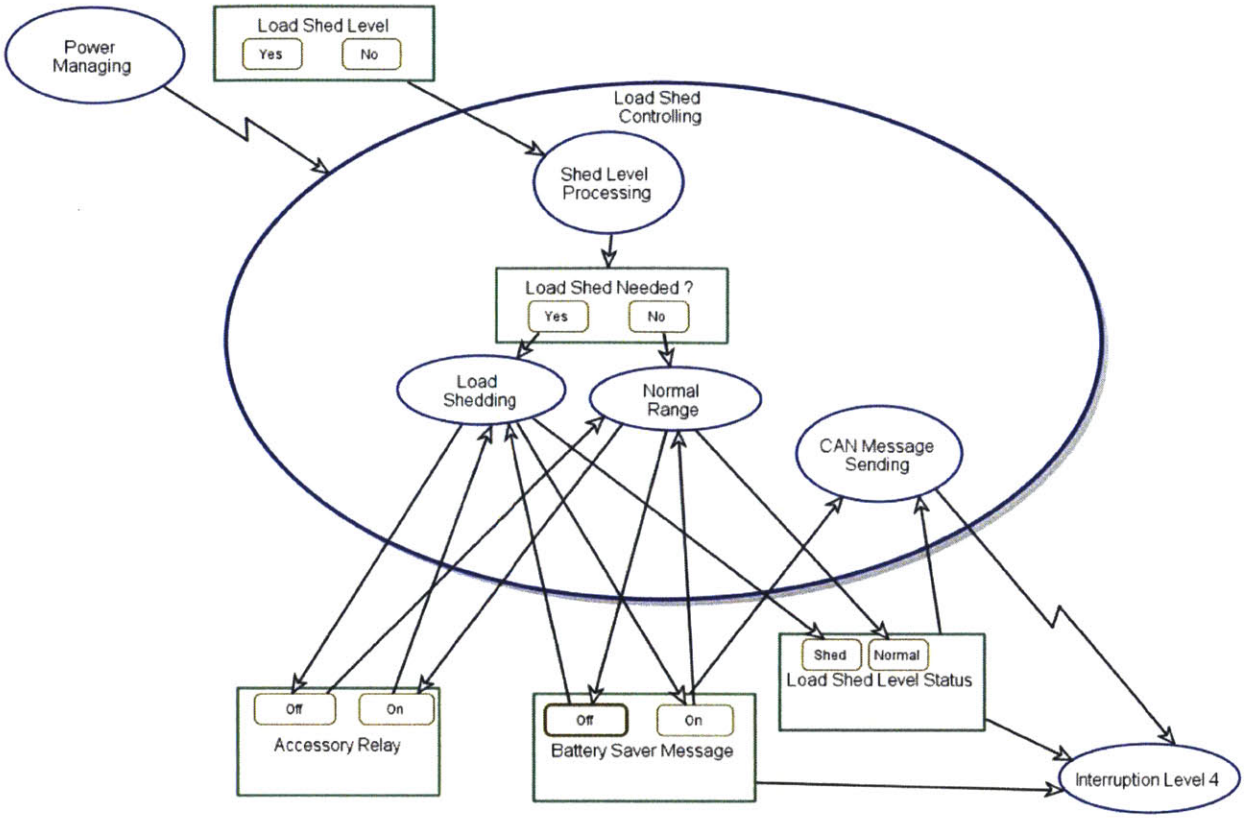


Figure 47 Load Shed Controlling in-zoomed

Load Shed Level Status can be Normal or Shed.
Load Shed Level can be Yes or No.
Accessory Relay can be On or Off.
Battery Saver Message can be On or Off.
Off is initial.
Power Managing invokes Load Shed Controlling.
Interruption Level 4 consumes Load Shed Level Status and Battery Saver Message.
Load Shed Controlling is physical.
Load Shed Controlling zooms into Shed Level Processing, Load Shedding, Normal Range, and CAN Message Sending, as well as Load Shed Needed ?.
Load Shed Needed ? can be Yes or No.
Shed Level Processing consumes Load Shed Level.
Shed Level Processing yields Load Shed Needed ?.
Load Shedding changes Accessory Relay from On to Off and Battery Saver Message from Off to On.
Load Shedding consumes Yes Load Shed Needed ?.
Load Shedding yields Shed Load Shed Level Status.
Normal Range changes Accessory Relay from Off to On and Battery Saver Message from On to Off.
Normal Range consumes No Load Shed Needed ?.
Normal Range yields Normal Load Shed Level Status.
CAN Message Sending consumes Battery Saver Message and Load Shed Level Status.
CAN Message Sending invokes Interruption Level 4.

4.3.2.3.1.2.4 Operational Range Controlling in-zoomed

As illustrated in Figure 48, this sub-process manages the BCM operational state. The BCM can detect to two possible sates: 1) Normal Operation Range, and 2) Out of Operation Range. The sub-process monitors the Battery Voltage Sensor Data and Battery State of Charge.

For the Normal Operation range, a battery voltage is above 11 volts and less than 16 volts along with a battery state of charge between 50% and 100%.

For the Out of Operation Range, There are two extreme scenarios considered by this sub-process:
Scenario 1- An automobile could potentially face battery voltage below 11 volts and depleted charge, thus the BCM will face a charging system conditions Below Operational Range that require to command the alternator to charge the battery close to the 100% of duty cycle and disable the accessory relay to preserve energy.

Scenario 2- Other condition could be a fully charged battery with voltage above 16 volts which is Above Operational Range, thus the BCM will require the alternator to not provide any energy or duty cycle of 0%, since this condition could cause a battery explosion.

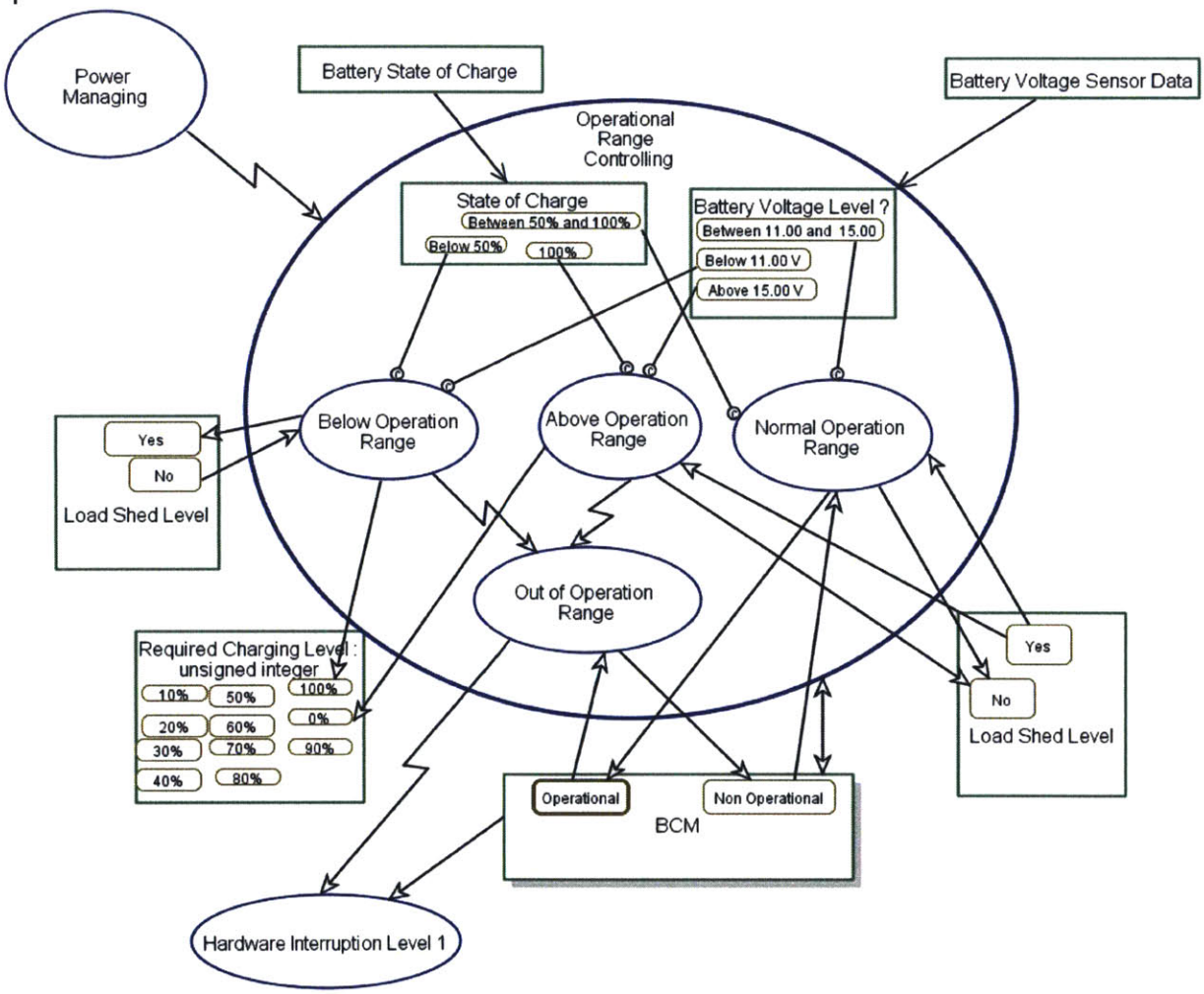


Figure 48 Operational Range Controlling in-zoomed

BCM is physical.
BCM can be Operational or Non Operational.
Operational is initial.
Battery Voltage Sensor Data relates to Battery Voltage Level ?.
Battery State of Charge relates to State of Charge.
Required Charging Level is of type unsigned integer.

Load Shed Level can be Yes or No.
Power Managing invokes Operational Range Controlling.
Hardware Interruption Level 1 consumes BCM.
Operational Range Controlling affects BCM.
Operational Range Controlling zooms into Above Operation Range, Below Operation Range, Normal Operation Range, and Out of Operation Range, as well as Battery Voltage Level ? and State of Charge.

Battery Voltage Level ? can be Below 11.00 V, Above 15.00 V, or Between 11.00 and 15.00.
State of Charge can be Below 50%, Between 50% and 100%, or 100%.
Above Operation Range occurs if Battery Voltage Level ? is Above 15.00 V and State of Charge is 100%.
Above Operation Range changes Load Shed Level from Yes to No.
Above Operation Range yields 0% Required Charging Level.
Above Operation Range invokes Out of Operation Range.
Below Operation Range occurs if State of Charge is Below 50% and Battery Voltage Level ? is Below 11.00 V.
Below Operation Range changes Load Shed Level from No to Yes.
Below Operation Range yields 100% Required Charging Level.
Below Operation Range invokes Out of Operation Range.
Normal Operation Range occurs if State of Charge is Between 50% and 100% and Battery Voltage Level ? is Between 11.00 and 15.00.
Normal Operation Range changes BCM from Non Operational to Operational and Load Shed Level from Yes to No.
Out of Operation Range changes BCM from Operational to Non Operational.
Out of Operation Range invokes Hardware Interruption Level 1.

4.3.2.3.1.2.5 Power Managing in-zoomed

As shown in Figure 49, this is a serial sub-process, and the sequence is executed as follows: State of Charge Determination, followed by Load Shed Determination, and finally Generator Charge level Determination. The sub-process yield three parameters: Battery State of Charge, Load Shed Level, and Required Charging Level, which also invokes the Operational Range Controlling, Generator Controlling and the Load Shed Controlling sub-processes during the execution.

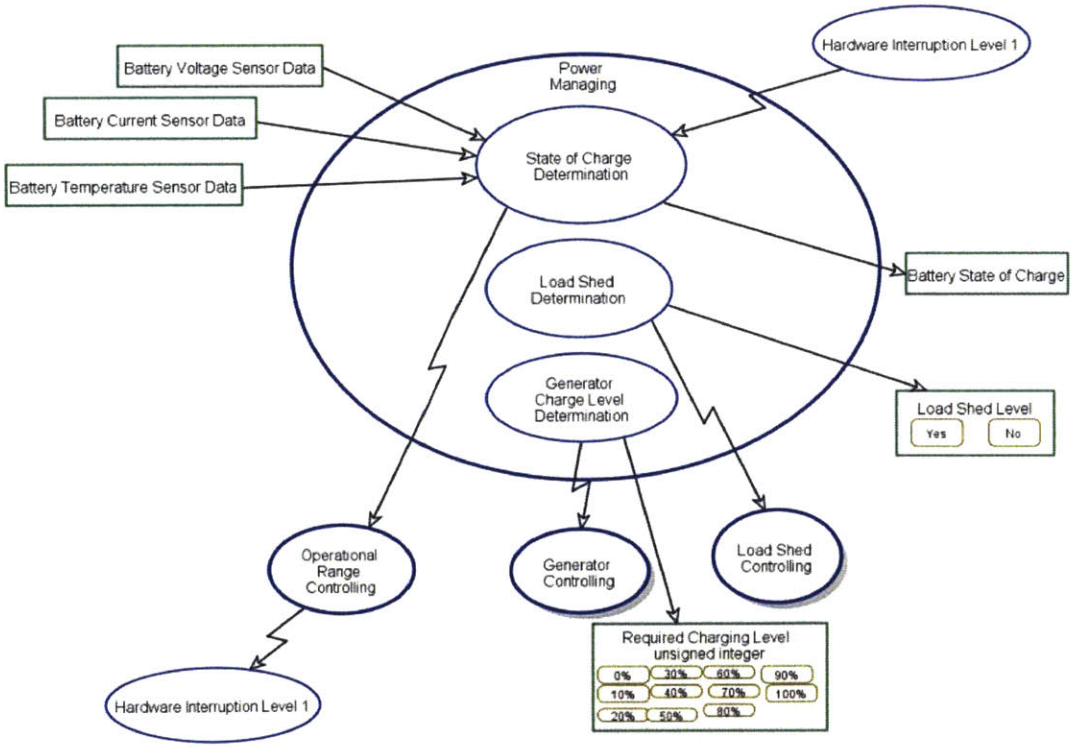


Figure 49 Power Managing in-zoomed

Required Charging Level is of type unsigned integer.
Load Shed Level can be Yes or No.
Generator Controlling is physical.
Load Shed Controlling is physical.
Operational Range Controlling invokes Hardware Interruption Level 1.
Hardware Interruption Level 1 invokes State of Charge Determination.
Power Managing zooms into State of Charge Determination, Load Shed Determination, and Generator Charge Level Determination.
State of Charge Determination consumes Battery Voltage Sensor Data, Battery Current Sensor Data, and Battery Temperature Sensor Data.
State of Charge Determination yields Battery State of Charge.
State of Charge Determination invokes Operational Range Controlling.
Load Shed Determination yields Load Shed Level.
Load Shed Determination invokes Load Shed Controlling.
Generator Charge Level Determination yields Required Charging Level.
Generator Charge Level Determination invokes Generator Controlling.

4.3.2.3.1.3 Features Interfacing in-zoomed

Figure 50 illustrates the Features Interfacing process. This process requires inputs from the Network Controlling, Time & Data Management, Inputs Monitoring and the Energy & Charging Controlling sub-processes, which interact with five parallel processes: Vehicle Access Interfacing, Security Interfacing, Engine Start Interfacing, Vehicle Safety Components Interfacing and the Lightning Controlling Interfacing that serve as a bridge to invoke the Controlling Processes Vehicle Access Controlling, Security Controlling, Starting Controlling, Safety Controlling and Lightning Controlling.

