Model Based Safety Analysis of Cyber Physical Systems

by

Sailesh Umamaheswara Kandula

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved December 2010 by the
Graduate Supervisory Committee:

Sandeep Gupta, Chair
Yann Hang Lee
Georgios Fainkeos

ARIZONA STATE UNIVERSITY

December 2010

ABSTRACT

Cyber Physical Systems (CPSs) are systems comprising of computational systems that interact with the physical world to perform sensing, communication, computation and actuation. Common examples of these systems include Body Area Networks (BANs), Autonomous Vehicles (AVs), Power Distribution Systems etc. The close coupling between cyber and physical worlds in a CPS manifests in two types of interactions between computing systems and the physical world: intentional and unintentional. Unintentional interactions result from the physical characteristics of the computing systems and often cause harm to the physical world, if the computing nodes are close to each other, these interactions may overlap thereby increasing the chances of causing a Safety hazard. Similarly, due to mobile nature of computing nodes in a CPS planned and unplanned interactions with the physical world occur. These interactions represent the behavior of a computing node while it is following a planned path and during faulty operations. Both of these interactions change over time due to the dynamics (motion) of the computing node and may overlap thereby causing harm to the physical world. Lack of proper modeling and analysis frameworks for these systems causes system designers to use ad-hoc techniques thereby further increasing their design and development time. The thesis addresses these problems by taking a holistic approach to model Computational, Physical and Cyber Physical Interactions (CPIs) aspects of a CPS and proposes modeling constructs for them. These constructs are analyzed using a safety analysis algorithm developed as part of the thesis. The algorithm computes the intersection of CPIs for both mobile as well as static computing nodes and determines the safety of the physical system. A framework is developed by extending AADL to support these modeling constructs; the safety analysis algorithm is implemented as OSATE plug-in. The applicability of the proposed approach is demonstrated by considering the safety of human tissue during the operations of BAN, and the safety of passengers traveling in an Autonomous Vehicle.

ACKNOWLEDGEMENTS

I would like to thank Dr. Sandeep Gupta for his constant motivation and guidance without which this work would not have been possible. I'm also thankful to my committee members Dr Georgios Fainekos and Dr. Yann Hang Lee for their being part of my committee and giving valuable feedback. I'm indebted to Tridib Mukherjee and Ayan Banerjee who were always there to guide me, working with them was a great learning experience. Last but not the least, i would like to thank my parents who have been a constant source of support and inspiration to me. I thank the National Science Foundation (through grants CNS-0855277) for funding this research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Cyber Physical Systems (CPSs) are system of systems in which computing nodes are embedded in the physical world and perform various task such as monitoring, control, computation, and communication. These systems have the potential to improve socio economic standards of living by addressing some of the most significant problems faced by our society such as reducing traffic congestion and road accidents, improving health care, green energy technologies and much more.

Cyber and physical sub systems of a CPS have been studied individually in past but such a view does not allow system designers to capture the two way interactions, a.k.a Cyber Physical Interactions (CPIs) that occur between them. As computing systems of the future become more complex and are deeply embedded in the physical world, a holistic view is required to properly analyze them. Further, Computing nodes in a CPS can be either static or mobile. This classification is important because CPIs of mobile computing nodes have additional characteristics than static computing nodes. For static computing nodes, there are two types of interactions with the physical world: 1) intended interactions : interactions that are initiated by computing nodes to accomplish a given task such as sensing, and communication, 2) unintended interactions : interactions that are not explicitly initiated by computing system but are the result of its physical nature and computing operations, e.g. tissue temperature rise due to heat dissipation by computation of sensors implanted in human body. Both intended and unintended interactions are spread over regions on the physical world and vary over time. For example, if the computing system has low power and high power states, then the amount of heat generated is different in both cases. Unintended interactions are often harmful to the physical world and can be viewed as side-effects of computing operations. If these interactions are not properly analyzed at design time they can cause safety hazards to the physical world. In addition, due to proximity of computing nodes unintended interactions can overlap and the resulting side effects can add up thereby increasing the chances of causing a safety hazard. For example, if two or more body sensors are placed close to each other, then the temperature rise will be faster as compared to the situation when there is only one sensor. Safety hazard is defined

as a situation in which one of the parameters of the physical system changes significantly and rises above a predefined threshold. In the previous example, if the temperature of the tissue rises to more than $39.2\,^{\circ}\mathrm{C}$, then there is a high probability of burn injury to the person.

CPSs in which computing nodes have mobility are referred as Mobile CPSs (MCPSs), e.g. Autonomous Vehicles (AVs). MCPSs in addition to intended and unintended interactions have two additional types of interactions, i) planned interactions: interactions with the physical world while a computing system is following a pre-planned trajectory. e.g:- an autonomous vehicle traveling along the planned path generated by itŠs navigation system . ii) unplanned interactions : interactions that result from incorrect operations of computing nodes, e.g. skidding of an autonomous vehicle along a curved road due to high speed. If planned or unplanned interactions of two or more mobile computing nodes overlap they can have detrimental effect on the physical world. For example, collision between two AVs may cause severe injury to human occupants traveling in them. Unplanned interactions of an MCPS occur only when certain properties of the computing node go above a threshold. For example, an AV skids along a curved road if it's velocity is above a certain threshold which again depends on the road conditions such as curvature of road and friction.Both unplanned and planned interactions of a computing node change over time, thus interactions which did not pose a threat to the physical world at a given point of time may overlap in future leading to a safety hazard.

1.1 Motivation

Safety of Cyber Physical Systems can be viewed from two perspectives, Operational safety and Interaction safety. Operational safety is defined as the safety of physical infrastructure or humans in spite of critical events in the environment. It can be achieved through fault tolerance mechanism [16], criticality management [35] and several other techniques. Interaction safety on the other hand is defined as the safety of the physical world in spite of unintended or unplanned interactions and is the main focus of this thesis.

To analyze the interaction safety of the physical world it's essential to model different aspects of CPIs at design time. This modeling can be done in primarily two ways : i) Formal modeling, and ii) Architectural modeling. Formal modeling is a technique where system designers use formal methods and techniques such as hybrid automata and reach-

ability analysis to analyze the safety of the system. This thesis seeks to complement such verification by facilitating the modeling and analysis from a system architectural perspective, i.e. allowing for safety analysis (at design time) from representation of system components, properties, and their inter-dependencies.

Architectural modeling of CPS has focused on modeling different software and hardware components along with their interactions [19]. Safety verification has also been considered for reliable, or error-free, operation of the software and hardware components [2].However, safety vulnerabilities can also be caused by conditions of the physical environment, thus it is important to capture the inter-dependencies of the cyber (i.e. software and hardware) components with the physical environment in a holistic manner.

1.2 Objective

The objective of the thesis is to **develop modeling abstractions that can capture the semantics of CPS and a safety analysis algorithm that uses them to the determine safety of the physical world**.

1.3 Challenges

The objective posses several challenges that needs to be addressed and are presented below:

1. The manifestation of CPIs varies across domains. For example. CPIs of autonomous vehicles significantly differ from CPIs of BSNs. The modeling abstractions that represent these CPIs should be generic enough to handle their diverse nature.

2. The safety analysis algorithm should be able to handle the dynamic nature of CPIs and aggregate side effect of the CPIs on the physical world.

1.4 Contributions

Main contributions of the thesis are summarized below:

1. Modeling abstractions of CPS which capture semantics of intended, unintended, planned and unplanned interactions between computing and physical systems.

2. Safety analysis algorithm that computes the intersection of spatial regions representing CPIs and determines potential harm to the physical world in O(n^2), n is the number of computing nodes.

3. AADL cps annex that implements the modeling constructs as an extension to AADL language.

4. Two Case Studies demonstrating the application of CPS modeling constructs and safety verification logic. These are:

    a) verifying the safety of human tissue due to heat dissipation of sensors implanted on human body using cps constructs and safety analysis algorithm.

    b) verifying the safety of passengers in an AV using CPS constructs and safety analysis algorithm.

1.5 Solution Approach

To achieve the above mentioned contributions a model based approach for analyzing the safety of the physical system is considered. Four modeling abstractions such as *Regions of Interest(ROIn), Region of Impact(ROIm), Intended Region of Mobility(IROm), Unintended region of mobility (UIROm)* are proposed to model CPIs. These constructs have sub-constructs that describe spatial region of CPIs for static computing nodes as well as mobility behavior of mobile computing nodes. Physical laws describing the dynamic nature of CPIs in a CPS, their behavior and side-effect on the physical world are also specified as a sub-construct. Safety of the physical system can be specified as thresholds on the physical system parameters. For example, in case of BAN, the temperature of the human skin should be less than $39.2\,^{\circ}\mathrm{C}$. A safety analysis algorithm is developed that computes spatial regions of CPIs, their intersection and uses equations specified as physical laws to determine if the safety condition of the physical system is violated or not.

Modeling abstractions are implemented as an annex by extending AADL [2], an industry standard language used for modeling embedded systems. To integrate with the AADL framework annex grammar, parser and semantic checker are developed .

Two case studies are being considered to show the applicability of the proposed constructs and safety analysis algorithm:

1. In the first case study, thermal side effects of communication of a network of body sensors and computation of a single pulse oximeter on the human tissue is considered. Thermal side effects are modeled by ROIms construct. The safety analysis algorithm first computes the temperature rise in each individual ROIm to determine the safety of the tissue. It then computes the intersection of multiple ROIms and to determine the aggregate side effect.

2. In the second case study, safety of passengers traveling in a pickup truck (AV) on Mile Post 44, Arizona 83 highway is determined. The identified road segment is a horizontal curve and the AV can potentially skid and collide with the guard rail. A recent report [14] published by National Cooperative Highway Research Program (NCHRP) identifies that the average crash rate along horizontal curves is three times more than any other horizontal road segment. The safety analysis algorithm computes the probability of serious injury to passengers using safety verification algorithm. The same metric is computed based on the number of accidents and other factors identified in the AZ-83 assessment report [41]. Comparison of the safety verification results with the results based on the AZ-83 assessment report shows an error in serious injury probability to be around 0.001.