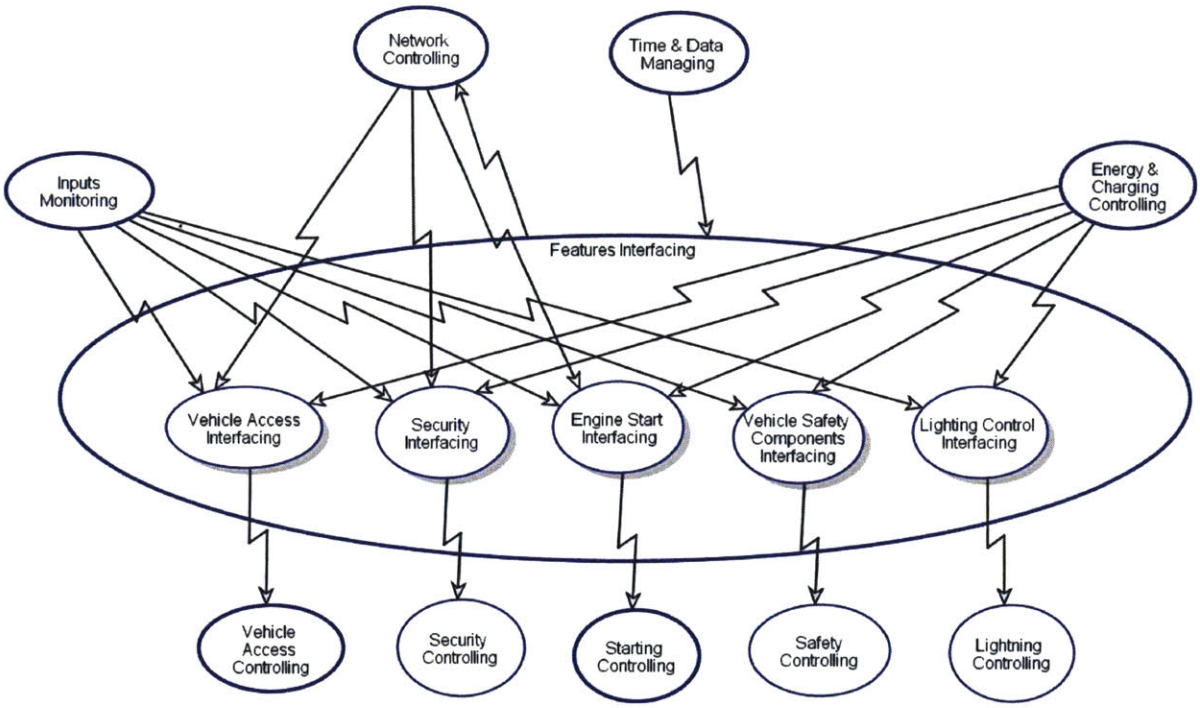


Figure 50 Features Interfacing in-zoomed

Energy & Charging Controlling invokes Engine Start Interfacing, Security Interfacing, Vehicle Safety Components Interfacing, Lighting Control Interfacing, and Vehicle Access Interfacing.
Network Controlling invokes Security Interfacing, Engine Start Interfacing, and Vehicle Access Interfacing.
Inputs Monitoring invokes Vehicle Access Interfacing, Security Interfacing, and Engine Start Interfacing.
Inputs Monitoring invokes Vehicle Safety Components Interfacing and Lighting Control Interfacing.
Time & Data Managing invokes Features Interfacing.
Features Interfacing invokes Network Controlling.

Features Interfacing zooms into Engine Start Interfacing, Lighting Control Interfacing, Security Interfacing, Vehicle Access Interfacing, and Vehicle Safety Components Interfacing.
Engine Start Interfacing is physical.
Engine Start Interfacing invokes Starting Controlling.
Lighting Control Interfacing is physical.
Lighting Control Interfacing invokes Lightning Controlling.
Security Interfacing is physical.
Security Interfacing invokes Security Controlling.
Vehicle Access Interfacing is physical.
Vehicle Access Interfacing invokes Vehicle Access Controlling.
Vehicle Safety Components Interfacing is physical.
Vehicle Safety Components Interfacing invokes Safety Controlling.

4.3.2.3.1.4 Starting Controlling

As shown in Figure 51, the Starting Controlling sub-process monitors and controls the components related to the engine start: 1) Fuel Pump Relay, 2) Starter Motor Relay (also known as Crank Relay), and Run Relay. The feature commands the components during the engine cranking in coordination with the power mode. The starting controlling event is triggered by the driver activation of the Ignition Switch when it reaches the engine crank stage. This process yields the activation of the relays when the Output Controlling is invoked.

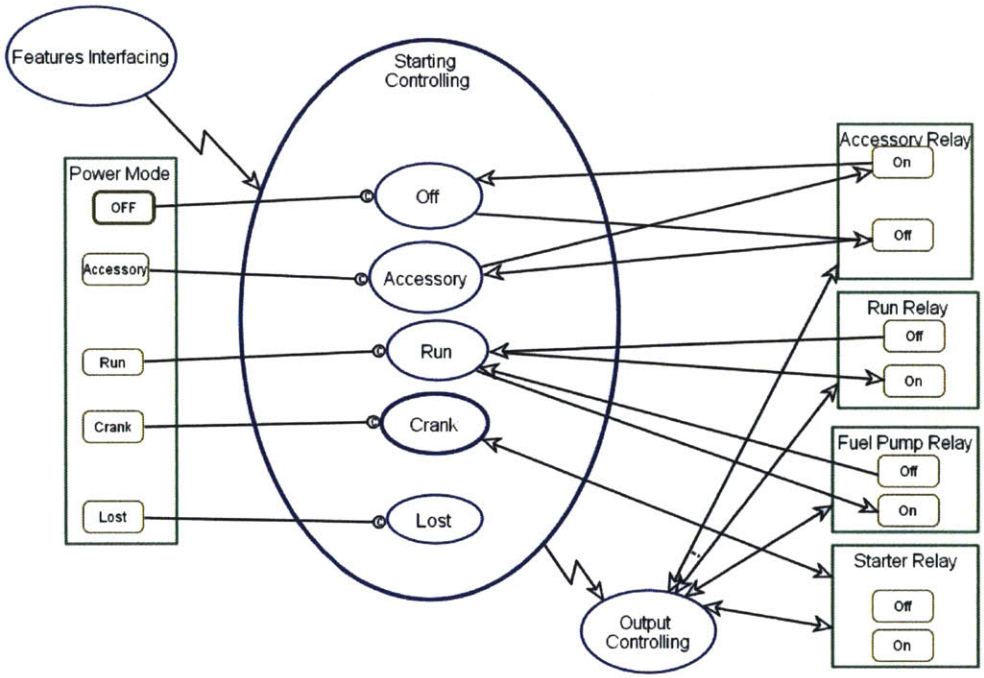


Figure 51 Starting Controlling

Power Mode can be OFF, Accessory, Run, Crank, or Lost.
OFF is initial.
Starter Relay can be On or Off.
Run Relay can be On or Off.
Accessory Relay can be On or Off.
Fuel Pump Relay can be On or Off.
Output Controlling affects either Accessory Relay or Run Relay.
Output Controlling affects Fuel Pump Relay and Starter Relay.
Features Interfacing invokes Starting Controlling.
Starting Controlling invokes Output Controlling.
Starting Controlling zooms into Off, Accessory, Run, Crank, and Lost.
Off occurs if Power Mode is OFF.

Off changes Accessory Relay from On to Off.
Accessory occurs if Power Mode is Accessory.
Accessory changes Accessory Relay from Off to On.
Run occurs if Power Mode is Run.
Run changes Run Relay from Off to On and Fuel Pump Relay from Off to On.
Crank occurs if Power Mode is Crank.
Crank affects Starter Relay.
Lost occurs if Power Mode is Lost.

4.3.2.3.1.4.1 Crank in-zoomed

The Crank process controls the starter motor and fuel pump relay when the driver selects the engine crank position with the Ignition switch. This process follows the "Run" power mode. Once the crank position is selected, the starter relay and the fuel pump shall be activated for 500 ms, then once the engine Crank process finishes, the starter relay is turned off and the Power Mode changes to Run. Figure 52 illustrates this process.

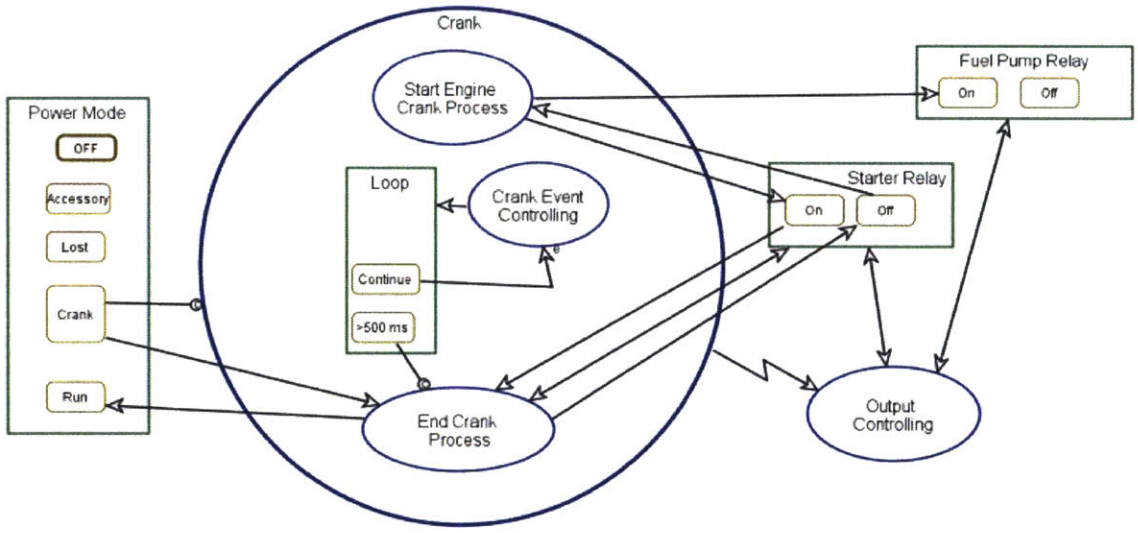


Figure 52 Crank in-zoomed

Power Mode can be OFF, Accessory, Run, Crank, or Lost.
OFF is initial.
Starter Relay can be On or Off.
Run Relay can be On or Off.
Accessory Relay can be On or Off.
Fuel Pump Relay can be On or Off.
Output Controlling affects either Accessory Relay or Run Relay.
Output Controlling affects Fuel Pump Relay and Starter Relay.
Features Interfacing invokes Starting Controlling.
Starting Controlling invokes Output Controlling.
Starting Controlling zooms into Off, Accessory, Run, Crank, and Lost.
Off occurs if Power Mode is OFF.
Off changes Accessory Relay from On to Off.
Accessory occurs if Power Mode is Accessory.
Accessory changes Accessory Relay from Off to On.
Run occurs if Power Mode is Run.
Run changes Run Relay from Off to On and Fuel Pump Relay from Off to On.
Crank occurs if Power Mode is Crank.
Crank affects Starter Relay.
Lost occurs if Power Mode is Lost.

4.3.2.3.1.5 Time & Data Management

As shown in Figure 53, this sub-process consists of a Time Counter that receives an input from the Clock Oscillator. The Time Counter keeps a base time of 1 ms that is updated to the Time variable from the data set and then taken by the Broadcast Time Change and is sent to CAN network by invoking the Features Interfacing process every second.

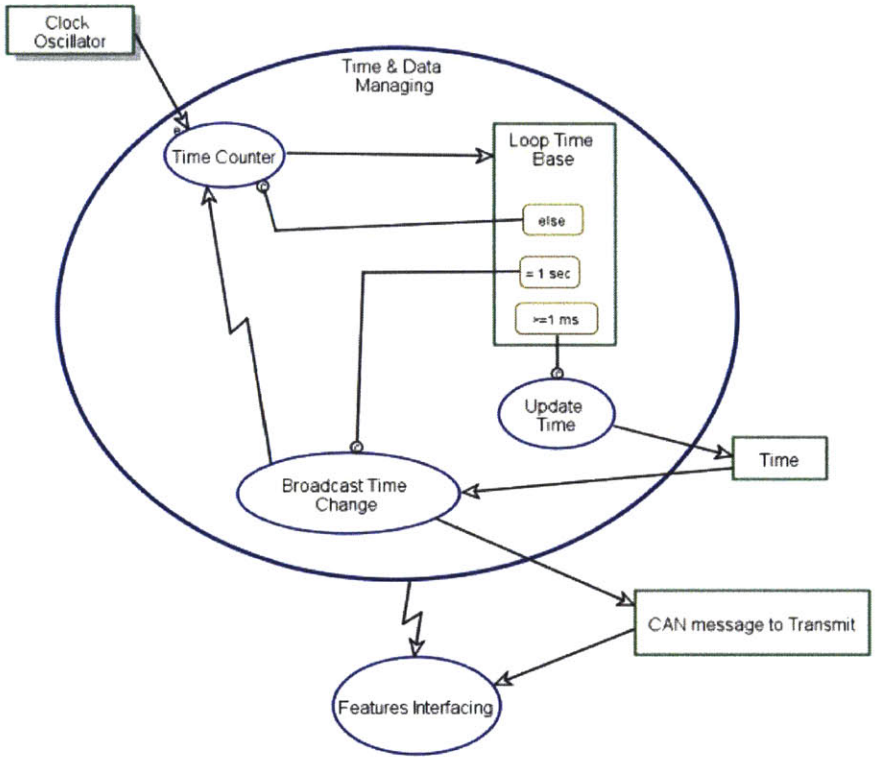


Figure 53 Time & Data Management

Clock Oscillator is physical.
Clock Oscillator triggers Time Counter.
Features Interfacing consumes CAN message to Transmit.
Time & Data Managing invokes Features Interfacing.
Time & Data Managing zooms into Time Counter, Update Time, and Broadcast Time Change, as well as Loop Time Base.
Loop Time Base can be >=1 ms or else.
Time Counter occurs if Loop Time Base is else.
Time Counter consumes Clock Oscillator.
Time Counter yields Loop Time Base.
Update Time occurs if Loop Time Base is >=1 ms.
Update Time yields Time.
Broadcast Time Change consumes Time.
Broadcast Time Change yields CAN message to Transmit.

4.3.2.3.1.6 Vehicle Access Controlling in zoomed

As shown in Figure 54, Vehicle Access is triggered by the remote control or Key Fob activation to perform any of the following commands: Doors Lock Request, Doors Unlock Request, Panic Request, and Trunk Open Request. The remote control command is processed and as a result the Door Latches. Horn, and Trunk latch could be activated by the Controlling sub-processes of Door Locks Controlling, Doors Unlock Controlling, Horn Controlling, and Trunk Latch Controlling.

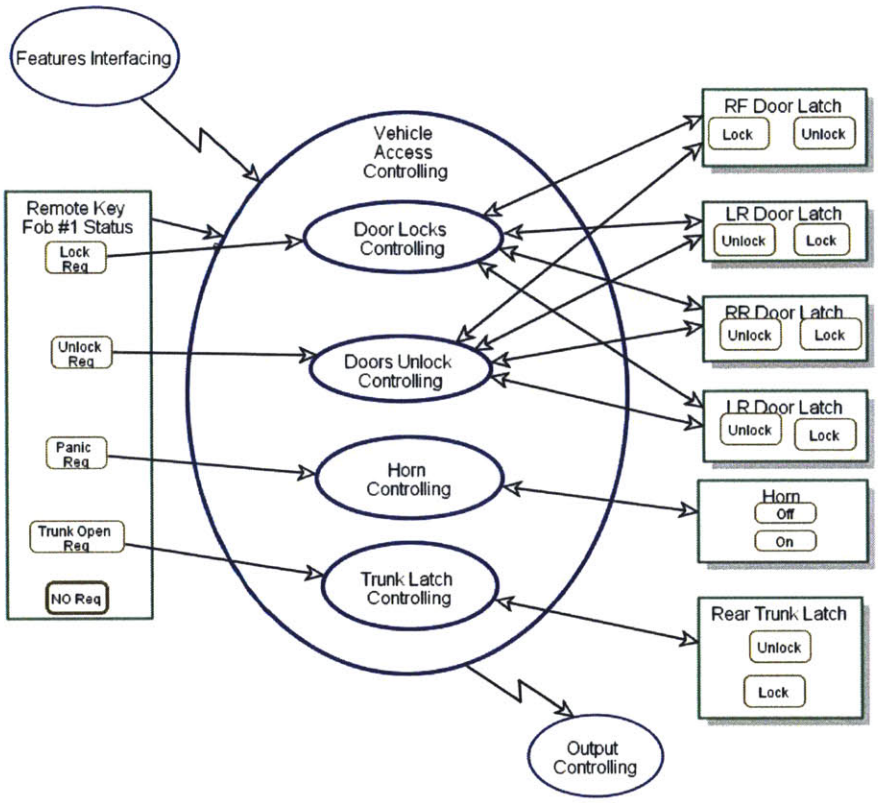


Figure 54 Vehicle Access Controlling in zoomed

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key Fob #1 Status triggers Vehicle Access Controlling when its state changes.
Horn is physical.
Horn can be Off or On.
RF Door Latch is physical.
RF Door Latch can be Lock or Unlock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RR Door Latch is physical.
RR Door Latch can be Unlock or Lock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
Rear Trunk Latch is physical.
Rear Trunk Latch can be Unlock or Lock.
Features Interfacing invokes Vehicle Access Controlling.
Vehicle Access Controlling consumes Remote Key Fob #1 Status.
Vehicle Access Controlling invokes Output Controlling.
Vehicle Access Controlling zooms into Door Locks Controlling, Doors Unlock Controlling, Horn Controlling, and Trunk Latch Controlling.
Door Locks Controlling affects RF Door Latch, LR Door Latch, RR Door Latch, and LR Door Latch.
Door Locks Controlling consumes Lock Req Remote Key Fob #1 Status.
Doors Unlock Controlling affects LR Door Latch, RR Door Latch, LR Door Latch, and RF Door Latch.
Doors Unlock Controlling consumes Unlock Req Remote Key Fob #1 Status.
Horn Controlling affects Horn.
Horn Controlling consumes Panic Req Remote Key Fob #1 Status.
Trunk Latch Controlling affects Rear Trunk Latch.
Trunk Latch Controlling consumes Trunk Open Req Remote Key Fob #1 Status.

4.3.2.3.1.6.1 Door Locks Controlling in-zoomed

As shown in Figure 55, this process receives the Remote Key Fob input which is handled by the Door Lock Processing sub-process that updates the Door Latches and the Door Latch Status and finally invokes the Output Controlling process that provides the locking pulse to latches.

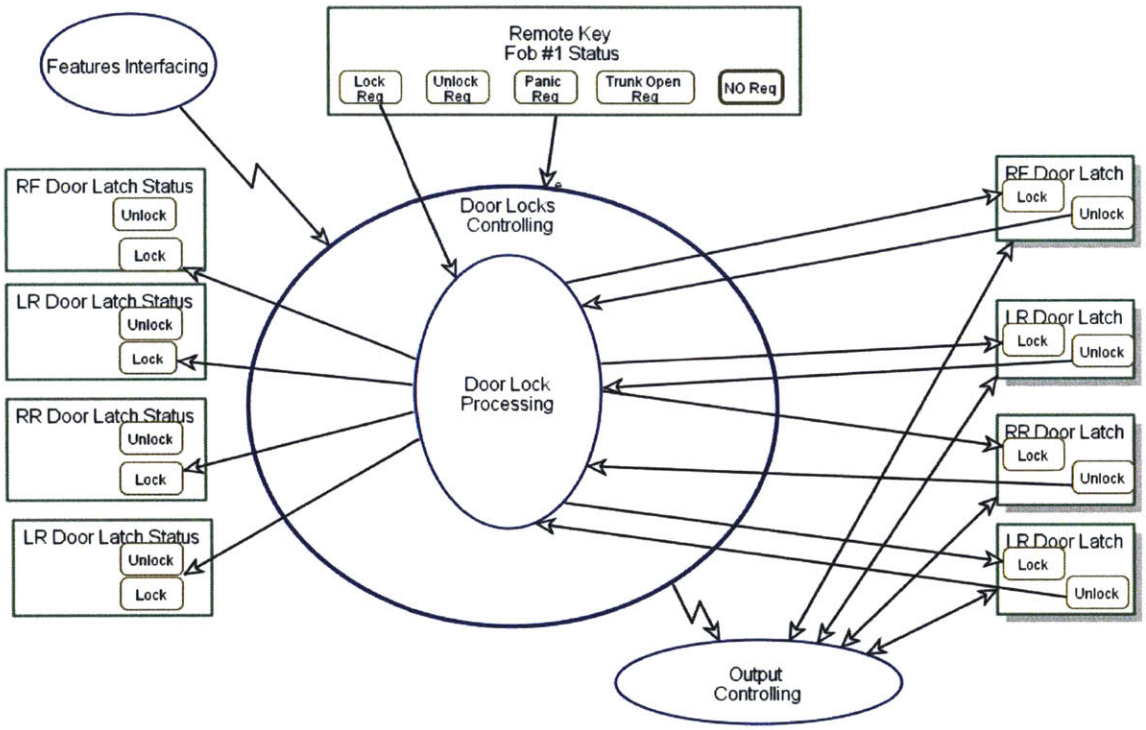


Figure 55 Door Locks Controlling in-zoomed

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key Fob #1 Status triggers Door Locks Controlling when its state changes.
RF Door Latch is physical.
RF Door Latch can be Lock or Unlock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RR Door Latch is physical.
RR Door Latch can be Unlock or Lock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RF Door Latch Status can be Unlock or Lock.
LR Door Latch Status can be Unlock or Lock.
RR Door Latch Status can be Unlock or Lock.
LR Door Latch Status can be Unlock or Lock.
Output Controlling affects LR Door Latch, RR Door Latch, LR Door Latch, and RF Door Latch.
Features Interfacing invokes Door Locks Controlling.
Door Locks Controlling consumes Remote Key Fob #1 Status.
Door Locks Controlling invokes Output Controlling.
Door Locks Controlling zooms into Door Lock Processing.
Door Lock Processing changes RF Door Latch from Unlock to Lock, LR Door Latch from Unlock to Lock, RR Door Latch from Unlock to Lock, and LR Door Latch from Unlock to Lock.
Door Lock Processing consumes Lock Req Remote Key Fob #1 Status.
Door Lock Processing yields Lock RF Door Latch Status, Lock LR Door Latch Status, Lock RR Door Latch Status, and Lock LR Door Latch Status.

4.3.2.3.1.6.2 Doors Unlock Controlling in-zoomed

As shown in Figure 56, this process receives the Remote Key Fob input, which is handled by the Door Unlock Processing sub-process that updates the Door Latches and the Door Latch Status and finally invokes the Output Controlling process that provides the unlocking pulse to latches.

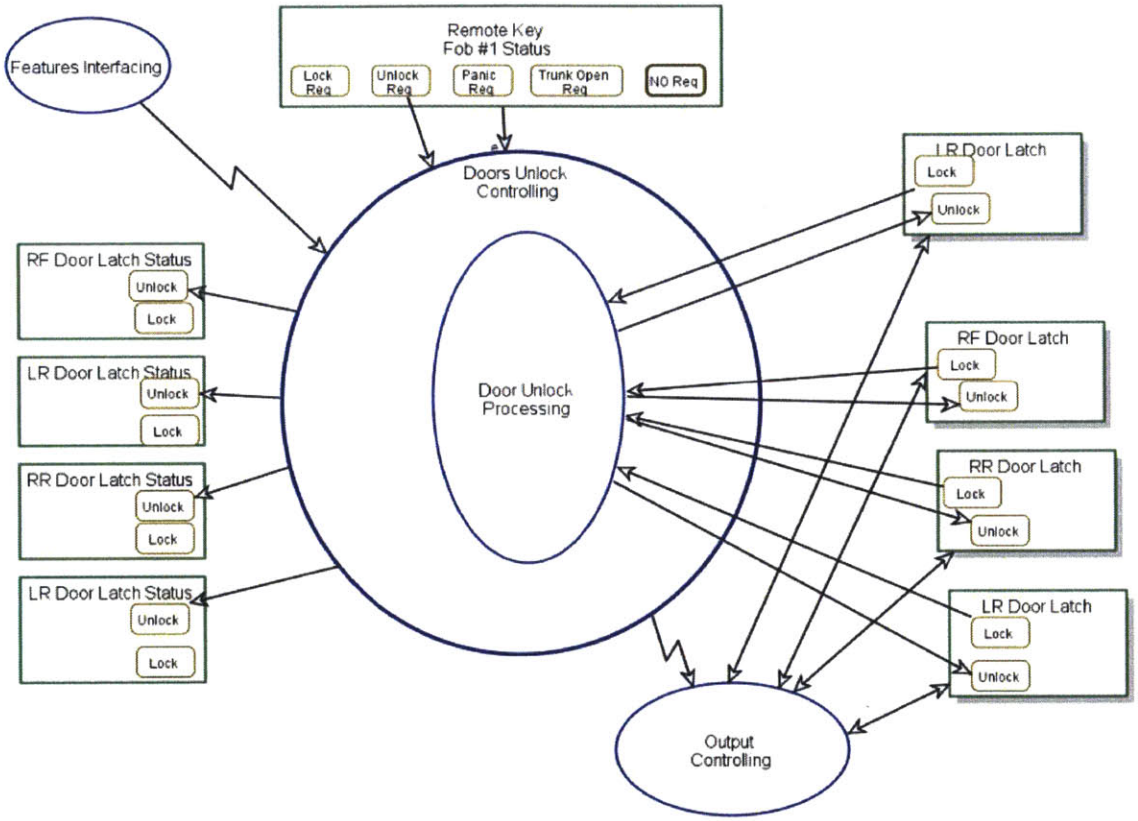


Figure 56 Doors Unlock Controlling in-zoomed

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key Fob #1 Status triggers Doors Unlock Controlling when its state changes.
RF Door Latch is physical.
RF Door Latch can be Lock or Unlock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RR Door Latch is physical.
RR Door Latch can be Unlock or Lock.
LR Door Latch is physical.
LR Door Latch can be Unlock or Lock.
RF Door Latch Status can be Unlock or Lock.
LR Door Latch Status can be Unlock or Lock.
RR Door Latch Status can be Unlock or Lock.
LR Door Latch Status can be Unlock or Lock.
Output Controlling affects LR Door Latch, RR Door Latch, LR Door Latch, and RF Door Latch.
Features Interfacing invokes Doors Unlock Controlling.
Doors Unlock Controlling consumes Unlock Req Remote Key Fob #1 Status and Remote Key Fob #1 Status.
Doors Unlock Controlling yields Unlock RF Door Latch Status, Unlock LR Door Latch Status, Unlock RR Door Latch Status, and Unlock LR Door Latch Status.
Doors Unlock Controlling invokes Output Controlling.
Doors Unlock Controlling zooms into Door Unlock Processing.
Door Unlock Processing changes LR Door Latch from Lock to Unlock, RF Door Latch from Lock to Unlock, RR Door Latch from Lock to Unlock, and LR Door Latch from Lock to Unlock.

4.3.2.3.1.6.3 Horn Controlling in-zoomed

As shown in Figure 57, this process receives the Remote Key Fob input, which is handled by the Door Horn Controlling sub-process that updates the Horn and the Horn Status and finally invokes the Output Controlling process that provides the horn activation. This command is known as Panic Event, transmitted by the vehicle driver via the Remote Key Fob. The Horn sounding is controlled by the Loop process that commands the horn sounding for 500 ms On and 500 ms Off cycle.

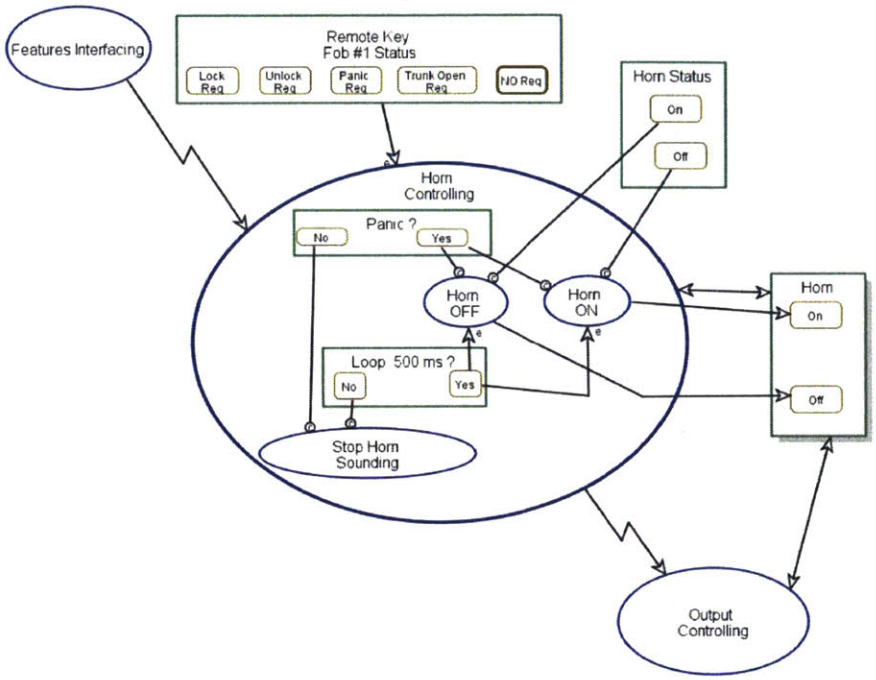


Figure 57 Horn Controlling in-zoomed

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key Fob #1 Status triggers Horn Controlling when its state changes.
Horn is physical.
Horn can be Off or On.
Horn Status can be On or Off.
Output Controlling affects Horn.
Features Interfacing invokes Horn Controlling.
Horn Controlling affects Horn.
Horn Controlling consumes Remote Key Fob #1 Status.
Horn Controlling invokes Output Controlling.
Horn Controlling zooms into Horn ON, Horn OFF, and Stop Horn Sounding, as well as Loop ? and Panic ?.
Loop ? can be Yes or No.
Loop ? triggers Horn OFF and Horn ON when it enters Yes.
Panic ? can be Yes or No.
Horn ON occurs if Horn Status is Off and Panic ? is Yes.
Horn ON consumes Yes Loop ?.
Horn ON yields On Horn.
Horn OFF occurs if Horn Status is On and Panic ? is Yes.
Horn OFF consumes Yes Loop ?.
Horn OFF yields Off Horn.
Stop Horn Sounding occurs if Loop ? is No and Panic ? is No.

4.3.2.3.1.6.4 Trunk Latch Controlling

As shown in Figure 58, this process receives the Remote Key Fob input which is handled by the Trunk Unlocking sub-process that updates the Trunk Latch and the Trunk Latch Status, and finally invokes the Output Controlling process that provides the unlocking pulse to latch.