Chapter 2

RELATED WORK

In this chapter, the thesis is compared with other relevant works from the literature. The related work is categorized into following areas:

2.1 Model Based Analysis of Embedded Systems

Embedded systems are often considered as the precursors of Cyber Physical Systems, since they have a computation aspect and interact with the physical world by means of sensors and actuators. Researchers have proposed different modeling tools such as AADL [2], Simulink [11], MARTE [13] and techniques [34] [20] [38] to analyze the safety of embedded systems systems. None of these tools support constructs to model Cyber physical Interactions. Model transformation is another commonly used technique to convert models from one modeling language to another, so that it's easily to analyze or there is a good tool support. In [20] authors propose a model transformation technique to verify the active safety systems of automobiles. The technique converts automobile design model specified using EAST-ADL2 to HIP-HoPS, a safety analysis tool. EAST-ADL2 constructs model physical world to a certain extent but they are specific to automobile domain and cannot be used to model generic CPIs. In [38] a dependability modeling framework that uses AADL Error Annex to model the error states and the fault-tolerance mechanism of a system is proposed. The framework can be used to analyze the propagation of errors between different subsystems thereby determining if the system is safe or not. Authors in [36] propose ANDES, a tool that uses Model based analysis techniques to analyze low latency and accuracy of Wireless Sensor Network operations. Both the above works, do not model physical world and the CPIs. In [44] authors develop a dynamic risk assessment strategy to assure the safety of autonomous vehicles. The strategy allows controller of an autonomous vehicle to assess the risk associate with each possible control action at a given point of time and determine the safest action to carry. The work was intended towards designing safe controllers, however the present thesis can be used on evaluating whether a given controller is safe by determining if it's control outputs cause harm to the physical world, as demonstrated in the case study involving autonomous vehicles.

2.2 Model Based Development of Cyber Physical Systems

Some of the tools mentioned in the previous section can be used to model computing components of a CPS. To model the physical world tools such as Modelica [7], Fluent [4] and Sysml [12] can be used. . In [39] authors propose architectural patterns for CPS development that can be formally verified. The focus of the work was on building formally verifiable components. This thesis is focused on verifying if the system in safe or unsafe.

2.3 Architecture modeling of Cyber Physical Systems

Architecture modeling of Cyber Physical Systems have been done by researchers in [37] and [31]. Their main contribution was on modeling intended cyber physical interactions such as sensing and actuation. For example, in [37] authors propose P2C and C2P connectors between cyber and physical subsystems of a Cyber Physical System. This thesis considers intended and unintended interactions of static computing nodes as well as planned and unplanned interactions of mobile computing nodes. In [8] authors perform a formal analysis of a system's architecture model (specified using AADL) by specifying the behavior of the system using Algebra of Communicating shared resources [8]. But, the authors don't provide any abstractions for modeling interactions between cyber and physical sub-systems of a CPS. This work on the other hand doesn't provide constructs for formally specifying the system behavior but provides abstractions for modeling cyber physical interactions.

2.4 Hybrid Automata and Formal Verification

Hybrid automata and formal verification techniques have been used by researchers in [29] to ensure the safety of autonomous robots. In a hybrid automata, certain states are designated as unsafe and reachability to these states are analyzed for safety verification. Authors in [15] use formal verification techniques to guarantee the safety of Autonomous Vehicles under sensor uncertainties or other disturbances. The safety criteria considered in that work also does not take into account the probability on passenger injury. This paper is complimentary to such verification that allows modeling abstractions from the system architectural perspective, provides specific constructs for semantically different planned and unplanned

interactions, and enables passenger safety verification for complex scenarios at the design time.

2.5 Operational Safety of Cyber Physical Systems

Operational safety for CPS has been considered by authors in [35]. The paper presents a framework for modeling the critical states and critical events in the system. Different metrics are proposed that evaluate the effectiveness of mitigative actions so that system can return to normal state. The framework can be modeled using existing AADL constructs however the interaction safety which requires modeling the CPIs can not be modeled.

Chapter 3

Cyber Physical Systems: Components and Properties

Cyber Physical Systems consists of three components or subsystems, these are: network of (or single) computing nodes, physical world or physical system, and cyber physical interactions.

3.1 Components of Cyber Physical Systems

The above mentioned components can have several properties and behaviors associated with them which are described below.

Computing Systems

Systems that have a computing, sensing, control, navigation and communication aspects associated with them are called computing systems. These systems can have following sub components :

- Sensing subsystem: Sensing subsystem of a computing system consists of a variety of sensors that sense the physical environment around it. These sensors often have a limited sensing range which might not be uniform in all directions [28]. For example, LIDAR used in autonomous vehicles [42]to detect nearby obstacles do not have a uniform pulse energy distribution [5].

- **Communication subsystem**: Communication subsystem of a computing system consists of radios for transmitting and receiving data from other computing nodes. Similar to sensing range, communication range might not be uniform across all directions. For example, radios used for communication between wireless sensor networks often have a disk shaped communication range [47].

- **Control subsystem**: Actuation subsystem of a computing system is responsible for controlling the physical system by generating appropriate control outputs. Control outputs are determined based on the information sensed by the sensing subsystem and the control logic inside the controller subsystem. For example, control system of an autonomous vehicle is responsible for generating speed and steering commands so that the vehicle stays on a predefined course/path.

- **Navigation subsystem**: For mobile computing nodes navigation sub system is another major component. This sub system is responsible for generating way-points or path along which the computing node will move. For example, in an autonomous vehicle navigation sub system generates a trajectory based on the destination location provided by user and it's current location.

Properties of Computing Systems

Computing systems can have three types of properties associated with them:

- **Computing Properties** : Properties of a computing system that affect it's computing, sensing, actuation and navigation sub system are defined as computing properties. For example, sampling time of sensors, priorities of threads running on a computing node, sleep and active states of a sensor etc. .

- **Physical Properties**: Properties of a computing system that are result of it's physical nature are defined as physical properties of a computing node. For example, mass and length of an autonomous vehicle, power dissipation of sensors deployed on human body etc. These properties often depend on the computing behavior of the system and can cause un-intentional interactions with the physical world.

- **Mobility Properties**: Properties of a computing system that characterize it's motion are defined as mobility properties. For example, velocity and direction of motion of an AV .

Physical Systems

Computing system are often embedded in the physical world with which it interacts. Similar to computing systems, physical systems have certain properties or behaviors associated with them. These are given below :

- **Physical Properties**: Properties of the physical system are defined as physical properties. For example, Blood perfusion rate of human tissues, radius of curvature and coefficient of friction of a road on which an autonomous vehicle travels .

- **Physical Laws**: In addition to the physical properties various physical laws are often associated with the physical world. These laws determine the change of physical

properties over time and sometimes over space. For example, heat equation determine the variation of temperature in a given region over time [21].

Physical properties and physical laws together determine the behavior of the physical system and resulting cyber phsyical interactions.

Cyber Physical Interactions

Cyber Physical Interactions(CPIs) in a CPS characterize the energy interactions (i.e. unintended interactions) as well as system defined interactions between computing-physical systems and computing-computing systems. CPIs for static computing nodes are different from mobile computing nodes and are described below.

Static Computing Nodes

For static computing nodes two types of CPIs exist, intended interactions and unintended interactions.

- **Intended Interactions**:Interactions that are system initiated and are essential for the functioning of CPS are called intended interactions. These interactions can be between two computing nodes or between a computing node and physical world. For example, communication between two sensors of a BAN, monitoring heart rate by EKG signals etc. Sensing and control sub-systems of a computing system cause intended interactions with the physical world. These two sub-systems are also connected using analog-digital and digital-analog converters. Sampling time and quantization parameters of these converters also effect the intended interactions. For example, if sampling time of LIDAR sensors [5] is high then the autonomous vehicle might not detect nearby obstacles and will fail to generate collision avoidance maneuvers leading to collisions.

- **Unintended Interactions**: Interactions due to operation of computing systems that have undesirable side effects on the physical world are called unintended interactions. These interactions represent transfer of energy between computing and physical systems and have a region associated with them. For example, heat dissipation from

sensor nodes can cause undesirable temperature rise of human tissue where it is deployed. In addition to space, these interactions also vary with time. For example, heat dissipation of computing node depends on it's current state. Unintended interactions are harmful to the physical world and may result in safety hazards.

- **Aggregate Effect**: The overlapping of unintended interactions (i.e spatial regions) of two or more computing nodes can cause the side effects to the physical world. These side effects can add up thereby increasing the chances of occurrence of a safety hazard. For example, the heat dissipation of two sensors can add up and cause more burn injuries than due to a single sensor .

Mobile Computing nodes

Due to mobility of computing nodes in a CPS, they exhibit two more types of interactions: planned interactions and unplanned interactions.

- **Planned Interactions**:Interactions that are caused by computing system while following a pre-planned trajectory are known as planned interactions. These interactions depend on the position, computing and physical behavior of a computing node, and can have more than one physical law (e.g., Newtons equations of motion, laws of thermodynamics etc) associated with them. E.g. Motion of an AV along the planned trajectory.

- **Unplanned Interactions**:Interactions that result from faulty operations of the computing system are called unplanned interactions. These interactions define the physical behavior of computing system during faulty operations. Unplanned interactions occur when the magnitude of certain physical properties of a computing system are above a **minimum threshold**. Due to the close coupling with the physical world, the threshold also depends on properties of the physical system. For example, skid of an autonomous vehicle along a curve is an unplanned interaction, it will only occur if the velocity of autonomous vehicle is above a certain threshold. This threshold depends on the curvature of road. Similar to planned interactions, unplanned interactions can have more than one physical law associated with them and vary with the position of a computing system. Both planned and unplanned interactions manifest as regions on

in the physical world which change with time. For example, in case of AVs the spatial regions indicate the position of AV during planned and unplanned interaction.

3.2 Examples of Cyber Physical Systems

In this sub section two examples of cyber physical systems are presented showing the above mentioned properties

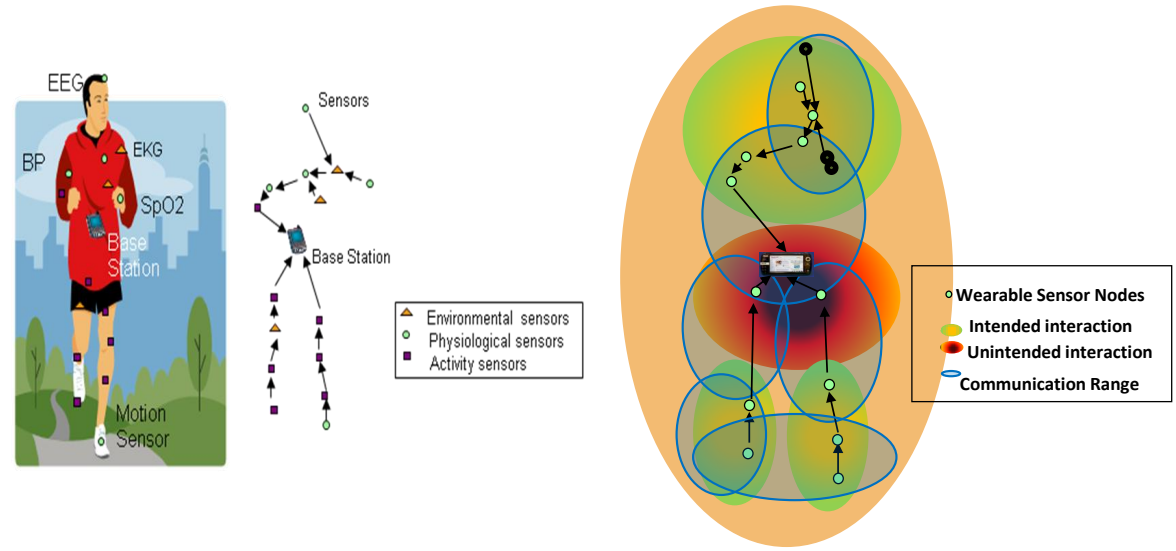
Cyber Physical Perspective of BSN

Body Sensor Network(BAN) are network of sensors implanted or worn on human body to monitor the physiological state (as shown in Fig 3.1). These sensors can run different tasks such as EKG based security algorithm [43], control algorithm for regulating mean arterial blood pressure [33] etc. In addition, sensors form either single hop or multi-hop networks to transmit the sensed data to a base station. The sensing and communication tasks in a BSN often produce heat due to which the surrounding tissue temperature rises [40]. If the temperature is not controlled, it can lead to burn injuries. Various components of a BSN (i.e sensors, medical devices etc) along with their operations can be viewed from a CPS perspective.

The sensors in BSN form the computing system, the human body whose state is being monitored forms the physical system. Sensing, communication and control tasks carried by sensors to ensure proper operations of BAN are the intended interactions whereas the heat dissipation of sensors and the resulting temperature rise will be the unintended interaction. Fig 3.1 illustrates these views.

Cyber Physical Perspective of Autonomous Vehicles

The architecture of an Autonomous vehicles(AVs) and their operations can be viewed from the perspective of Mobile Cyber Physical Systems (MCPS) and are shown in Fig 3.2. Sensing, navigation and control sub systems govern the decision making of an autonomous vehicle and have a computing nature, thus they can be represented as a computing (cyber) system. Physical environment around an autonomous vehicles such as road conditions, obstacles and it's own vehicle dynamics form a physical system. The interaction between these two systems cause an autonomous to move and thus can be modeled as CPIs . Two types of CPIs exist : i) The trajectory generated by navigation subsystem of an AV cor-



Operational view Logical view Cyber Physical view

Figure 3.1: Cyber Physical System's Perspective of Body Sensor Networks

responds to planned interaction because the motion of autonomous vehicle is preplanned and is intentional from it's perspective. Whereas, ii) The path traveled by an autonomous vehicle during a skid is due to it's improper operations (e.g. incorrect velocity control output) and is not preplanned, thus it can be represented as unplanned interaction. In addition, the interaction between computing sub-system and vehicle dynamics of an AV which occurs by means of sensors, actuators, D/A, A/D are considered as CPIs. CPS perspective of AV is shown in Figure 3.2.

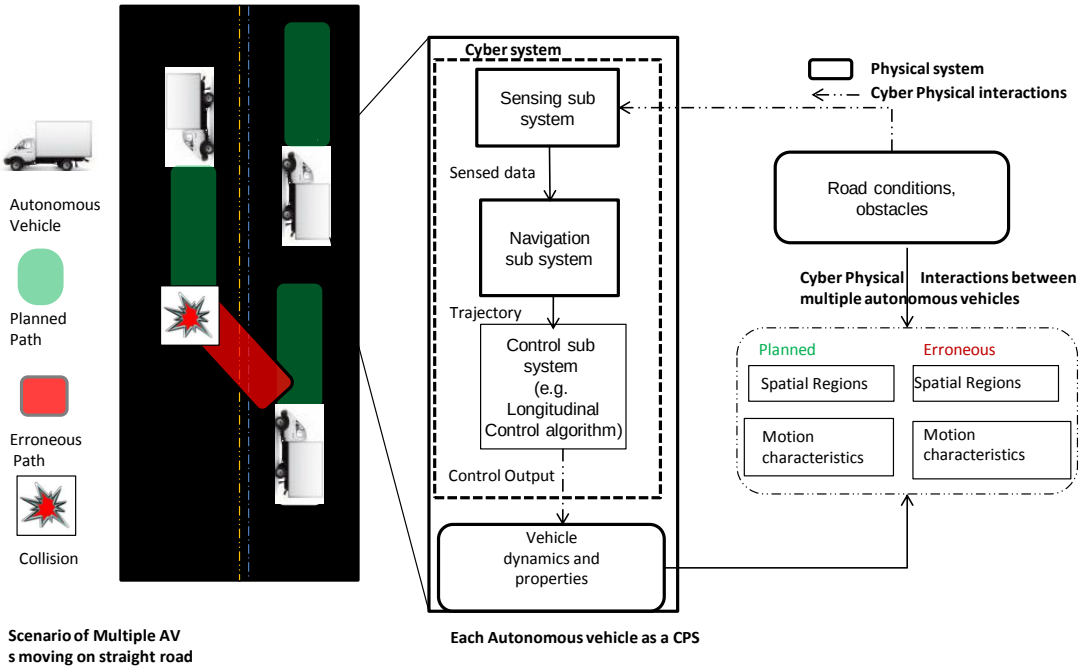


Figure 3.2: Cyber Physical System's Perspective of Autonomous Vehicles

Chapter 4

Modeling Cyber Physical Systems

In this chapter, first the modeling requirements for CPS are presented and then based on them corresponding modeling abstractions are presented. The modeling abstractions are domain independent and can be used to model different aspects of CPS such as intentional, unintentional, planned and unplanned interactions.

4.1 Modeling Requirements of CPS

CPS modeling requirements are based on the characteristics and properties specified in the previous chapter. A modeling framework should support:

- Modeling of computing system's individual sub systems, their computing and physical properties 3.1.
- Modeling of computing system's motion characteristics under planned and unplanned interactions. 3.1
- Modeling of physical system's properties and physical laws 3.1 that affect CPIs.
- A CPS can have large number of computing nodes, each with their own set of properties and CPIs with the physical system. Thus modeling of a computing unit and it's local CPI should be supported in the framework.
- Modeling of regions in physical world denoting the extent of CPIs 3.1,3.1.
- Modeling of side effects on the physical world due to CPIs 3.1.
- Modeling of control logic that generate planned motion and control unplanned motion. 3.1
- Modeling safety thresholds of the physical system 3.1.

4.2 Modeling Abstractions of CPS

In this section, generic modeling abstractions of CPS are presented. These constructs are shown pictorially in figure4.1

Modeling Computing System

Each computing node is modeled as *Computing unit* construct. This construct can represent a computing, sensing or actuation system and translates to modeling requirement . It facilitates the specification of computing, physical, and mobility properties of a computing node using following sub constructs. First, *Computing Property*: The construct characterizes the computing property of a computing node. A computing property can have units and data types associated with it. Second, *Physical Property*: This property characterizes the physical behavior of a computing node.

Modeling Physical System

Physical unit: A physical unit construct is used to abstractly represent a physical system. It translates to requirement 4.1. To denote physical system's properties and associated dynamics following sub constructs are developed. First, *Physical Property*: The construct characterizes the properties of the physical system. Similar to computing system's properties, a physical property can have units and data types associated with it. Second, *Physical Process*: This construct is used to model physical laws governing a physical system. Algebraic and differential equations representing a physical law can be specified here.