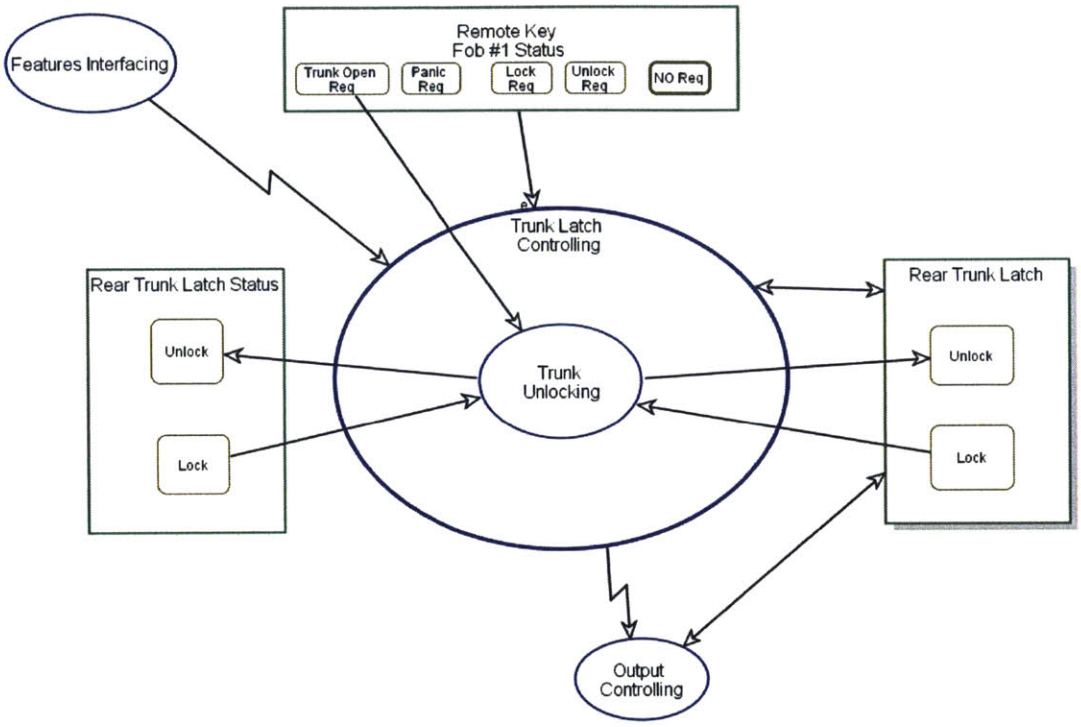


Figure 58 Trunk Latch Controlling in-zooming

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key Fob #1 Status triggers Trunk Latch Controlling when its state changes.
Rear Trunk Latch is physical.
Rear Trunk Latch can be Unlock or Lock.
Rear Trunk Latch Status can be Unlock or Lock.
Output Controlling affects Rear Trunk Latch.
Features Interfacing invokes Trunk Latch Controlling.
Trunk Latch Controlling affects Rear Trunk Latch.
Trunk Latch Controlling consumes Remote Key Fob #1 Status.
Trunk Latch Controlling invokes Output Controlling.
Trunk Latch Controlling zooms into Trunk Unlocking.
Trunk Unlocking changes Rear Trunk Latch from Lock to Unlock and Rear Trunk Latch Status from Lock to Unlock.
Trunk Unlocking consumes Trunk Open Req Remote Key Fob #1 Status.

4.3.2.3.1.7 Inputs Monitoring in-zoomed

As shown in Figure 59, this sub-process monitors the hardware inputs from the Door Ajar and the Remote Key Fob. There are two types of events:
Event type 1: Door Opened or Door closed triggers the Door Switch Monitoring for any of the vehicle doors: LR Door, LF Door, RF Door, RR Door, and Trunk.
Event type 2: Remote Key Fob command triggers Fob Actuation Monitoring. Then the output from the process yields on a Hardware Interruption Level 3.

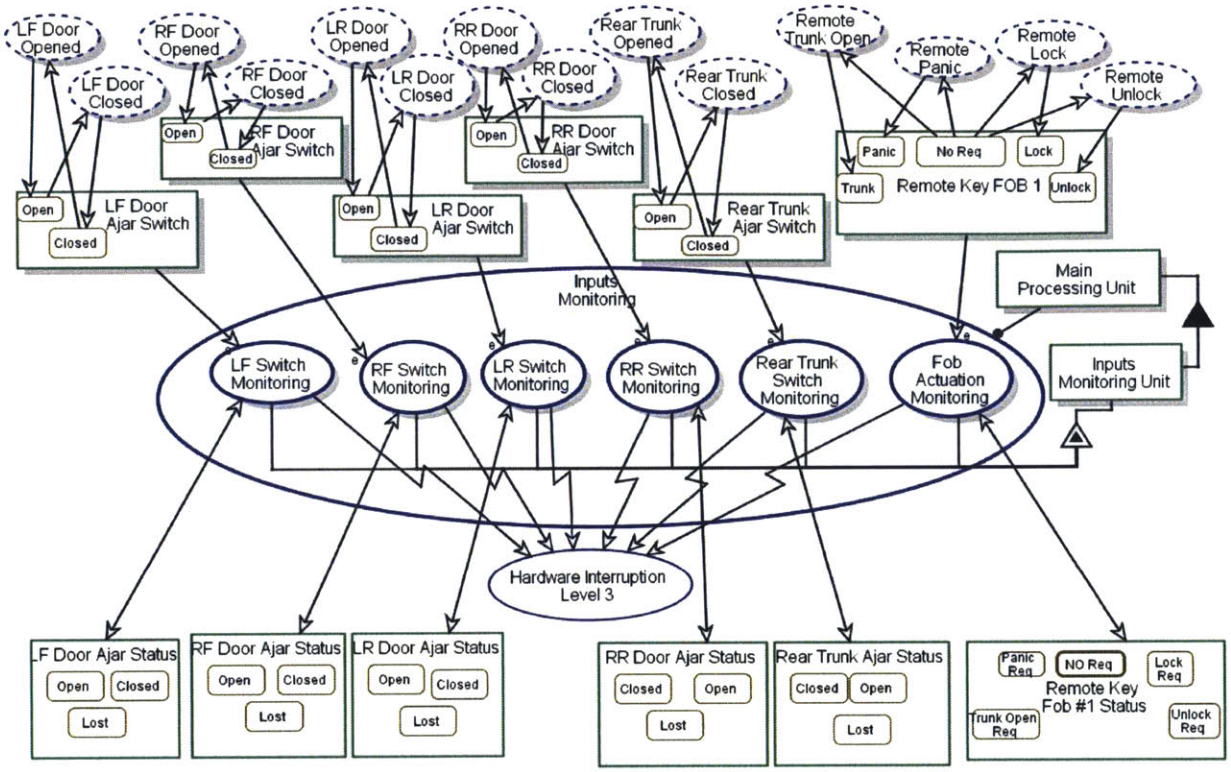


Figure 59 Inputs Monitoring in-zoomed

Main Processing Unit is physical.
Main Processing Unit consists of Inputs Monitoring Unit.
    Inputs Monitoring Unit is physical.
    Inputs Monitoring Unit exhibits LF Switch Monitoring, RF Switch Monitoring, LR Switch Monitoring, RR Switch Monitoring, Rear Trunk Switch Monitoring, and Fob Actuation Monitoring.
Main Processing Unit handles Inputs Monitoring.
LF Door Ajar Status can be Open, Closed, or Lost.
LR Door Ajar Status can be Open, Closed, or Lost.
RF Door Ajar Status can be Open, Closed, or Lost.
RR Door Ajar Status can be Closed, Open, or Lost.
Rear Trunk Ajar Status can be Closed, Open, or Lost.
Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
    NO Req is initial.
LF Door Ajar Switch is physical.
LF Door Ajar Switch can be Open or Closed.
LF Door Ajar Switch triggers LF Switch Monitoring when its state changes.
RF Door Ajar Switch is physical.
RF Door Ajar Switch can be Open or Closed.
RF Door Ajar Switch triggers RF Switch Monitoring when its state changes.
LR Door Ajar Switch is physical.
LR Door Ajar Switch can be Open or Closed.
LR Door Ajar Switch triggers LR Switch Monitoring when its state changes.
RR Door Ajar Switch is physical.
RR Door Ajar Switch can be Open or Closed.
RR Door Ajar Switch triggers RR Switch Monitoring when its state changes.
Rear Trunk Ajar Switch is physical.
Rear Trunk Ajar Switch can be Open or Closed.
Rear Trunk Ajar Switch triggers Rear Trunk Switch Monitoring when its state changes.
Remote Key FOB 1 is physical.
Remote Key FOB 1 can be Panic, No Req, Lock, Unlock, or Trunk.
Remote Key FOB 1 triggers Fob Actuation Monitoring when its state changes.
LF Door Opened is environmental and physical.

LF Door Opened changes LF Door Ajar Switch from Closed to Open.
Remote Unlock is environmental and physical.
Remote Unlock changes Remote Key FOB 1 from No Req to Unlock.
LF Door Closed is environmental and physical.
LF Door Closed changes LF Door Ajar Switch from Open to Closed.
RF Door Closed is environmental and physical.
RF Door Closed changes RF Door Ajar Switch from Open to Closed.
RF Door Opened is environmental and physical.
RF Door Opened changes RF Door Ajar Switch from Closed to Open.
LR Door Closed is environmental and physical.
LR Door Closed changes LR Door Ajar Switch from Open to Closed.
LR Door Opened is environmental and physical.
LR Door Opened changes LR Door Ajar Switch from Closed to Open.
RR Door Closed is environmental and physical.
RR Door Closed changes RR Door Ajar Switch from Open to Closed.
RR Door Opened is environmental and physical.
RR Door Opened changes RR Door Ajar Switch from Closed to Open.
Rear Trunk Closed is environmental and physical.
Rear Trunk Closed changes Rear Trunk Ajar Switch from Open to Closed.
Rear Trunk Opened is environmental and physical.
Rear Trunk Opened changes Rear Trunk Ajar Switch from Closed to Open.
Remote Lock is environmental and physical.
Remote Lock changes Remote Key FOB 1 from No Req to Lock.
Remote Panic is environmental and physical.
Remote Panic changes Remote Key FOB 1 from No Req to Panic.
Remote Trunk Open is environmental and physical.
Remote Trunk Open changes Remote Key FOB 1 from No Req to Trunk.
Inputs Monitoring zooms into Fob Actuation Monitoring, Rear Trunk Switch Monitoring, RR Switch Monitoring, LR Switch Monitoring, RF Switch Monitoring, and LF Switch Monitoring.

- Fob Actuation Monitoring is physical.
- Fob Actuation Monitoring affects Remote Key Fob #1 Status.
- Fob Actuation Monitoring consumes Remote Key FOB 1.
- Fob Actuation Monitoring invokes Hardware Interruption Level 3.
- Rear Trunk Switch Monitoring is physical.
- Rear Trunk Switch Monitoring affects Rear Trunk Ajar Status.
- Rear Trunk Switch Monitoring consumes Rear Trunk Ajar Switch.
- Rear Trunk Switch Monitoring invokes Hardware Interruption Level 3.
- RR Switch Monitoring is physical.
- RR Switch Monitoring affects RR Door Ajar Status.
- RR Switch Monitoring consumes RR Door Ajar Switch.
- RR Switch Monitoring invokes Hardware Interruption Level 3.
- LR Switch Monitoring is physical.
- LR Switch Monitoring affects LR Door Ajar Status.
- LR Switch Monitoring consumes LR Door Ajar Switch.
- LR Switch Monitoring invokes Hardware Interruption Level 3.
- RF Switch Monitoring is physical.
- RF Switch Monitoring affects RF Door Ajar Status.
- RF Switch Monitoring consumes RF Door Ajar Switch.
- RF Switch Monitoring invokes Hardware Interruption Level 3.
- LF Switch Monitoring is physical.
- LF Switch Monitoring affects LF Door Ajar Status.
- LF Switch Monitoring consumes LF Door Ajar Switch.
- LF Switch Monitoring invokes Hardware Interruption Level 3.

4.3.2.3.1.7.1 Fob Actuation Monitoring in-zoomed

As shown in Figure 60, this process monitors the Remote Key Fob actuation. It verifies the validity of the command received from the remote control device and yields a Hardware Interruption Level 3 if the command received is valid.

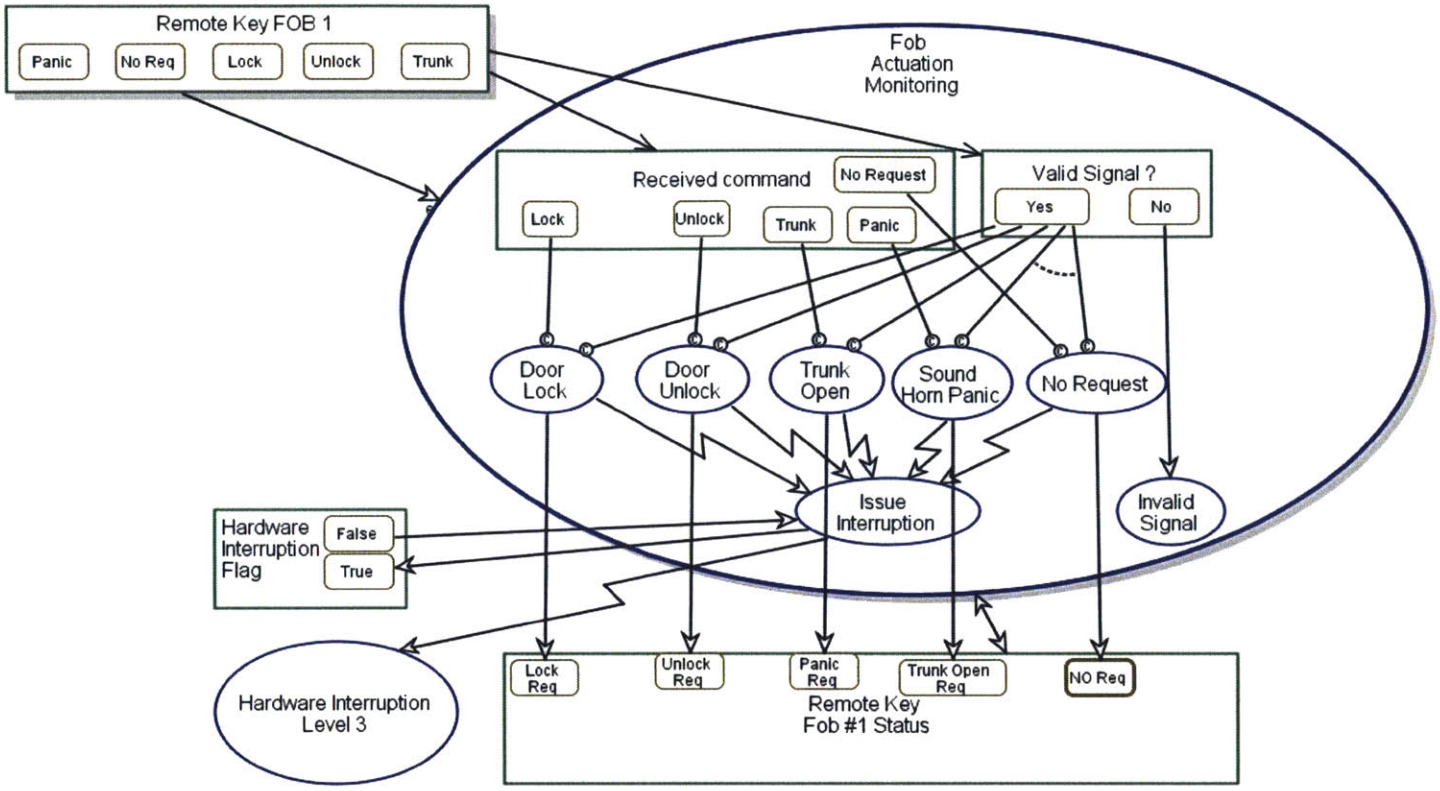


Figure 60 Fob Actuation Monitoring in-zoomed

Remote Key Fob #1 Status can be Lock Req, Unlock Req, Panic Req, Trunk Open Req, or NO Req.
NO Req is initial.
Remote Key FOB 1 is physical.
Remote Key FOB 1 can be Panic, No Req, Lock, Unlock, or Trunk.
Remote Key FOB 1 relates to Received command.
Remote Key FOB 1 relates to Valid Signal ?.
Remote Key FOB 1 triggers Fob Actuation Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
Fob Actuation Monitoring is physical.
Fob Actuation Monitoring affects Remote Key Fob #1 Status.
Fob Actuation Monitoring consumes Remote Key FOB 1.
Fob Actuation Monitoring zooms into Trunk Open, Sound Horn Panic, Door Lock, Door Unlock, No Request, Invalid Signal, and Issue Interruption, as well as Received command and Valid Signal ?.
Received command can be Lock, Unlock, Trunk, Panic, or No Request.
Valid Signal ? can be Yes or No.
Trunk Open occurs if Valid Signal ? is Yes and Received command is Trunk.
Trunk Open yields Panic Req Remote Key Fob #1 Status.
Trunk Open invokes Issue Interruption.
Sound Horn Panic occurs if Valid Signal ? is Yes and Received command is Panic.
Sound Horn Panic yields Trunk Open Req Remote Key Fob #1 Status.
Sound Horn Panic invokes Issue Interruption.
Door Lock occurs if Valid Signal ? is Yes and Received command is Lock.
Door Lock yields Lock Req Remote Key Fob #1 Status.
Door Lock invokes Issue Interruption.
Door Unlock occurs if Valid Signal ? is Yes and Received command is Unlock.
Door Unlock yields Unlock Req Remote Key Fob #1 Status.
Door Unlock invokes Issue Interruption.
No Request occurs if Valid Signal ? is Yes and Received command is No Request.
No Request yields NO Req Remote Key Fob #1 Status.
No Request invokes Issue Interruption.
Invalid Signal consumes No Valid Signal ?.
Issue Interruption changes Hardware Interruption Flag from False to True.
Issue Interruption invokes Hardware Interruption Level 3.