Modeling Cyber Physical Interactions

CPIs of a computing node is modeled using following constructs. First, *Region Of Interest(ROIn)*:This construct facilitates the modeling of the intentional interactions in a CPS. It has two sub-constructs, *Monitored Parameters*: This construct models the system parameters that are affected by the intentional interactions. Monitored parameters are the subsets of physical properties of either physical system or computing system. *Region Boundary*: This construct represents the limits of the bounded region within which the intentional interactions are confined. The region boundary depends on the variation of the monitored parameters. *Region Of Impact(ROIm)*:This construct facilitates the modeling of the unintended interaction and the side effects on the physical world in a CPS. It addresses requirement . It has two sub-constructs, first *Impacted Parameters*: This construct models the physical system's parameters that are affected by the unintentional interactions. Impacted Parameters are the subsets of physical system properties. Second, *Physical Process*: This

construct is used to model side effects which are generally represented as physical laws. *Region Boundary*:This construct is similar to region boundary in RegionOfInterest. However, region boundary of ROIm depends on the physical properties and dynamics. Since the physical dynamics are governed by differential equations, the region boundary can be specified by boundary conditions on the equations. These conditions are generally limits on the physical properties outside the ROIm. For example, in case of temperature rise in human body, we can assume that the body temperature is $39.2\,^{\circ}\mathrm{C}$ outside the ROIm. We can then employ this boundary condition to the associated differential equation to obtain region boundary. *Unintended Region of Mobility (UIROm)* Spatial regions of the physical world in which unplanned interactions of a computing system occur over a given period of time is modeled by UIROm construct. UIROm at a particular time instant gives the region in physical space that is occupied by a computing node because of it's physical characteristics. This construct addresses requirement 4.1. A UIROm can be characterized from *Physical Process* and *Computing Mobility* sub-constructs.*Impacted Parameters*: Impacted parameter are same as defined in case of *Region of Impact*. ***Physical Process***: The behavior of a computing system during an unplanned interaction is modeled using this construct. This behavior can be defined in terms of either the control system's logic during an unplanned interaction or by a physical law. For example, during an skid situation if the autonomous vehicle is equipped with a traction control system, it will automatically detect the skid and try to bring the vehicle back on road. The traction control system can be modeled using this construct. This construct also describes how impacted parameters change over time.*Computing Mobility*: This construct is used to describes the initial position of a computing node and associated motion equations that completely capture the mobile behavior of a computing node during an unintended interaction4.1. For example, if the motion equations of a autonomous vehicle are different due to the working of traction control system, then it can be described using this construct. *Minimum Threshold*: This construct is used to model the condition that for unplanned interactions to occur, impacting parameters should be greater than a specific threshold. *Intended Region of Mobility (IROm)*: Spatial extent of the physical world in which planned interaction of a computing system occur over a given period of time is modeled by IROm construct. Similar to UIROm, IROm can be character-

ized from **Physical Process** and **Computing Mobility** constructs. However, the difference being that in *Physical Process* we can define the controller's logic of keeping the vehicle in intended region of mobility (i.e. along the planned paths) and in computing mobility we can define the motion dynamics of computing system during planned interaction. This construct addresses requirement 4.1.

Modeling Cyber Physical System

All the previous constructs are used to model different components of a CPS. However to model multiple computing systems in a CPS and the fact that each each computing node has it's own CPI, additional constructs are needed which are described further. *LCPS*: A computing node and it's individual CPI are modeled using LCPS (Local Cyber Physical System) construct. Each LCPS has a computing unit and can have *Region of Interest*, *Region of Impact*, *Intended Region of Mobility* and *Unintended region of mobility* as sub constructs. *GCPS*: Multiple LCPSs in CPS form a GCPS (Global Cyber physical system), this construct has multiple LCPS's as it's sub constructs. *Safety Threshold* This construct allows system designers to specify limits on impacted parameters beyond which the system is declared to be unsafe. *Aggregate Effect*: In a CPS, interactions can also occur between two or more LCPS's. The interactions usually occurs when either *ROIn's* or *ROIm's* of LCPS's overlap. For example, the cumulative thermal effect of computing nodes (in a BSN) on a particular area of physical environment (overlapping of ROIMs) can be modeled using this construct.

Analysis governing parameters

In addition to the above mentioned constructs that are based on MCPS characteristics, few more constructs are defined that govern complexity and accuracy of safety verification logic. *Time Duration*: Time duration ($t_{duration}$), is the duration for which the system will be analyzed in the safety verification logic. During this period, if the safety verification logic detects any safety hazard which can lead to violation of safety threshold, the system will be declared as unsafe. *Sampling Time*: Sampling time is a parameter used in the safety verification logic to repeatedly compute computing systems position, impacting, impacted parameters, UIROm, IROm and their intersections. Sampling time affects both complex-

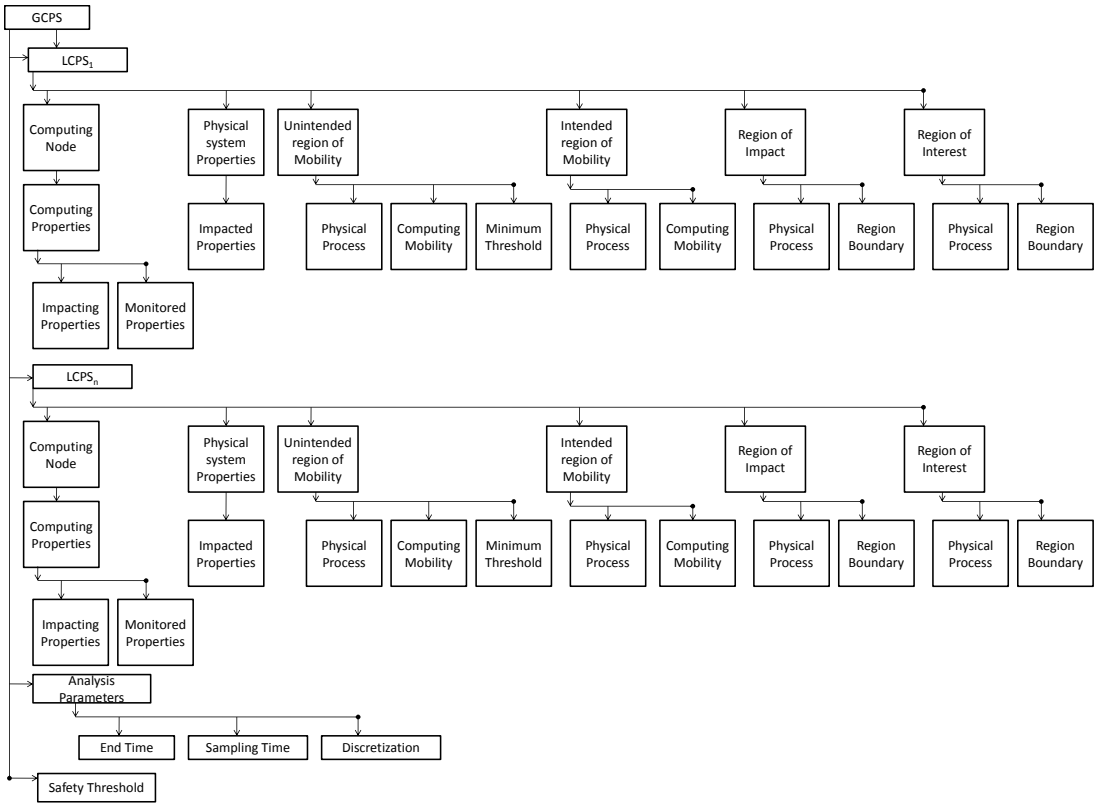


Figure 4.1: CPS constructs

ity of safety verification logic and accuracy of analysis results as explained in Chapter 5. *Discretization*:This construct allows system designers to provide differential equation solver parameters such as initial condition, boundary value conditions and discretization values. The solver configuration will be used if the CPS model contains a physical processes that are specified in terms of differential equations.

4.3 CPS Annex: A Modeling Framework for Cyber Physical Systems

CPS modeling constructs described in the previous subsection are implemented by extending the AADL language.

Introduction to AADL

AADL [2] is an industry standard language for modeling the architecture of Embedded Real Time Systems. The language provides several abstractions specific to embedded system thereby allowing designers to model the system in a iterative manner, as a result, they can verify that the design meets the requirements at every stage. Another benefit of using

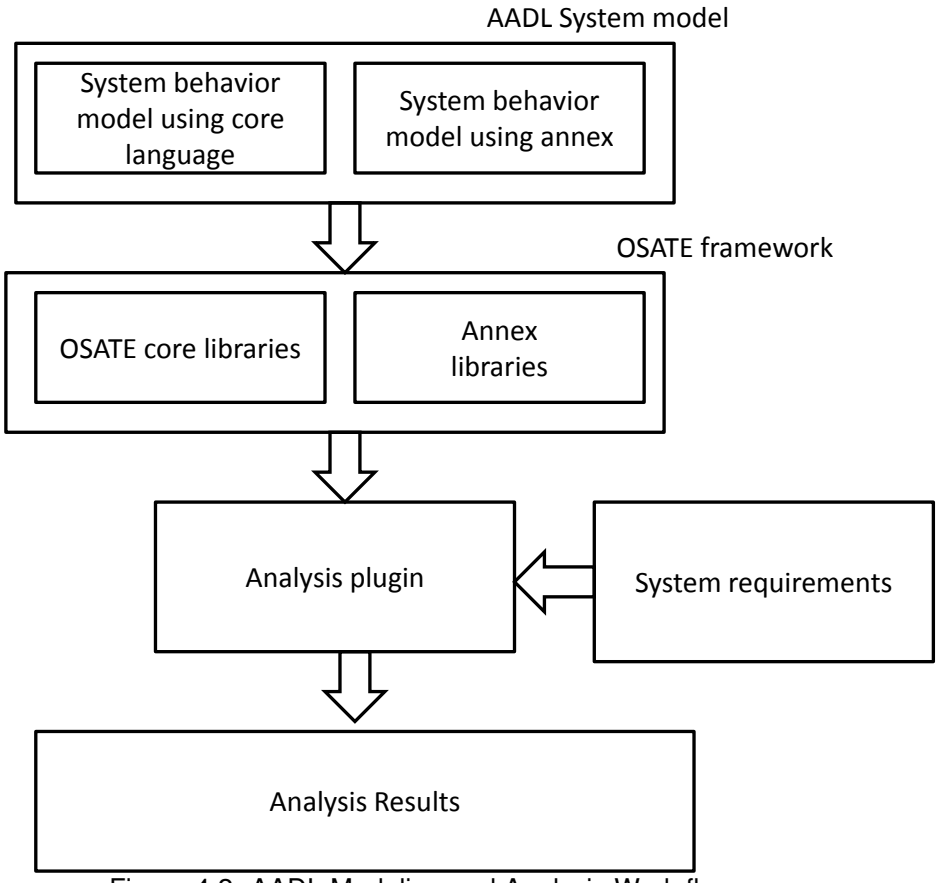


Figure 4.2: AADL Modeling and Analysis Work-flow

AADL lies in extensibility. The core language can be easily extended to model system's fault tolerance, error handling and several other behaviors. The extensions are often referred as annexes in AADL terminology. Custom Plug-ins can be developed to analyze AADL models using OSATE framework [2]. OSATE framework provides several libraries that can parse AADL models, gather information(model details) and pass it to analysis plug-ins. If the AADL models contain user defined annexes, then the system designers should also provide libraries that can parse annexes when invoked by OSATE framework. Figure 4.2 shows the various stages involved during the modeling and analysis of AADL models.

Limitations of AADL

AADL core language does not support the specification of physical systems and cyber physical interactions. In particular physical dynamics which are generally represented by algebraic and differential equations are not supported in AADL. In addition, the cps modeling abstractions developed in previous subsections cannot be semantically represented by any of the existing components. Thus, to model Cyber Physical Systems in AADL, the core

```
Expression := SubExpression { (+ |-) SubExpression }
SubExpession:=Term { (* | /)Term}
Term:= Derivative Expression | Port Identifier |AADL  Property |
        DataAccessIndetifier.subcomponentidentifier
DerivativeExpression  :==  Del<DerivativeOrder><DepdentVariable><IndependentVariable>
                          |Pdel <DerivativeOrder><DependentVariable>
                            (<IndependentVariable>)+
```

Figure 4.3: Algebraic and Differential Equation's grammar

```
PhysicallProcess :=
(Initalization)(0-1) (Boundary Conditions)(0-1)
Expression+

Initalization :=
Expression+
```

Figure 4.4: Physical Process Construct

language should be extended with cpsannex.

CPS Annex for modeling CPS

In this section, cps annex grammar and cps annex library are presented. As shown in figure 4.2, cps annex library is integrated with OSATE framework so that it will be automatically invoked as and when an annex construct is encountered. *Specifying algebraic and differential equations*: Differential and algebraic equations are essential to model dynamics of the physical world, mobile behavior of computing nodes, side effects on the physical world due to cyber physical interactions etc. It is specified using grammar defined in Figure .

where *Del* is used to denote a derivative and *Pdel* to denote a partial derivative. Expression rule denotes a single algebraic statement.

Specifying dynamics of the physical world and contol logic of computing system: *Physical Process* construct is being developed to capture the dynamics of the physical world and unintended interactions of computing system (as shown in Figure 4.5). The construct is built on top of *Expression* construct and multiple expressions can be specified within it. In addition, there are *Initialization* and *Boundary Condition* constructs that can be used to initialize various parameters and specify boundary conditions for differential equations. At present only Dirichlet boundary conditions can be specified.

```
Intended Region of Mobility := {
( Physical Process)
 (Computing mobility)
}
Computing Mobility :={
Expression+
}
```

Figure 4.5: Intended Region of Mobility

```
Unintended Region of Mobility := {
( Physical Process)
 (Computing mobility)
 (MinimumThreshold)
}
Computing Mobility :={
```

Figure 4.6: UnIntended Region of Mobility Grammer

Specifying Cyber Physical Interactions: There can be four different types of CPIs as stated in the previous subsection, these are intended region of mobility, unintended region of mobility, region of impact and region of interest. Corresponding to them, four different constructs are developed *IntendedRegionOfMobility*, *UnIntendedRegionOfMobility*, *RegionOfImpact* and *RegionOfInterest*. *IntendedRegionOfMobility*: The construct is used to model the planned paths as well as the control logic of a computing node. It has two main sub constructs as shown in Figure 4.5, *PhysicalProcess* and *ComputingMobility*. *Physical Process* is described previously, *ComputingMobility* models motion equations. Motion equations consists of a set of algebraic or differential equations.

UnIntendedRegionOfMobility: The construct is used to model erroneous paths of a computing node. Similar to *IntendedRegionOfMobility* this construct has *Physical Process* and *ComputingMobility* as sub constructs. In addition, it has *MinimumThreshold* construct, which models the condition for erroneous interaction to occur. *MinimumThreshold* construct can have an algebraic conditions within it. *UnIntendedRegionOfMobility* construct is shown in Figure 4.6.

RegionOfImpact: The *RegionOfImpact* construct models the harmful side effects of

```
RegionBoundary:=
(Length = Number ; Width = Number)|
(IDENT CONDITIONOP NUMBER) |
(Expression)
```

Figure 4.7: Region of Impact Grammar

the computing node on the physical world. It has two main sub-constructs *PhysicalProcess* and *Region Boundary*. As stated in previous section ,*PhysicalProcess* can be used to model the equations that are associated with the side effect. *Region Boundary* on the other hand is used to specify the extent(spatial boundary) upto which the physical system will be effected. The boundary can be specified in three ways: 1) Specifying the length and width of the region, this can be used if region is a rectangle. 2) Specifying thresholds on one of the parameters (e.g. section of the human tissue where the temperature is above $37.0\,^{\circ}\mathrm{C}$), this can be used if the region is irregular 3) If the region corresponds to a geometric object, e.g., circle, then they can specify the perimeter equation of this object. The grammar for this construct is shown in Figure 4.7.

RegionOfInterest: The *RegionOfInterest* construct has two sub-constructs *PhysicalProcess* and *RegionBoundary*. These two constructs are already explained.

Chapter 5

Safety Analysis Algorithm for Cyber Physical Systems

In this section an algorithm is presented for modeling the safety of Cyber Physical Systems. The algorithm uses the modeling constructs presented in chapter 4.

5.1 Algorithm Description

The safety analysis algorithm for determining the safety of the physical system in a CPS can be broken down into two parts : i) analyzing interactions due to mobile nature of computing nodes, Algorithm 1 and ii)analyzing energy interactions, Algorithm 2.

Analyzing interactions due to mobility of computing nodes

The objective of safety verification logic is to determine if two or more planned or unplanned interactions in a CPS intersect, and then compute the value of impacted parameter to determine if the safety threshold is violated. Possible safety hazards are analyzed by checking three conditions : 1) intersection of IROm with another IROm: this hazard analysis checks if planned interactions of two or more computing units overlap, even though intended interactions are explicitly initiated by computing nodes it is still possible for safety criteria to get violated. ii) Intersection between UIROm and IROm : this hazard analysis checks if planned interaction of one computing node intersects with unplanned interaction of another computing node. iii) Intersection between UIROm/IROm and physical object : this hazard analysis checks if unplanned interactions of two or more computing nodes or a computing node and a physical object overlap, leading to a potentially dangerous situation.

Inputs and Outputs of the algorithm

The algorithm takes as input a CPS model specified using the above mentioned constructs and outputs whether the system is safe or unsafe.

Algorithm Description

Safety verification algorithm is presented in Algorithm 1. The algorithm begins with detecting unintended interactions and sets up ctime (current time, in Step 2) equal to zero, a variable that denotes the time instant at which we are evaluating values of impacting, impacted and IROm parameters (Step 4). After this, the algorithm determines if any planned or unplanned

interactions exist in CPS by checking the value of any impacting parameter is less than minimum threshold (Steps 5-7), if the condition succeeds then there are no unplanned interactions in the system and we have to check for overlap of any intended interactions (Step 22) . On the other hand, if unplanned interactions exist, then we compute the value of UIROm (using equations specified in physical process and mobility constructs) for each computing node(Step 8) and check it's intersection with UIROm (Steps 10-14)) and IROm (Step 16-21) of other computing nodes. During these step once an overlapping region is detected the algorithm computes the value of impacted parameter (Steps 12,18 and 24) to check if it's value goes above the threshold value (safety criteria). If the threshold is not violated we increments $ctime$ by sampling time and proceed to next iteration (Step 31).

Algorithm Termination Criteria

The algorithm terminates under the following situations :

- Value of an impacted parameter goes above the safety threshold leading to an unsafe situation.
- Value of current time exceeds time duration. This condition indicates that the value of impacted parameters did not rise above the safety threshold and the system is safe.

Algorithm Complexity

The complexity of the algorithm depends on the sampling time($t_{sampling}$) and number of computing nodes, n. Assuming it takes a constant time to compute intersection of UIROm and IROm regions (Step 16) of two computing nodes, it will take O(n^2) steps to compute the same intersection for all n nodes. Since we are also checking intersection of IROm and IRom (Step 22), UIRom and UIRom (Step10) it will take O(2*n^2) more steps in each iteration. Thus, the total number of steps in each iteration are O(3*n^2). We repeat the iteration $t_{duration}/\ t_{sampling}$ times (Steps 3-32). Hence, the total complexity of the algorithm will be **O(**$t_{duration}$*****n^2**/** $t_{sampling}$**)**.

If we increase $t_sampling$ time to reduce the complexity, we are less frequently computing the values of impacting, impacted, IROm(ctime) and UIROm(ctime), this may cause the algorithm to miss detection of any overlapping regions, leading to incorrect results. Thus, **there exists a trade-off between complexity and accuracy in the safety**

verification logic. It is up-to system designers discretion to choose an appropriate value of $t_{sampling}$.

Analyzing Energy Interactions

Algorithm 2 analyzes the energy interactions of the system. The algorithm begins by checking the value of physical system parameters (impacted parameters) that are affected by side effects due to unintended interactions. Since unintended interactions are modeled using ROIm construct and physical process sub-construct, the value of impacted parameters are computed using those two constructs. If the physical processes in the model contain differential equations, they can be numerically solved using analysis parameters that are specified in the CPS model. In the next step, the intersection of ROIm's of multiple LCPSs is checked, if they intersect the value of impacted parameter in these regions is recomputed using additive effect construct. This step essentially analyzes the aggregate side effect of to multiple computing nodes on a particular region of the physical environment. Assuming it takes O(k) to evaluate the physical process, the complexity of this part of the algorithm will be O(n^2*k).