4.3.2.3.1.7.2 LF Switch Monitoring in-zoomed

As shown in Figure 61, this process monitors the Left Front Door switch activation when a door opened or a door closed event occurs. It then verifies the validity of the signal received from the door and yields a Hardware Interruption Level 3 if the signal received is valid.

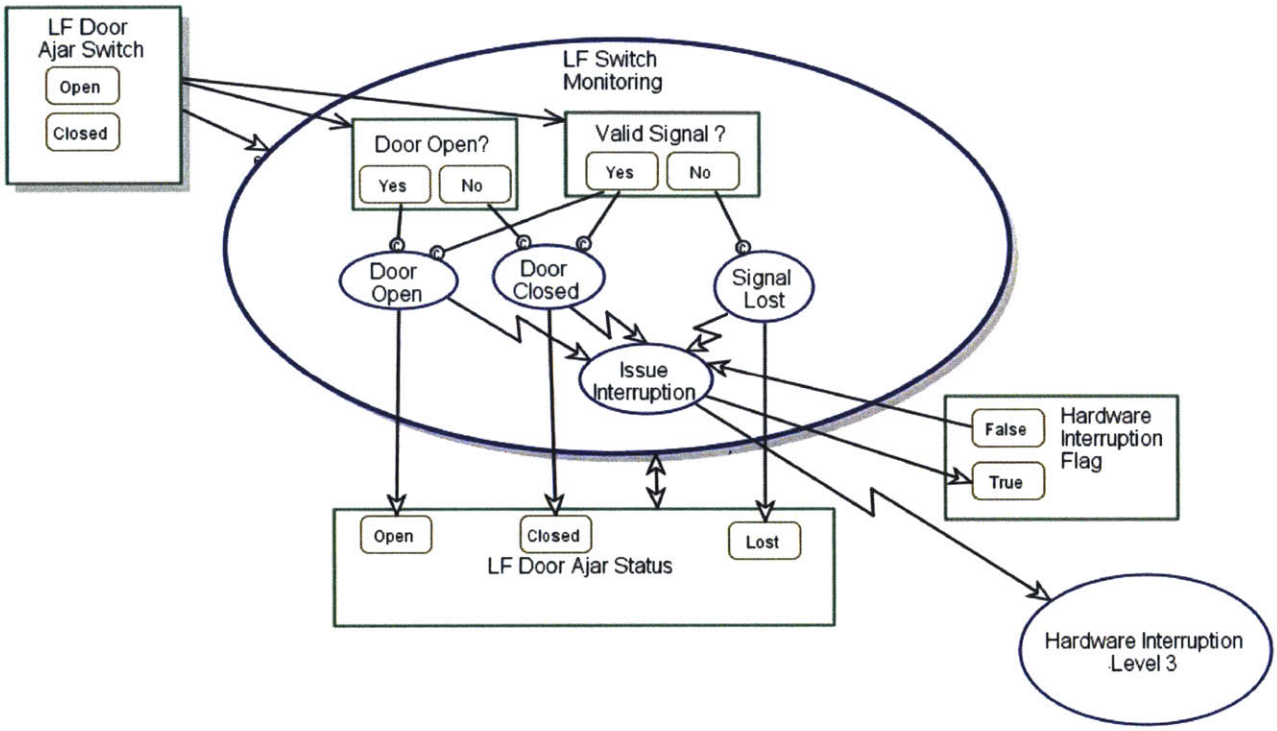


Figure 61 LF Switch Monitoring in-zoomed

LF Door Ajar Status can be Open, Closed, or Lost.
LF Door Ajar Switch is physical.
LF Door Ajar Switch can be Open or Closed.
LF Door Ajar Switch relates to Door Open?.
LF Door Ajar Switch relates to Valid Signal ?.
LF Door Ajar Switch triggers LF Switch Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
LF Switch Monitoring is physical.
LF Switch Monitoring affects LF Door Ajar Status.
LF Switch Monitoring consumes LF Door Ajar Switch.
LF Switch Monitoring zooms into Door Closed, Signal Lost, Door Open, and Issue Interruption, as well as Door Open? and Valid Signal ?.

- Door Open? can be Yes or No.
- Valid Signal ? can be Yes or No.
- Door Closed occurs if Valid Signal ? is Yes and Door Open? is No.
- Door Closed yields Closed LF Door Ajar Status.
- Door Closed invokes Issue Interruption.
- Signal Lost occurs if Valid Signal ? is No.
- Signal Lost yields Lost LF Door Ajar Status.
- Signal Lost invokes Issue Interruption.
- Door Open occurs if Valid Signal ? is Yes and Door Open? is Yes.
- Door Open yields Open LF Door Ajar Status.
- Door Open invokes Issue Interruption.
- Issue Interruption changes Hardware Interruption Flag from False to True.
- Issue Interruption invokes Hardware Interruption Level 3.

4.3.2.3.1.7.3 LR Switch Monitoring in-zoomed

As shown in Figure 62, this process monitors the Left Rear Door switch activation when a door opened or a door closed event occurs. It then verifies the validity of the signal received from the door and yields on a Hardware Interruption Level 3 if the signal received is valid.

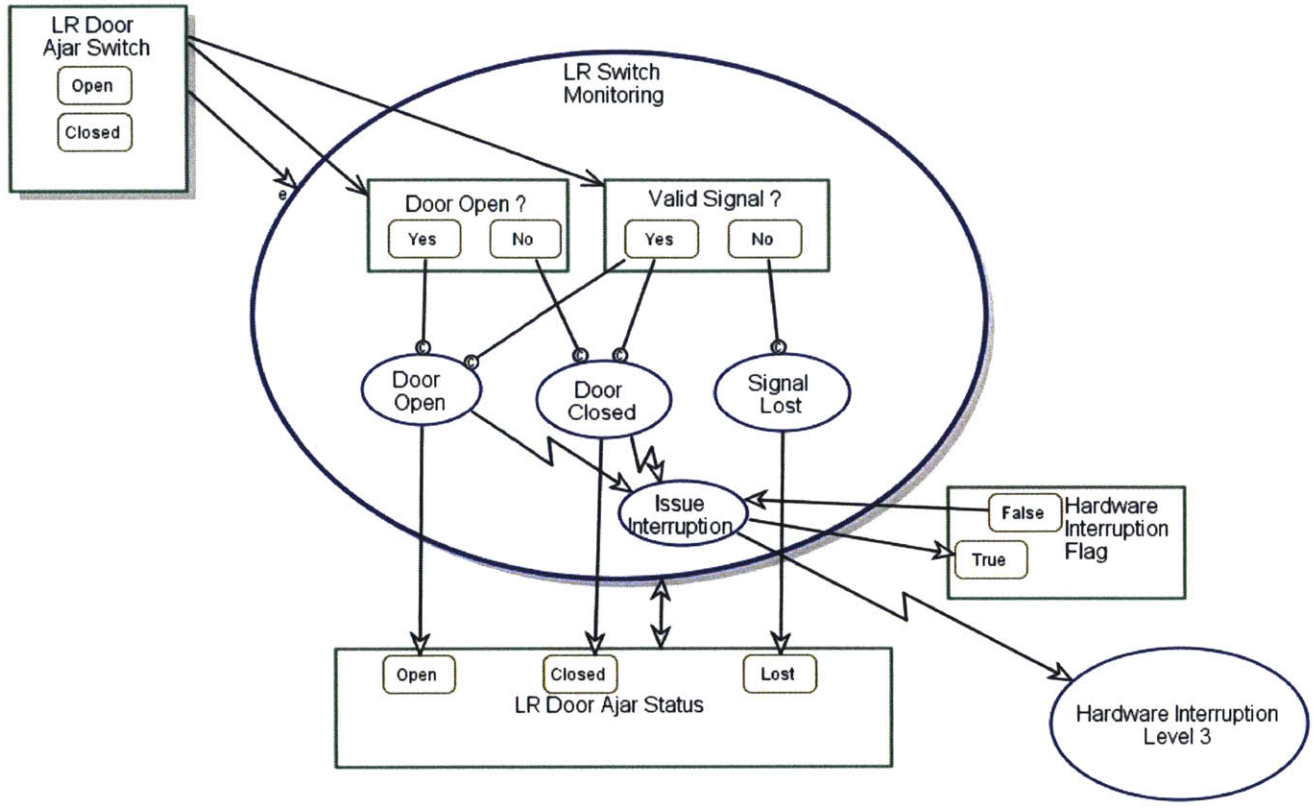


Figure 62 LR Switch Monitoring in-zoomed

LR Door Ajar Status can be Open, Closed, or Lost.
LR Door Ajar Switch is physical.
LR Door Ajar Switch can be Open or Closed.
LR Door Ajar Switch relates to Door Open ?.
LR Door Ajar Switch relates to Valid Signal ?.
LR Door Ajar Switch triggers LR Switch Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
LR Switch Monitoring is physical.
LR Switch Monitoring affects LR Door Ajar Status.
LR Switch Monitoring consumes LR Door Ajar Switch.
LR Switch Monitoring zooms into Signal Lost, Door Open, Door Closed, and Issue Interruption, as well as Door Open ? and Valid Signal ?.

- Door Open ? can be Yes or No.
- Valid Signal ? can be Yes or No.
- Signal Lost occurs if Valid Signal ? is No.
- Signal Lost yields Lost LR Door Ajar Status.
- Door Open occurs if Valid Signal ? is Yes and Door Open ? is Yes.
- Door Open yields Open LR Door Ajar Status.
- Door Open invokes Issue Interruption.
- Door Closed occurs if Valid Signal ? is Yes and Door Open ? is No.
- Door Closed yields Closed LR Door Ajar Status.
- Door Closed invokes Issue Interruption.
- Issue Interruption changes Hardware Interruption Flag from False to True.
- Issue Interruption invokes Hardware Interruption Level 3.

4.3.2.3.1.7.4 RF Switch Monitoring in-zoomed

As shown in Figure 63, this process monitors the Right Front Door switch activation when a door opened or a door closed event occurs. It then verifies the validity of the signal received from the door and yields on a Hardware Interruption Level 3 if the signal received is valid.

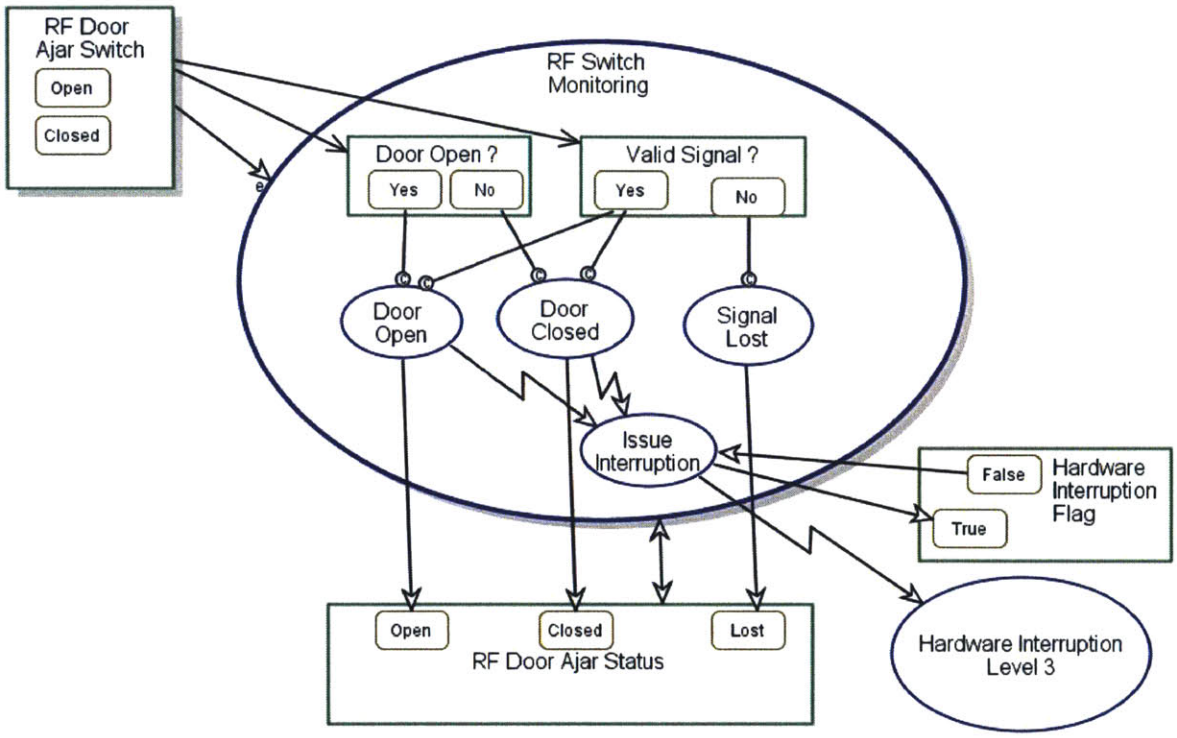


Figure 63 RF Switch Monitoring in-zoomed

RF Door Ajar Status can be Open, Closed, or Lost.
RF Door Ajar Switch is physical.
RF Door Ajar Switch can be Open or Closed.
RF Door Ajar Switch relates to Door Open ?.
RF Door Ajar Switch relates to Valid Signal ?.
RF Door Ajar Switch triggers RF Switch Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
RF Switch Monitoring is physical.
RF Switch Monitoring affects RF Door Ajar Status.
RF Switch Monitoring consumes RF Door Ajar Switch.
RF Switch Monitoring zooms into Door Closed, Signal Lost, Door Open, and Issue Interruption, as well as Door Open ? and Valid Signal ?.