Algorithm 1 Algorithm: Safety Verification Logic Using CPS Constructs

Set $ctime$ = 0
while $ctime \leq t_{duration}$ **do**
For all computing nodes compute impacting parameters of each computing node using equations equations specified in physical process construct of IROm
if impacting parameters $\leq$ Minimum threshold **then**
Goto Line 27
else
Compute UIROm($ctime$)
end if
if UIROm(ctime) of two or more computing nodes intersect **then**
Compute the value of impacted parameters
if impacted parameters $\geq$ any safety threshold **then**
Declare system as unsafe,return
end if
if IROm(ctime) of one computing nodes intersects with UIROm(ctime) of other computing nodes **then**
Compute the value of impacted parameters
if impacted parameters violate any safety threshold **then**
Declare System as unsafe, return
end if
end if
end if
end while
if UIROm(ctime) of a computing node intersects with a Physical Object **then**
Compute the value of impacted parameters
if impacted parameters violate any safety threshold **then**
Declare System as unsafe, return
end if
end if
if IROm(ctime) of two or more computing nodes intersect **then**
Compute the value of impacted parameters
if impacted parameters violate any safety threshold **then**
Declare System as unsafe, return
end if
end if
if $ctime$ == $t_{duration}$ **then**
Declare system as safe, return
end if
increment $ctime$ by $t_{sampling}$

Algorithm 2 Algorithm: Safety Verification Logic Using CPS Constructs, analyzing energy interactions

1: Compute the value of impacted parameter within each LCPS in the ROIm using equations provided in Physical Process and analysis parameters
2: **if** impacted parameters violate any safety threshold **then**
3: Declare System as unsafe, return
4: **end if**
5: **if** ROIm of two or more computing nodes intersect **then**
6: Recompute value of impacted parameters in the intersected region using aggregate relation
7: **if** impacted parameters violate any safety threshold **then**
8: Declare System as unsafe, return
9: **end if**
10: **end if**

Chapter 6

Application of CPS Modeling Constructs and Safety Analysis Algorithm

In this chapter, two case studies are presented that uses CPS modeling constructs and safety analysis algorithm to verify the safety of the physical system. These case studies are : i) evaluating thermal safety of BSN, and ii) evaluating safety of passengers traveling in an Autonomous Vehicle that collides with a guard while negotiating a curve.

6.1 Evaluating Thermal Safety of BSN

In this section we consider the safety of human body under two different operations of BAN - computation and communication.

Scenario Description

To determine the effect of computation on human tissue we will consider a TelosB mote interfaced with finger tip Pulse Oximeter sensor. The pulse oximeter probe is always in contact with the tissue, during it's operation it passes light at a particular frequency. The energy transfer from pulse oximeter probe to the finger skin is the major source of thermal energy in the system. To derive physiological values from sensed light intensity, the senor node performs computation which results in generation of heat. Here it is assumed that there is no other pulse oximeter node in proximity with the sensor node. the computation workload on sensor node is assumed to be constant over the period of operation of the sensor node.

To consider the effect of a communication on human tissue, we consider a multi-hop wireless communication network on sensor nodes implanted in human body. The sensor nodes consists of a low capability computing entity , radio and sensor interfaced to it. Due to communication work load sensor nodes dissipate heat which rises the temperature of the human tissue, it is further assumed that the senor nodes are placed close to each other so that the heat dissipation of two or more sensor node adds up thereby increasing the average temperature of human tissue. The multi-hop communication protocol considered in this section is cluster based [23]. In this protocol from the set of worker nodes few leader nodes are selected by the base station. Each non leader worker node has to select a leader node as it's parent to forward the sensed information. The worker nodes select a leader by

listening to the beacon messages generated by various leader nodes and the selecting the one with high signal to noise ratio. The leader node along with worker nodes that select it as it's parent form a cluster, with the leader being the cluster head. Worker nodes forward the information to cluster heads, which forward the same to the base station. In [40] authors show that the high workload on cluster head can significantly increase the temperature of the human skin. The temperature rise is mainly due to absorption of electromagnetic radiation from the antenna head during cluster head communication with the base station. The authors also propose a thermal aware algorithm for periodically rotating the cluster head to reduce the temperature rise.

Analysis Methodology

ISO 60601 standard for Medical Electrical Equipment defines safety of a health system on human body, as avoidance of unacceptable risks during it's normal or faulty operations. The standard lists several safety categories of which one of the most important is thermal safety. Thermal safety is characterized by a threshold temperature such that if a device exceeds this temperature during it's operation then the physical environment incurs thermal damage. During thermal safety analysis the temperature of the physical environment of a device is monitored during several stages of it's operation and checked against thermal safety threshold. The principle of hazard based safety engineering is often employed for thermal safety analysis of a health system as suggested by [45]. Hazard based safety analysis techniques requires following steps for analyzing thermal safety of BAN. **Characterization of energy source**: The principal energy sources are the *Computing Units* in the model whose *Physical Properties* characterize the energy sources. They can be of two types - 1) constant temperature source (a heat source that supplies energy at a constant temperature) and 2) constant power source (a heat source that supplies energy at a constant power). Thermal safety standards often limit the maximum temperature reached by constant temperature sources.

Thermal characterization of physical environment: The Physical unit in case of a BAN is primarily the human body and is modeled as a mass that absorbs heat. The Region Of Impact is defined on the physical unit and the Physical properties of it characterize its thermal behavior in terms of a thermal damage parameter. Thermal damage parameter

is estimated based on an Arrhenius model of temperature rise of the physical unit with respect to duration of exposure to heat source as suggested by by Moritz and Henrique [24]. A threshold temperature can be calculated such that if the temperature of the physical unit exceeds this threshold then thermal damage is sure to occur. The threshold temperature T_{thresh} is given by

$$T_{thresh} = \frac{E_a}{R[ln(\tau) - R.ln(\frac{1}{A})]}, \tag{6.1}$$

Heat transfer process: The heat transfer process controls the manner in which the temperature rise of the physical unit occurs. Heat transfer is due to thermal conductance and thermal radiation from heat sources and convective cooling. The thermal conductance from heat source to physical unit can be given by,

$$H = K \bigtriangledown^2 T, \tag{6.2}$$

where, H is the amount of heat added to the system from heat source, K is the thermal conductivity of the physical unit and $\bigtriangledown^2$T gives the distribution of temperature in the monitored area. Thermal radiation can be accounted for using Stefan's law for hot body radiation where constant temperature sources are considered to calculate the heat input to the system as follows:

$$H = S \times \sigma(T_r^4 - T_a^4), \tag{6.3}$$

where S is the Surface area of the hot body, σ is the emissivity, T_r is the surface temperature of hot body and T_a is the ambient temperature. Convective heat transfer can be modeled using a linear model H = -b(T-T_b) where b is constant coefficient and T_b is temperature of convective fluid in the physical system (for example blood). The negative sign indicates that it has a cooling effect to the system. Heat transfer due to electromagnetic radiation is modeled in terms of the Specific Absorption Rate (SAR). The SAR value is dependent on the characteristics of the antennae in the electromagnetic energy source and is evaluated based on the methodology suggested by [40].

Modeling thermal side-effects of BSN computation using CPS constructs

In this subsection, the thermal side effect of pulse oximeter on the human tissue is modeled using CPS constructs presented in Chapter 4. The pulse oximeter is modeled as a computing system. Computing properties of a pulse oximeter such as current drawn (which

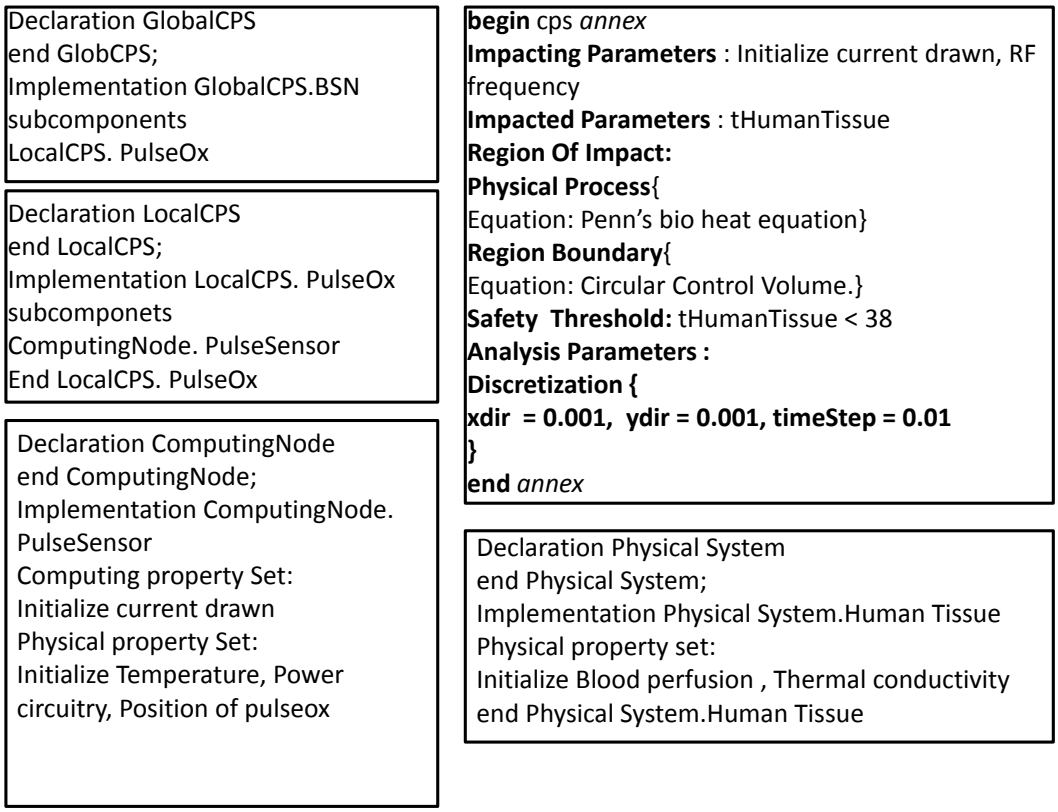


Figure 6.1: Modeling thermal side effects of computation of Body Sensor Networks using AADL

internally depends on the state of pulse oximeter) , and RF frequency properties of the radio are modeled as Computing system properties. The physical properties of pulse oximeter such as power dissipation of the circuitry are modeled as physical system properties. Pulse oximeter and the surrounding tissue is modeled as LCPS. Heating of tissue due to operations of pulse oximeter is a side effect, so it is modeled using RegionOfImpact(ROIm) construct. The region of the human tissue around the pulse oximeter that get's heated is modeled as enclosing region construct. This region is modeled as a circular control volume. The temperature rise of the human tissue is based on Penn's equation (Equations 6.1 and 6.2) and is modeled using **Physical Process** construct. Sine Penn's equation is a partial differential equation and is solved numerically, additional information on the discretization such as size of each cell in x direction and y direction, and the time step is provided as part of **Discretization** construct. These models are explained in presented in Figure 6.2.

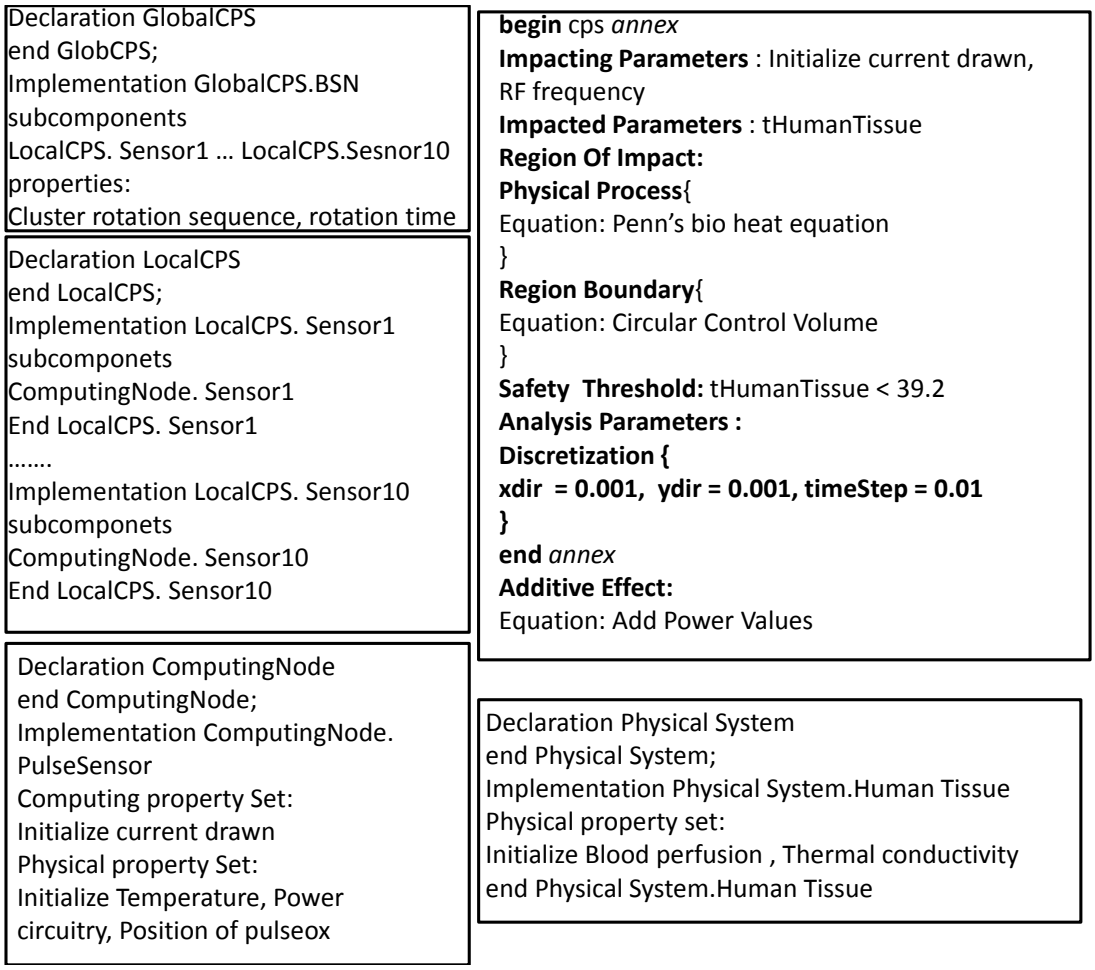


Figure 6.2: Modeling communication side effects of Body Sensor Networks using AADL

Modeling thermal side-effects of BSN communication using CPS constructs

The thermal side effects of multi-hop BSN can be modeled using MCPS constructs in a manner similar to Section 6.1. The cluster rotation sequence which tries to minimize the heating of surrounding tissue is modeled as a property in AADL. Each computing node and it's interaction with the surrounding physical tissue is modeled as LCPS. As explained in Chapter 3, due to proximity of computing sensors RegionOfImpacts of computing senors can overlap, this situation is modeled using **AdditiveEffect** construct. In multi-hop BSN, the aggregate side effect is equal to addition of SAR values of different sensors.

Table 6.1: Skin temperatures after eight hours of pulse oximeter operation at different device temperatures (Burn threshold $39\,^{\circ}\mathrm{C}$)

Device Temperature	Maximum Skin Temperature
$43\,^{\circ}\mathrm{C}$	$38.2\,^{\circ}\mathrm{C}$
$43.5\,^{\circ}\mathrm{C}$	$38.5\,^{\circ}\mathrm{C}$
$44.0\,^{\circ}\mathrm{C}$	$39.2\,^{\circ}\mathrm{C}$
$44.5\,^{\circ}\mathrm{C}$	$39.4\,^{\circ}\mathrm{C}$
$45.0\,^{\circ}\mathrm{C}$	$39.7\,^{\circ}\mathrm{C}$

Analysis Results and Verification

The pulse oximeter model presented in Section 6.1 was analyzed using safety analysis algorithm (Chapter 5),to determine the safety of human tissue. The algorithm computes the temperature at different points of skin using by solving Penn's equation using FDTD approach. Table 6.1 shows the maximum temperature reached during eight hours of operation of pulse oximeter at various device temperatures. A sample thermal map of temperature distribution for pulse oximeter device temperature of $44\,^{\circ}\mathrm{C}$ is shown in Figure 6.3. It can be seen that the maximum temperature rise if $39.2\,^{\circ}\mathrm{C}$, which violates thermal safety requirement. An experimental study done by [22] indicates that at pulse oximeter device temperature of $44\,^{\circ}\mathrm{C}$ blisters begin to appear on human skin.

In order to determine the tissue temperature rise due to communication operations of BSN, Penn's bio heat equation is solved using FDTD solvers. The aggregate side effect is calculated by summing SAR values at each grid point. The power consumption of leader worker node executing Ayushman application was 60mW, while that of non leader worker node was 12mW. This power consumption was experimentally measured for TelosB motes at 0 db - 7db radio attenuation. Each leadership sequence was operated for duration of 1000 sec and then for a duration of two days. The results for 100 seconds exposure match with [40]. Table 6.2, shows temperature rise for different leadership sequences. The maximum temperature is less than $39\,^{\circ}\mathrm{C}$ for all the cases. It can be observed from the results that for short duration of exposure, temperature rise of different leadership sequences does not vary. However for prolonged exposures, the leader ship sequence plays an important role.

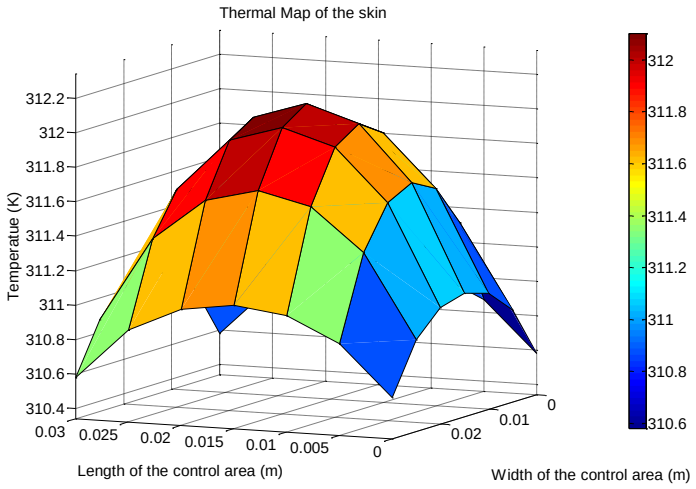


Figure 6.3: Thermal map of fingertip skin for 8 hrs of pulse oximeter operation at $10\,^{\circ}\mathrm{C}$ temperature

Table 6.2: Tissue temperature rise for different leadership sequences (Burn threshold $39\,^{\circ}\mathrm{C}$)

Leadership Sequence	**Maximum Tissue Temperature**	
	1000 Sec Exposure	2 Days Exposure
(5 2 8 6 1 7 3 4 10 9)	$37.1145\,^{\circ}\mathrm{C}$	$37.1632\,^{\circ}\mathrm{C}$
(5 7 4 1 6 10 8 2 9 3)	$37.1130\,^{\circ}\mathrm{C}$	$37.1614\,^{\circ}\mathrm{C}$
(1 6 9 10 2 7 5 4 3 8)	$37.1124\,^{\circ}\mathrm{C}$	$37.1757\,^{\circ}\mathrm{C}$
(5 7 1 9 10 8 4 2 6 3)	$37.1130\,^{\circ}\mathrm{C}$	$37.1585\,^{\circ}\mathrm{C}$

6.2 Evaluating Safety of Autonomous Vehicles

This section presents a case study to apply the modeling and analysis framework for verifying safety of passengers traveling in an AV along a horizontal curve. Horizontal curves are road segments that provide gradual transition between two tangential roadway strips. The following subsections describe the scenario considered in the case study (Section 6.2), AV's behavior along horizontal curves (Section 6.2), modeling AV's behavior and properties of horizontal curve using MCPS constructs (Section 6.2), safety analysis (Section 6.2), and validation of the safety analysis (Section 6.2).

Scenario Description

We consider an autonomous pickup truck (an AV) driving along Mile Post 44 (a horizontal curve) on Arizona-83 highway [41]. Figure 6.4 shows direction of motion of the AV (the dashed arrow in the figure) along the horizontal curve. The solid line in the figure represents

the intended trajectory of the AV (as planned by the AV's navigation subsystem) through the right most lane. The trajectory is assumed to be along the center of the lane. The dashed line (shown parallel to the planned trajectory of the AV) represents the guard rail beside the right most lane. The distance between the guard rail and the AV is the sum of the shoulder width and half the lane width. Table 6.4 lists: (i) the parameters of the curve such as curve radius [41], distance between the AV and the guard rail [10], the coefficient of friction of road [3], and the type of guard rail along the curve [9]; (ii) the parameters of the AV such as type of AV (i.e. pickup truck), AV's mass (i.e. the mass of a pickup truck) [9] and (iii) the parameters related to the safety verification logic such as sampling time (i.e. the time between two consecutive computations of AV's speed) and the total time duration for which safety analysis is performed. The next subsection describes the control algorithms that are part of the AV.