- Door Open ? can be Yes or No.
- Valid Signal ? can be Yes or No.
- Door Closed occurs if Valid Signal ? is Yes and Door Open ? is No.
- Door Closed yields Closed RF Door Ajar Status.
- Door Closed invokes Issue Interruption.
- Signal Lost occurs if Valid Signal ? is No.
- Signal Lost yields Lost RF Door Ajar Status.
- Door Open occurs if Valid Signal ? is Yes and Door Open ? is Yes.
- Door Open yields Open RF Door Ajar Status.
- Door Open invokes Issue Interruption.
- Issue Interruption changes Hardware Interruption Flag from False to True.
- Issue Interruption invokes Hardware Interruption Level 3.

4.3.2.3.1.7.5 RR Switch Monitoring in-zoomed

As shown in Figure 64, this process monitors the Right Rear Door switch activation when a door opened or a door closed event occurs, then it verifies the validity of the signal received from the door and yields on a Hardware Interruption Level 3 if the signal received is valid.

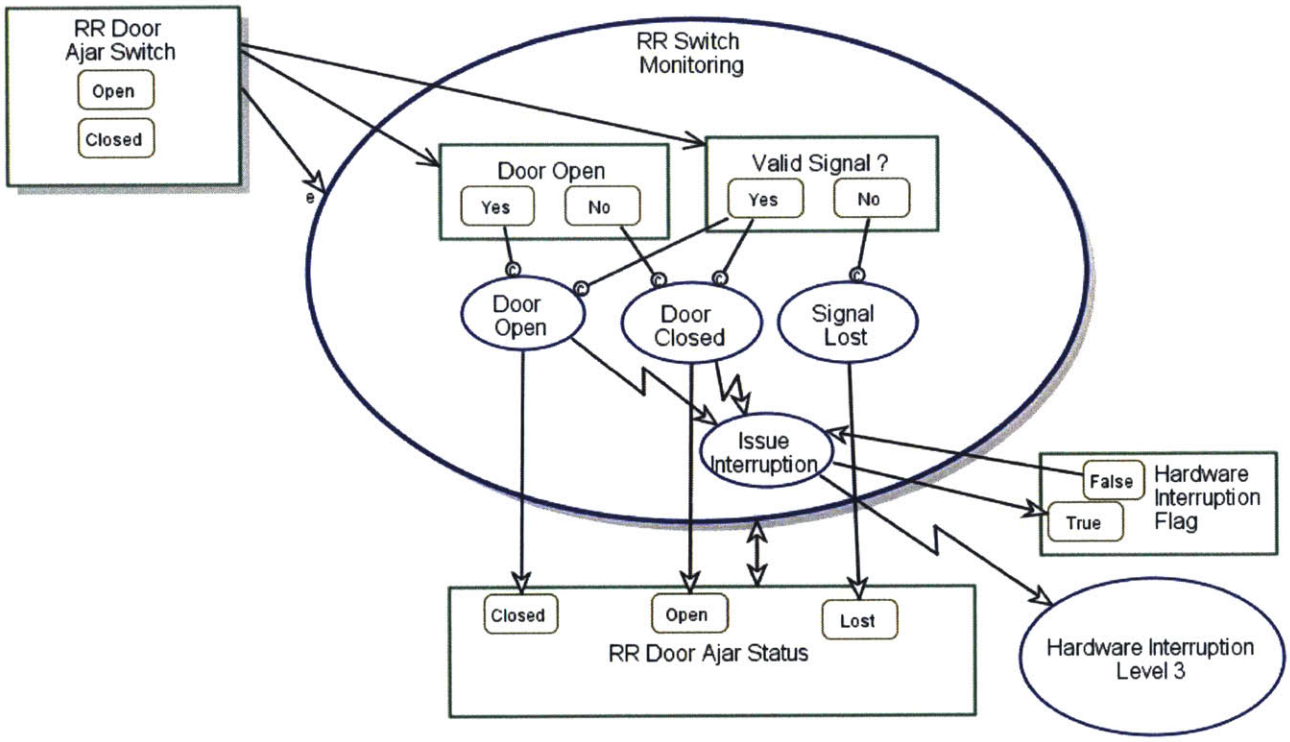


Figure 64 RR Switch Monitoring in-zoomed

RR Door Ajar Status can be Closed, Open, or Lost.
RR Door Ajar Switch is physical.
RR Door Ajar Switch can be Open or Closed.
RR Door Ajar Switch relates to Door Open.
RR Door Ajar Switch relates to Valid Signal ?.
RR Door Ajar Switch triggers RR Switch Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
RR Switch Monitoring is physical.
RR Switch Monitoring affects RR Door Ajar Status.
RR Switch Monitoring consumes RR Door Ajar Switch.
RR Switch Monitoring zooms into Door Open, Signal Lost, Door Closed, and Issue Interruption, as well as Door Open and Valid Signal ?.

- Door Open can be Yes or No.
- Valid Signal ? can be Yes or No.
- Door Open occurs if Valid Signal ? is Yes and Door Open is Yes.
- Door Open yields Closed RR Door Ajar Status.
- Door Open invokes Issue Interruption.
- Signal Lost occurs if Valid Signal ? is No.
- Signal Lost yields Lost RR Door Ajar Status.
- Door Closed occurs if Valid Signal ? is Yes and Door Open is No.
- Door Closed yields Open RR Door Ajar Status.
- Door Closed invokes Issue Interruption.
- Issue Interruption changes Hardware Interruption Flag from False to True.
- Issue Interruption invokes Hardware Interruption Level 3.

4.3.2.3.1.7.6 Rear Trunk Switch Monitoring in-zoomed

As shown in Figure 65, this process monitors the Rear Trunk switch activation when a door opened or a door closed event occurs, then it verifies the validity of the signal received from the door and yields on a Hardware Interruption Level 3 if the signal received is valid.

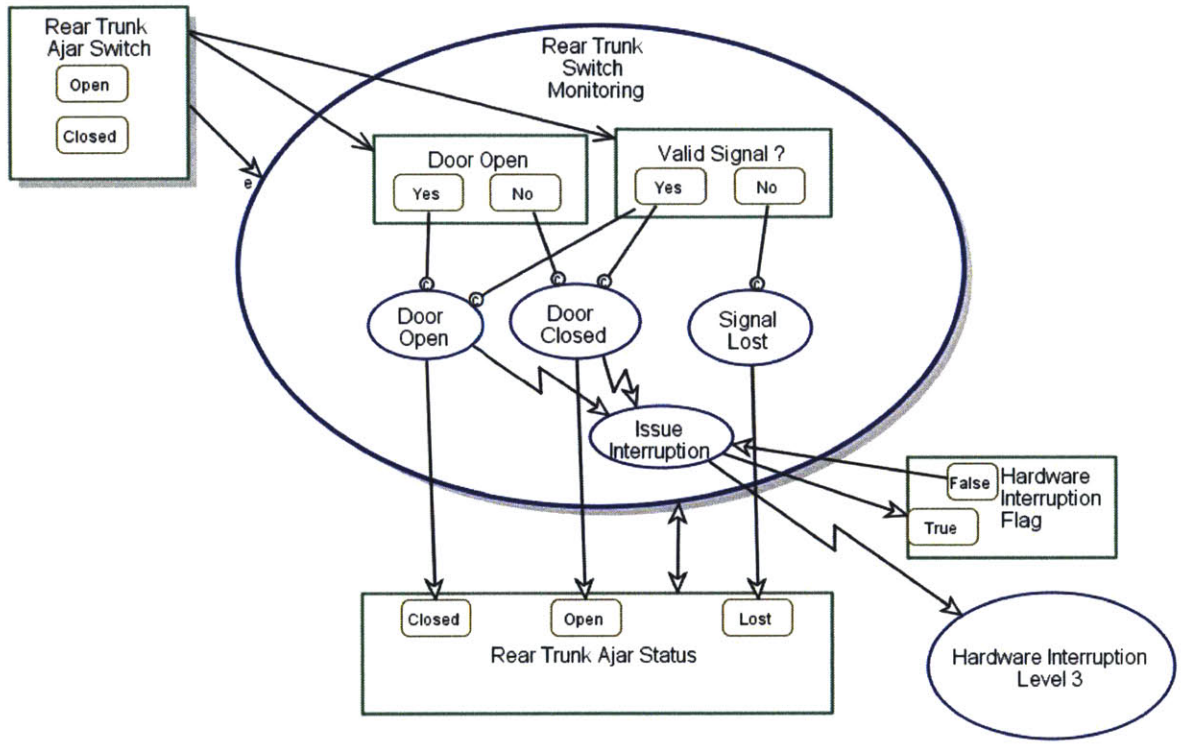


Figure 65 Rear Trunk Switch Monitoring in-zoomed

Rear Trunk Ajar Status can be Closed, Open, or Lost.
Rear Trunk Ajar Switch is physical.
Rear Trunk Ajar Switch can be Open or Closed.
Rear Trunk Ajar Switch relates to Door Open.
Rear Trunk Ajar Switch relates to Valid Signal ?.
Rear Trunk Ajar Switch triggers Rear Trunk Switch Monitoring when its state changes.
Hardware Interruption Flag can be True or False.
Rear Trunk Switch Monitoring is physical.
Rear Trunk Switch Monitoring affects Rear Trunk Ajar Status.
Rear Trunk Switch Monitoring consumes Rear Trunk Ajar Switch.
Rear Trunk Switch Monitoring zooms into Door Closed, Signal Lost, Door Open, and Issue Interruption, as well as Door Open and Valid Signal ?.

- Door Open can be Yes or No.
- Valid Signal ? can be Yes or No.
- Door Closed occurs if Valid Signal ? is Yes and Door Open is No.
- Door Closed yields Open Rear Trunk Ajar Status.
- Door Closed invokes Issue Interruption.
- Signal Lost occurs if Valid Signal ? is No.
- Signal Lost yields Lost Rear Trunk Ajar Status.
- Door Open occurs if Valid Signal ? is Yes and Door Open is Yes.
- Door Open yields Closed Rear Trunk Ajar Status.
- Door Open invokes Issue Interruption.
- Issue Interruption changes Hardware Interruption Flag from False to True.
- Issue Interruption invokes Hardware Interruption Level 3.

4.4 Conclusion of Chapter 4

The model created with OPM contains the generic elements of a Body Control Module (BCM), generally used to manage the body electronics subsystem in an automobile. Although the complexity and the number of subsystems in the automobile could be vast, the OPM model presented in this chapter contains some of the main top-level functions that are usually incorporated in a BCM.

CHAPTER 5 GUIDELINES FOR AUTOMOTIVE ELECTRONIC SYSTEMS SOFTWARE DEVELOPMENT

The previous chapters introduced the complexity of automotive systems and the Systems Engineering point of view that was used to help understand the relationship between the components that form the different vehicle subsystems.

The OPM model of a Body Control Module (BCM) introduced in Chapter 4 provided the basic ECU structure used in the automotive industry at the physical and functional levels. The following guidelines complement the OPM model for electronic systems software development.

5.1 Guidelines for Hardware elements in automobile systems

1. An electronic control module to be used in an automobile system shall be designed considering the system architecture, the functional sub-system and the expected interactions between the control module the transducers and actuators related to the functional feature in the subsystem, without taking the unknown system definition for granted and omitting specific details.

   Motivation: In several experiences documented on section 3.2.4, during the design of ECU's the programs faced challenges to achieve the desired function due to a misunderstanding of the system architecture, causing system faults at the hardware level.

2. The Inputs and Outputs used in the electronic control module shall be managed by pre-defined interface circuitry that consistently meets the OEM standards for robust design.

   Motivation: Many problems found, in the experiences documented on section 3.2.4, caused systemic faults due to new ECU designs that featured circuit interfaces with components not recommended by the OEM or integrated circuit drivers that were never tested before for automotive usage or where not defined in the requirements in a consistent way.

3. The connectors and terminals used for the electronic control module shall feature rigid tolerances that prevent: Insertion/exertion problems, such as loose terminals or other environmental problems such as water intrusion, corrosion, dendrite growth, etc. A common practice of cost reduction initiatives, decrease the robustness of the interconnections and cause hidden systemic faults that later cause catastrophic effects to systems.

   Motivation: The systemic problems such as: Durability, System Performance, Interface device faults, documented on section 3.2.4, (NHTSA, 2014), were arguably the root cause of vehicle faults reported by customers.

4. The trends in the electronic industry influence the introduction of smaller and faster microprocessors, this means smaller geometries and faster response times that affect the EMC immunity. The printed circuit board designs used in electronic control modules has to be revised when introducing small scale chips.

   Motivation: The systemic problems related to Electromagnetic Compatibility due to the introduction of faster microprocessors and power drivers, which are packed in chips (integrated circuits) featuring higher density that goes beyond the VLSI technologies, imply scale in the nanometers and time in the range of the giga-hertz. According to (Beeker, 2014), the Electromagnetic field behavior can cause wave reflection and additional phenomenon that affects the components on the PCB that in some cases could suffer damage from ESD.

5.2 Guidelines for System Functions in automobile systems

1. The software functional structure for an electronic control module shall feature specialization by functional domains, and the system characteristics and functional requirements shall be based on realistic performance attributes.

   Motivation: The systemic problems related to Algorithm Faults, documented in section 3.2.4, (NHTSA, 2014), are often caused by design flaws due to the models are designed with ideal scenarios which did not reflect real life conditions.

2. The engineers involved in the design of an ECU shall be educated to learn how the electronic systems perform in an automobile subsystem and must have in mind that an automobile is a collection of sub-system domains and not individual components that operate on their own. The engineers have to learn this by participating in vehicle performance validation, manufacturing troubleshooting or any other opportunity to grasp the functionality details.

   Motivation: The knowledge limitations about the vehicle performance and the role of a component in a system can be addressed when the model designer has been exposed to field experience. This helps to consider all the factors that affect a given process, so modeling of real life scenarios come easily compared to when the person has no idea about this aspects.

3. The software designed for an electronic control module shall feature interruption management as a critical building block for embedded software.

   Motivation: Embedded software has been a common practice by OEM's during the last 10 years. This method considers that a microprocessor could run several processes in parallel and not face time delays. However, based on experiences documented in section 3.2.4, in several cases, ECU's featuring embedded code resulted in systemic faults due to wrong assumptions, such as the one that asynchronous processes will be running in parallel with no delay or effect on other processes. In practice, microprocessor interruptions affected variables causing algorithm malfunction.

4. Do not assume that the code could be 100% "carry-over" to use in new vehicle architecture(s).

   Motivation: The use of "carryover" components in the automotive industry is very common practice, and when the stakeholders stress the use of such practices, the engineers often re-use models with inherited errors. When a part is used in a new architecture, the environment to which the carry over part is going to be used is not the same as the one it was designed for, thus it will likely under-perform.

5. Functions related to engine start (engine crank) or engine stop (engine off) could cause a state that triggers malfunctions during vehicle initialization or shutdown.

   Motivation: Prior experiences of the author while performing troubleshooting activities in several vehicle lines resulted in a project called Portable USB power mode simulator tool [28] (Quezada, Kirchhoff, & De Stefano, 2012), to which the lessons learned concluded that a state or condition that at times can be linked as a potential trigger event for certain vehicular electrical system failure modes are often highly intermittent and difficult to isolate and diagnose.

6. The system response to power fluctuations of the battery feed power shall be considered in the algorithm to avoid entering to a software lock-up.

   Motivation: The engine cranking in an automobile can cause sudden battery voltage fluctuations or transients that affect the system functional performance, creating problems that in several cases cause the software interruptions to enter into unexpected states that were perceived as a software lockup. Several problems experienced by the author, also documented in section 3.2.4, relate to System Performance reports.

7. Customer dynamic conditions during normal and extreme vehicle conditions shall be considered in the algorithms to get a more robust design.

   Motivation: The design often assumes that the customer will never reach certain conditions that put the system outside of the operational range. In some of the problems documented in section 3.2.4, (NHTSA, 2014), the System Capability or boundaries were limited, causing an apparent vehicles under-performance (engine stall, ignition switch troubles, loss of system power, etc.)

[28] Portable USB power mode simulator tool Patent: US 8150671 B2, can be found at http://www.google.com/patents/US8150671

CHAPTER 6 CONCLUSIONS AND AREAS FOR FURTHER STUDY

This research provides guidelines for Automotive Electronic Systems Software Development. At its core is a detailed OPM conceptual model of the system to-be. The model and guidelines provided here will enable electronic systems designers, engineers, stakeholders to better understand the system, create new features, and help software developers to see the system components in a new way thanks to the OPM graphical advantages. The OPM model along with the guidelines are expected to help address the problems identified in the various stages of the automobile design lifecycle.

A considerable upfront investment of design time and effort is required to create the system level requirements that are usually provided as a waterfall to engineering organizations, suppliers, etc. System architects must ensure that all the base functions used in the system processes, models and software code are validated in order to provide robust designs.

Future Research

Further studies related to this research include the following:

1. Complete a more detailed and testable model to demonstrate and verify the effectiveness of the proposed model-based systems engineering approach to the design of vehicle body electronics.

2. Develop a tool in OPM that manages a project lifecycle and allows management of the system requirements for the duration of the project.

3. Develop a tool that generates the test cases from the OPM functional models.

Bibliography

Allen, T. (2007). *The Organization and Architecture of Innovation Managing the Flow of Technology.* Burlington, MA, U.S.: Elsevier Inc.

Beeker, D. (2014). *Freescale semiconductor.* Retrieved from Freescale Semiconductor website: http://www.freescale.com/files/training/doc/dwf/AMF_AUT_T0750.pdf

Berger, M. L. (2001). *The Automobile in American History and Culture: A Reference Guide.* Westport, CT: Greenwood Press.

Daimler AG. (2014). *Karl Benz.* Retrieved July 31, 2014, from Daimler website: http://www.daimler.com/dccom/0-5-1333261-49-1279445-1-0-0-0-0-1-36-7145-0-0-0-0-0-0-0.html

Davidz, H. L. (2006, September 12). *2006_DISSERTATION_n-opsswd_Sept_12_2006.ppt.* Retrieved Oct 15, 2014, from INCOSE Website: http://www.incose.org/wma/library/docs/INCOSE_2006_DISSERTATION_n-opsswd_Sept_12_2006.ppt

Davison, W. (1840). *Horse and Carriage.* Retrieved March 1st, 2013, from From Old Books website: http://www.fromoldbooks.org/Davison-Ornaments/pages/0162-horse-and-carriage/

De Weck, O., & Lynesis, J. (2013). Lecture 6 Introduction to Project Dynamics. *SDM Project Management Lecture 6*, 50.

De Weck, O., & Lynesis, J. (2013). Lecture 7 The re-work Cycle. *SDM Project Management Lecture 7*, 17-45.

De Weck, O., & Lynesis, J. (2013). *Successfully Designing and Managing Complex Projects.* Cambridge, MA: MIT Press, First Edition - Draft.

De Weck, O., Lynesis, J., & Moser, B. (2013). Lecture 1:"Introduction to Project Management". *SDM Project Management Class Fall 2013.* Cambridge, MA, U.S.: MIT ESD.

Dearborn Group Technologies. (2001, April 30). Introduction to In-vehicle Networking. *Seminar: CAN Workshop* (pp. 14,17,18,19,35). Dearborn: Dearborn Group Technologies.

Dell, R. (2001). *Understanding Batteries.* Cambridge, UK: Royal Society of Chemistry.

Dori, D. (2002). *Object-Process Methodology -- A Holistic System Paradigm.* Berlin, Heidelberg, New York: Springer.

Dori, D. (2002). *Object-process methodology: A holistic systems paradigm.* Berlin: Springer Verlag.

Dori, D. (2014). Lecture #2 MBSE Introduction. *MIT SDM ESD.S40 Model Based Systems Engineering*, 33.

Embley, D., & Thalheim, B. (2011). *Handbook of Conceptual Modeling.* Berlin: Springer Berlin Heidelberg.

Energy & Commerce Committee, U. S. (2014, July). *The GM Ignition Switch Recall: Investigation Update.* Retrieved July 15, 2014, from Energy and Commerce Comitee: http://energycommerce.house.gov/hearing/the-gm-ignition-switch-recall-investigation-update

Estefan, J. A. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies. *INCOSE MBSE Initiative*, 43.

Forrester, J. (2010, December 31). *System Dynamics: the Foundation Under Systems Thinking.* Retrieved October 15, 2014, from CLEXCHANGE website: http://clexchange.org/ftp/documents/system-dynamics/SD2011-01SDFoundationunderST.pdf

Forsberg, K., & Mooz, H. (1992). The Relationship of Systems Engineering to the Project Cycle. *Engineering Management Journal, 4, No. 3*, pp. 36-43.

General Electric. (2014). *What is Six Sigma ?* Retrieved Oct 15, 2014, from http://www.ge.com/: http://www.ge.com/en/company/companyinfo/quality/whatis.htm

General Motors Corporation. (2005). *Business Systems and Business Standards Discussions and Information APQP Process.* Retrieved October 15, 2014, from Elsmar website: http://elsmar.com/pdf_files/Quality%20and%20Other%20Manuals/GM%20APQP%20Manual%20-%20GM1927%20-%20Revision%201.doc

General Motors Corporation. (2010, May). *GM's Road to Virtual Product Development.* Retrieved from GM Heritage Center website: http://history.gmheritagecenter.com/wiki/index.php/GM's_Road_to_Virtual_Product_Development

General Motors Corporation. (2010). *Project TRILBY.* Retrieved from GM Heritage Center website: https://history.gmheritagecenter.com/wiki/index.php/Project_TRILBY

Hellestrand, G. (2014, July). *ESL Development Gets A Leg Up.* (Chip Design Magazine) Retrieved July 31, 2014, from Chip Design Magazine website: http://chipdesignmag.com/display.php?articleId=57

Huges, P. A. (1996). *History of the electric car 1828-1912 from Trouve to Morrison.* Retrieved March 1, 2013, from Web Archive website: http://web.archive.org/web/20111113023143/http://factoidz.com/history-of-the-electric-car-1828-1912-from-trouve-to-morrison/

INCOSE. (2006, October 2). *Systems Engineering Handbook.* Retrieved July 31, 2014, from INCOSE website: http://www.incose.org/practice/fellowsconsensus.aspx

INCOSE. (2007, September). *SYSTEMS ENGINEERING VISION 2020.* Retrieved Nov 28, 2014, from INCOSE website: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf

ISO-14229. (2014). *ISO14229 :1:2006 Road vehicles — Unified diagnostic services.* Retrieved from International Standards Organization (ISO) website: https://www.iso.org/obp/ui/#iso:std:iso:14229:-5:ed-1:v1:en

Leveson, N. (2011). *Engineering a Safer World Systems Thinking Applied to Safety* (Vols. ISBN 978-0-262-01662-9). Cambridge, MA, U.S.: MIT Press, Cambridge, Massachusetts.

Levine, M. (2011, May). *Inside GM's State-of-the-Art Powertrain Engineering Center.* Retrieved from Pick-up Trucks.com website: http://news.pickuptrucks.com/2011/05/inside-gms-state-of-the-art-powertrain-engineering-center-.html

Macias Anaya, N. (1999). *Thesis "Engineering Design Lead Team Drivers Analysis".* Cambridge: MIT Sloan School of Management .

MacMahon, D. (2009). *Some EV history.* Retrieved March 1, 2013, from Econogics website: http://www.econogics.com/ev/evhistry.htm

NHTSA. (2014, 11 30). *FLAT FILE COPIES OF NHTSA/ODI DATABASES.* Retrieved from US Department of Transportation: http://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm

PBS. (2010). *Electric car timeline.* Retrieved July 31, 2014, from PBS website: http://www.pbs.org/now/shows/223/electric-car-timeline.html

Quezada, J., Kirchhoff, R., & De Stefano, R. (2012). *Patent No. US 8150671 B2.* United States of America. Retrieved from http://www.google.com/patents/US8150671

Razza Rahil, H. (2014). *Full Form of Programming Language.* Retrieved Oct 15, 2014, from Razzil website: http://www.razzil.com/images/programming_language.jpg

Reynolds, M., & Holwell, S. (2010). *Systems Approaches to Managing Change: A Practical Guide* (Vols. DOI 10.1007/978-1-84882-809-4_2). The Open University 2010 in Association with Springer-Verlag London Limited.

Rince, J.-C. (2012, October). *Freescale Automotive Teaching Lab INSA Tolouse.* (Freescale) Retrieved July 31, 2014, from INSA website: https://etud.insa-toulouse.fr/~ssahin/5ESE-BE%20Automobile/Supports%20de%20cours/INSA_FSL_Automotive_Lab_Class_2012-2013_2.pdf

Rozanski, N., & Woods, E. (2005). *Stakeholders in Software Systems Architecture.* Retrieved Oct 15, 2014, from Viewpoints and Perspectives website: http://www.viewpoints-and-perspectives.info/home/stakeholders/

SAE. (1993). *SAE J-1739 Potential Failure Mode and Effect Analysis (FEMA) Reference Manual.* SAE.

Siemens PLM. (2014). *Unigraphics in the 80's 1986-1989.* Retrieved from PLM WORLD the voice of Siemens PLM software users website: from: http://www.plmworld.org/media/llqldpqz.jpg

Six sigma consulting group. (2014, October). *About six sigma.* Retrieved October 15, 2014, from 6 sigma website: http://www.6sigma.us/six-sigma.php

Teske, L. (2007). Virtual Vehicle Development Process at GM. *1st Hyper works Technology Conference* (p. 3 to 13). Berlin, Germany: Altair Hyperworks, UK. Retrieved from http://www.altairhyperworks.co.uk/html/en-gb/keynote2/teske_gm.pdf

The United States Library of Congress. (2014, August 11). *Topics in Chronicling America - Electric Cars in America (1891-1922).* Retrieved July 31, 2014, from United States Library of congress website: http://www.loc.gov/rr/news/topics/electricCars.html

Towsend, J., Cavusgil, T., & Baba, M. (2009). *Global Integration of Brands and New Product Development at General Motors.* Retrieved October 15, 2014, from Research Gate website: http://www.researchgate.net/publication/227841945_Global_Integration_of_Brands_and_New_Product_Development_at_General_Motors

U.S. Department of Transportation. (2014, March). Transportation Systems Safety Hazard Analysis Tool. *Transportation Systems Safety HAT.* Cambridge, MA, USA: John A. Volp National Transportation Systems Center.

University of Groningen Netherlands. (2013, May). *Sibrandus Stratingh (1785-1841) Professor of Chemistry and Technology.* Retrieved July 31, 2014, from University of Groningen website: http://www.rug.nl/science-and-society/university-museum/prominent-professors/stratingh

Valukas, A. R. (2014, May 29). *Read the Valukas report on GM's ignition recall.* Retrieved Jul 15, 2014, from Detroit News website: http://www.detroitnews.com/article/20140605/SPECIAL01/140605001

Wikipedia. (2005). *File:Henry_Ford_-_Quadricycle,_1905.jpg#filehistory.* Retrieved March 1, 2013, from Wikipedia website: http://en.wikipedia.org/wiki/File:Henry_Ford_-_Quadricycle,_1905.jpg#filehistory

Wikipedia. (2014, November). */wiki/On-board_diagnostics.* Retrieved November 28, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/On-board_diagnostics

Wikipedia. (2014). *wiki/CAN_bus.* Retrieved August 15, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/CAN_bus

Wikipedia. (2014, November 28). *wiki/Henry_Ford.* Retrieved November 28, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Henry_Ford

Wikipedia. (2014). *wiki/History_of_the_electric_vehicle.* Retrieved July 31, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/History_of_the_electric_vehicle

Wikipedia. (2014). *wiki/Network_topology.* Retrieved July 31, 2014, from Wikipedia website: http://en.wikipedia.org/wiki/Network_topology

Appendix A Pictures of the GM office layout in the year 1956 [29]

The pictures shown below present the GM office layout in 1956.

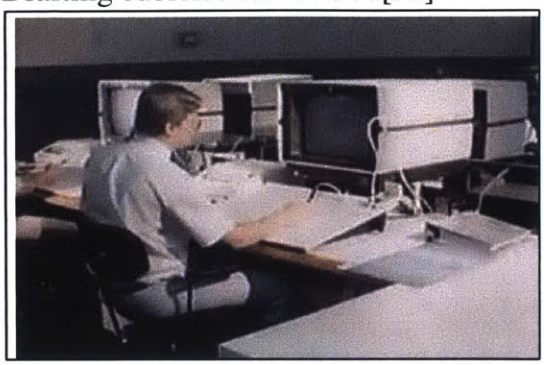
The office of engineers & drafting designers

One of the first computers – an analog computer used at Milford Proving Grounds in 1949

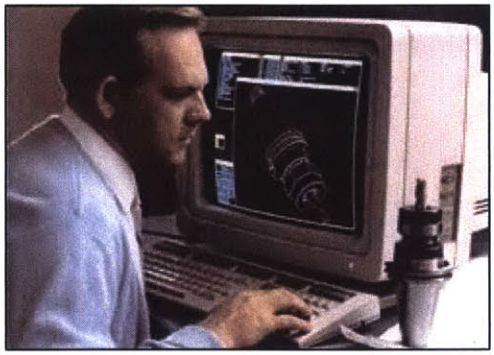
[29] Pictures retrieved March 2014 from: (General Motors Corporation, 2010)

Appendix B Pictures of the office layout in the 70's and 80's

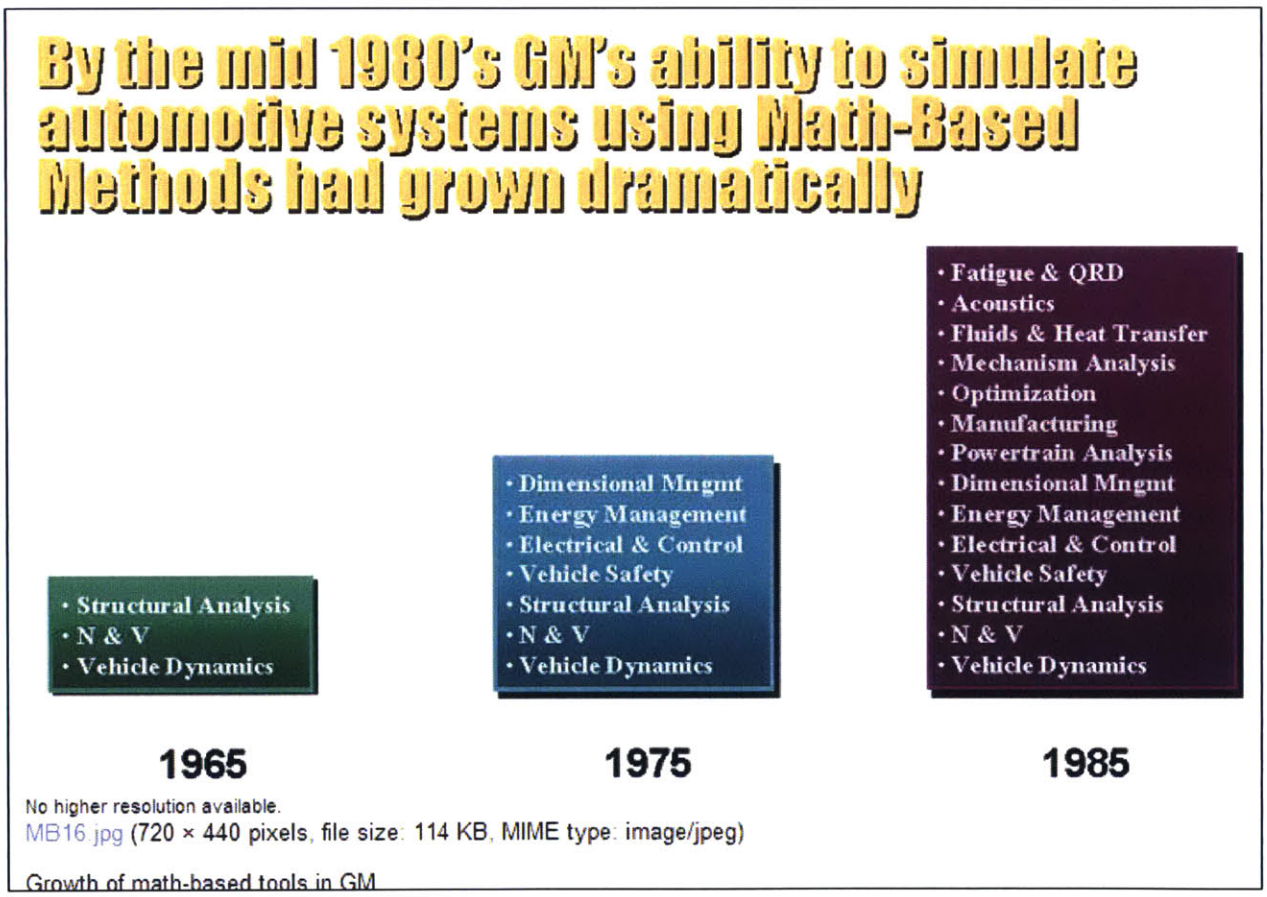
Drafting cubicles for in 1971[30]



The design engineer cubicle's in 1986 [31]



The growth of math based simulation methods in GM by 1985 [32]



[30] Picture retrieved March 2014 from: (General Motors Corporation, 2010)
[31] Picture retrieved March 2014 from (Siemens PLM, 2014)
[32] Image retrieved March 2014 from: (General Motors Corporation, 2010)

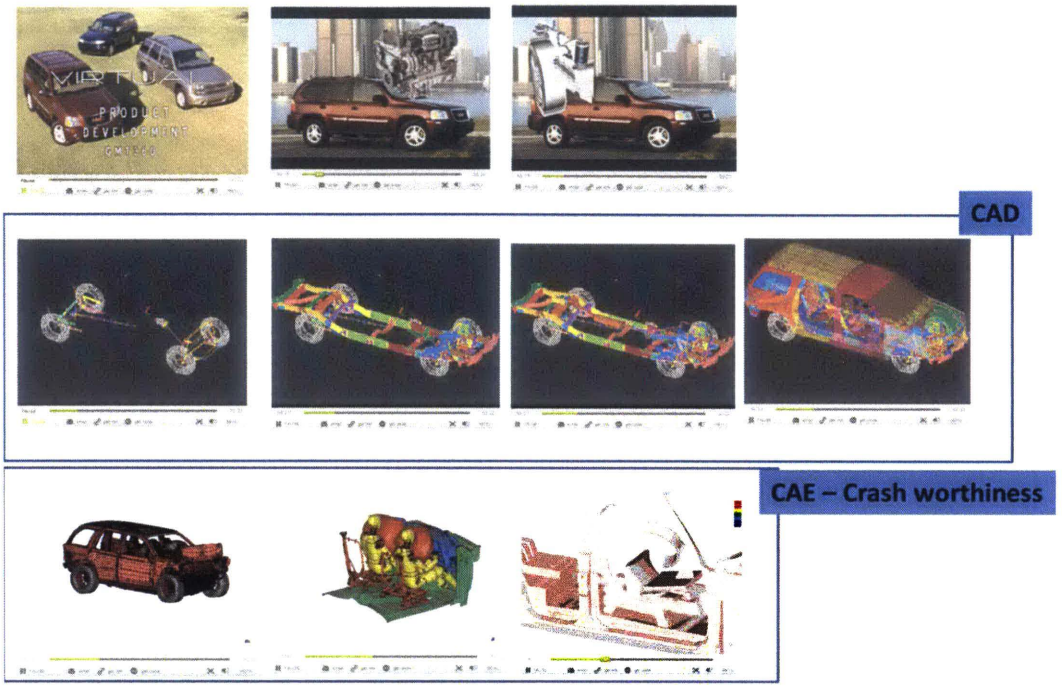
Appendix C Picture from GM Powertrain Engineering Development Center in Pontiac, MI [33]

"Computer aided design was soon supplemented with computer simulation, which tested parts performance virtually before a physical copy was ever fabricated and tested in the real world. Modeling and simulation became more sophisticated as the lessons learned from the previous generation of car or truck were applied to the next generation" (Levine, 2011)

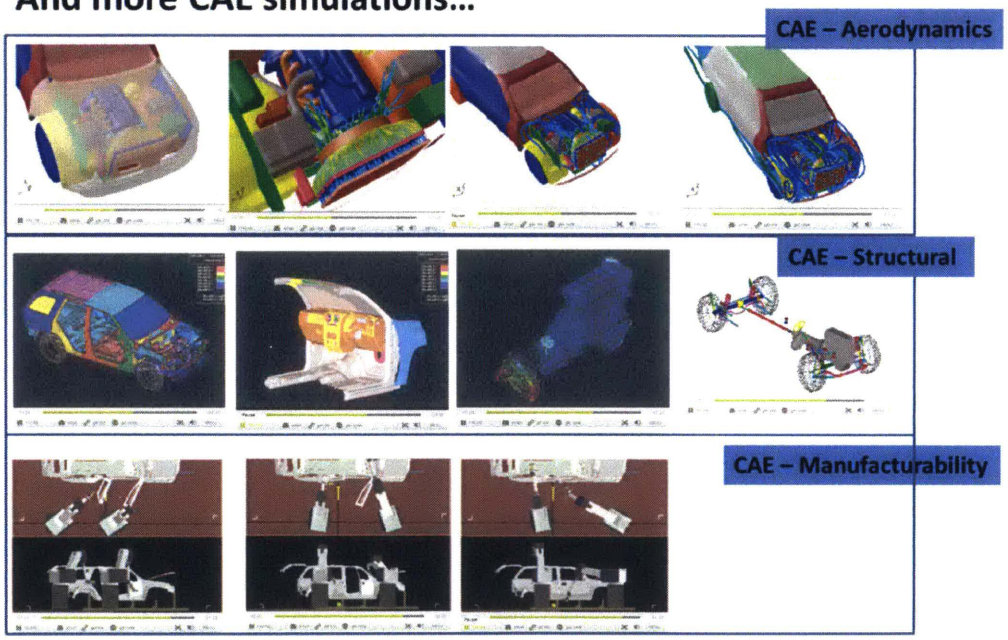
[33] Picture retrieved on March 2014 from: (Levine, 2011)

Appendix D The CAD capabilities of Design GM in the 1990's [34]

The CAD capability enhanced the product development



And more CAE simulations...



[34] Picture retrieved March 2014 from: (General Motors Corporation, 2010)

Appendix E The virtual Math Modeling Design GM [35]

Heritage CENTER Generations of GM History

- Home
- Main Page

Search

Go

Categories

- Brands & Products
- The Business
- Design
- Diversity
- Environment & Energy
- Eras
- Firsts
- Former Divisions
- Innovation & Technology
- People
- Places
- Racing
- Shows & Events
- I was there...

Page

Project TRILBY

I was there

With the advent and proliferation of electronics and computers in automotive vehicles in the early '80's, a Corporate project named TRILBY was started under the auspices of GM Research (GMR) to investigate how this could be done more rationally. More specifically, in the TRILBY's "Charter Letter" of March 23, 1984, Bob Frosch, then VP of GMR, stated TRILBY's mission as "to create the methodology and technology for designing cars from an overall systems point of view, marrying our best knowledge of automotive science and technology to modern systems, controls, computer and electronics technology. The emphasis is on control of all aspects of vehicle operation....TRILBY is a character in an 1894 novel of the same name by George du Maurier. In the novel she is under the control of Dr. Svengali." I was part of the team that launched TRILBY, and was subsequently responsible for developing various math-based tools and methods to support this novel (at that time) systems approach to automotive vehicle creation.

In 1984, a small team of us had moved from GMR to the Top of Troy Building to formulate TRILBY, and subsequently to our permanent headquarters at 1151 Crooks Road in Troy. We recruited a "volunteer army" from within GMR, and subsequently several engineers and managers were added from other operating units. TRILBY, at its peak, had over 100 engineers and scientists on staff.

Media

Cute logo used by Project TRILBY. The hat is a Trilby and it symbolizes "top to bottom" or "head to foot" control of vehicles

I was there...

Tell us your story »

Hazardous Road

GM's Smith Presses For Sweeping Changes But Questions Arise

He Seeks to Give Giant Firm Entrepreneurial Flavor; Will He Give Up Control?

Following Sloan's Footsteps

By Melinda Grenier Guiles
Staff Reporter of The Wall Street Journal

"I've heard people say, 'Oh, Mr. Sloan would be spinning in his grave if he knew what you were doing.' I say that's nuts. He'd be doing exactly what I'm doing. I really believe that."
—General Motors Corp. Chairman Roger B. Smith

DETROIT—Can cars be engineered and designed in a completely different way, unlike any contemplated today? General Motors Corp. plans to have 100 people studying the question in a supersecret project code-named Trilby.

The original TRILBY team, while formulating the project realized that the only rational way to effectively integrate electronics, computers, and controls into vehicles was via a "top-down" or systems engineering approach as was subsequently reflected in Bob Frosch's charter letter (above). We travelled around the country looking for partners for TRILBY. In fact, GM's acquisition of Hughes was partially driven by upper management's support of TRILBY – and, in particular, Roger Smith's support. We had just gotten going on TRILBY when Roger mentioned us to the *Wall Street Journal* and we ended up on the front page on March 14, 1985 (left sidebar)! The TRILBY team defined probably the first systems engineering (albeit not totally complete) process for GM, developed a number of new vehicle control concepts and demonstration vehicles including integrated chassis control and 4-wheel steering concepts, and developed a number of state-of-the-art facilities including a hardware in the loop (HIL) laboratory. We also defined how human factors considerations could be better integrated into the vehicle creation process, and assembled an extremely capable human factors team. TRILBY technical people developed many computer-aided engineering and control system design tools some of which are still being used today.

Although originally touted as a 5 year program, the Corporate "appetite" for systems engineering and its tools increased markedly. Hence, in 1988 Project TRILBY ended. A large number of the technical people formed the nucleus of GM's Systems Engineering Center (a CPC organization at that time). The remainder moved back to GMR where they continued to contribute significantly to advanced vehicle and control concepts over the years.

Project TRILBY was one of the greatest experiences of my 32 year career with GM!

Trilby Vehicle Design System

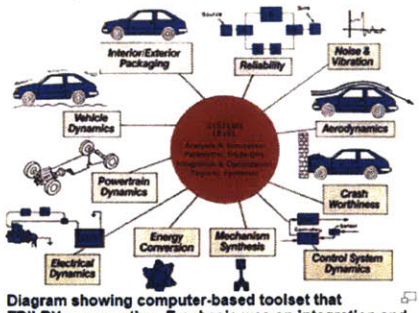
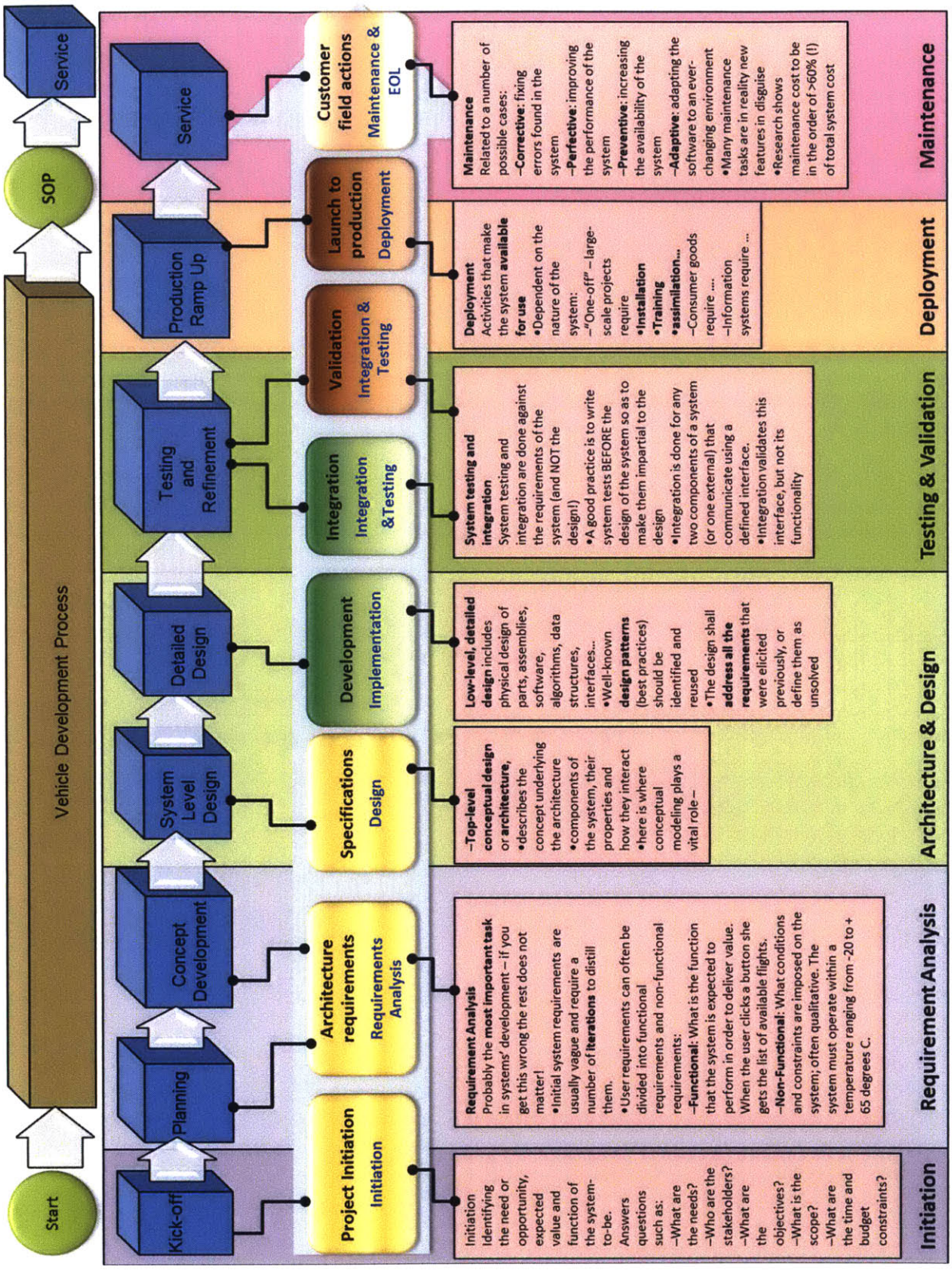


Diagram showing computer-based toolset that TRILBY was creating. Emphasis was on integration and having a coarse to fine toolset. Also, TRILBY added reliability, flexible tools, and robust design capabilities to GM's capabilities.

[35] Picture retrieved March 2014 from: (General Motors Corporation, 2010)

Appendix F The lifecycle stages in the automobile domain [36]



[36] This illustration was constructed based on the MBSE class Lecture 2 (Dori, Lecture #2 MBSE Introduction, 2014) and the Vehicle Development Process discussed on section 2.4.