Control Algorithms of the Autonomous Vehicle

We assume that the AV employs lateral and longitudinal control algorithms [32] as part of the control subsystem. The lateral control algorithm generates steering angle (i.e lateral control angle) based on AV's current position and the next way point as given by the following equation:

$$\delta = arctan[2l(3y - xtan\theta)/x^2], \tag{6.4}$$

where δ is the steering angle, l is the wheel base of the vehicle (i.e.), (x, y) is the next way point, and θ is the heading angle (i.e.). The longitudinal control algorithm generates speed based on the preceding and following AV's speed as per the equation given below:

$$v_r = v_p + k1(v_p - v_f) + k2(L_r - L_m) \tag{6.5}$$

where v_r is the speed of the AV, v_p is the speed of the preceding vehicle, L_r is the minimum longitudinal distance between two vehicles, L_m is the measured inter vehicular distance, $k1 = m1\|L_r - L_m\|/\|L_r\|$, $k2 = m2k1$, and $m1$ and $m2$ are control gains.

The vehicle dynamics of AV during planned motion is given by [26]:

$$x' = v_r * cos\theta, y' = v_r * sin\theta, \theta = \frac{v_r}{l} * tan\delta, \tag{6.6}$$

where l is the wheel base of the vehicle.

Table 6.3: Abbreviated Injury Scale

AIS	SEVERITY	TYPE OF INJURY
0	None	None
1	Minor	Superficial Injury
2	Moderate	Recoverable
3	Serious	Possibly Recoverable
4	Severe	Not Fully Recoverable Without Care
5	Critical	Not Fully Recoverable with Care
6	Fatal	Unsurvivable

We assume that the speed of the autonomous pickup truck is also within the observed speed limits. For the case study, we assume that any preceding and following vehicle have the same speed as that of the autonomous pickup truck. They are not shown in figure 6.4 since there would be no collision among the vehicles; thus causing no safety hazard. From safety verification perspective however it is important to analyze the probability of passenger injury because of collision with the guard rail when the AV skids because of speeding along the horizontal curve. The posted speed limit on the curve is 45 MPH. However, the observed speed of vehicles is 45-60 MPH [41]. From our observation, based on vehicle behavior (described in Section 6.2) the minimum speed at which the AV can skid is 48MPH on this horizontal curve.

Passenger Safety

The speed of the AV has an impact on the passenger safety when it skids. The passenger traveling in an AV is considered unsafe if the probability of a serious injury (Table 6.3) is greater than zero. The probability of serious injury, given the change in vehicle velocity in a collision is x, is computed in the safety verification logic using the following equation:

$$P = \frac{1}{1 + exp(4.0139 - 0.1252x)}. \tag{6.7}$$

In order to compute the probability of serious passenger injury in the safety verification logic, it is important to determine the speed and the angle at which the AV collides with the guard rail. These parameters can be obtained by analyzing the behavior of AV along the curve. The following subsection describes AV's behavior given the lateral and longitudinal control operations. The safety verification logic is then validated in Section 6.2 by comparing the computed serious injury probability with the probability of serious injury in a pickup

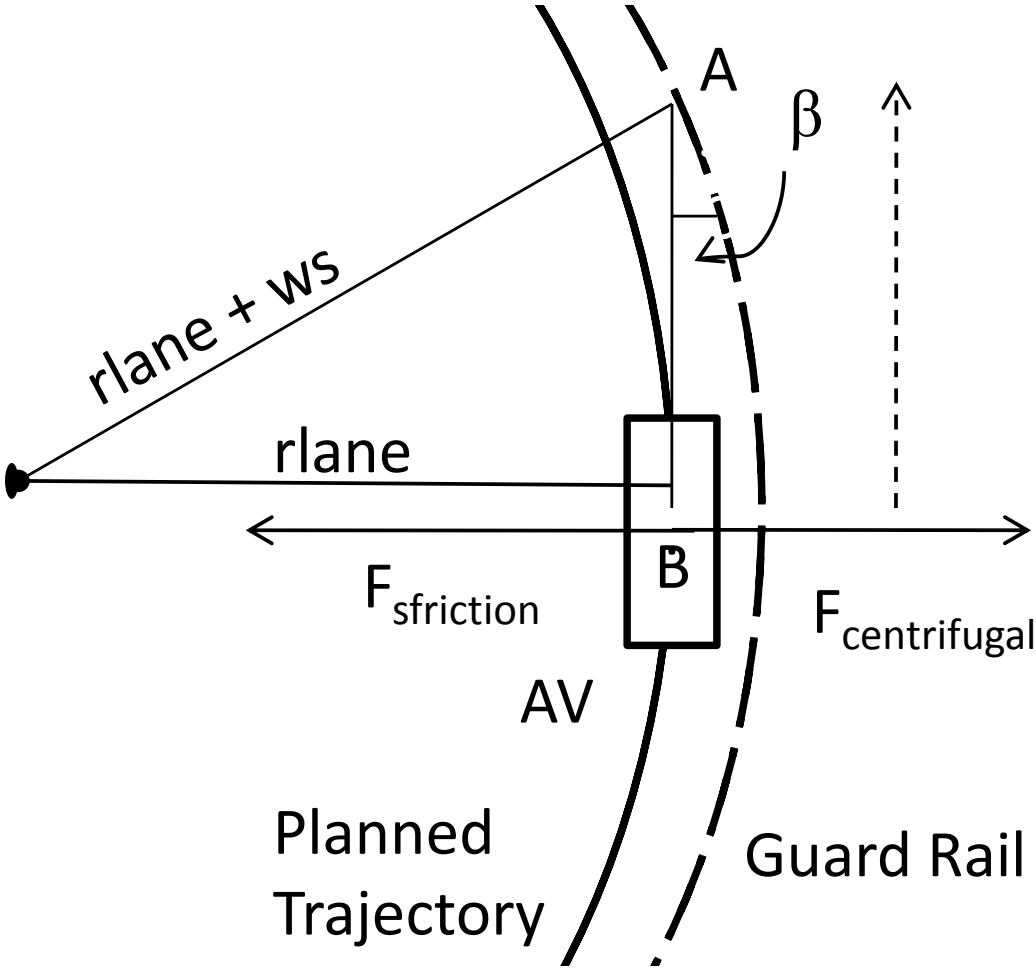


Figure 6.4: Autonomous Vehicle driving along a Horizontal Curve, r1 is the radius of lane 2

Table 6.4: Model Parameters

Parameter	Value	Parameter	Value
Coefficient of Static Friction	0.2	Radius of lane 1	50m
Mass of Vehicle 1	1600kg	Mass of Vehicle 2	2000kg
Sampling Time	0.005s	Type of Guard Rail	W-Bean

truck due to speeding along the same road segment based on data provided in AZ-83 assessment report [41].

Autonomous Vehicle's Behavior Along a Horizontal Curve

Speed of an AV can be used to characterize its behavior along horizontal curve by curvilinear equations of motion [18]. These equations are explained briefly using a free body

diagram shown in Figure 6.4. In the figure, r_{lane} denotes the radius of the cure, v_r is the instantaneous velocity of AV (represented by OB), $F_{centrifugal}$ and $F_{sfriction}$ represent centrifugal and frictional forces acting on it. $F_{centrifugal}$ is an outward acting force that pushes the vehicle away from the center O, whereas $F_{sfriction}$ provides necessary traction towards the center. Net force F_{normal} acting on a vehicle is the difference of these two forces (Equation 6.9), to negotiate a curve without skidding, this force should be negative (Equation 6.10). However, if the condition is violated, the vehicle will skid tangentially (same as direction of its instantaneous velocity) and collide with the guard rail. The angle (show as β) at which the AV collides the guard rail is called the impact angle. This angle is equal to ∠AOB because if we draw at tangent at point A, β is equal to 90 ° - ∠OAB, which is same as ∠AOB. Thus, ∠AOB given by Equation 6.8

$$\beta = \arccos \frac{rlane}{rlane + ws}. \tag{6.8}$$

where: ws: width of shoulder between guard rail and lane rlane: radius of lane β: impact angle

$$F_{normal} = F_{centrifugal} - F_{sfriction}. \tag{6.9}$$

$$F_{normal} < 0. \tag{6.10}$$

In the next subsection, the behavior of AV along a horizontal curve using MCPS constructs is modeled. These models are also implemented using AADL [2], and is shown in figure 6.5.

Modeling using MCPS constructs

AV's properties such as it's mass, velocity, steering angle, wheel base, and its type (i.e. pickup truck) are modeled as **computing properties**. Sampling time and quantization parameters of A/D, D/A connectors that interface sensors and actuators with the physical dynamics of AV are modeled as **computing properties**. Velocity (a.k.a. impact velocity) and angle (a.k.a. impact angle) at which the AV collides with the guard rails are **impacting parameters**. The properties of the curve, i.e. radius, coefficient of friction of road, type of

guard rail, distance between the AV and the guard rail are **Physical System Parameters**. Human injury is an **impacted parameter**. The behavior of an AV during skid is modeled as **UIROm**. Equation 6.10, which provides the condition for a vehicle to skid, is specified using the **Minimum Threshold** construct. Equation 6.8, which represents tangential direction of AV's skid, is modeled using **Computing Mobility**.

The velocity of an AV after collision (a.k.a. final velocity) with guard rail depends on the impact velocity and impact angle. This relation is shown in Table 6.5 and is specified using the **Physical Process** construct. The table is obtained by using LSDyna simulation software [6]. Pickup truck and guard rail models from [9] are input to the simulator. The probability of serious passenger injury given the difference of final and impact velocity (as shown in Equation 6.7) is modeled using **Physical Process** construct. The behavior of AV along planned trajectory generated by its navigation system can be modeled using **IROm** construct. The output trajectory is modeled by **Computing Mobility** construct. The lateral and longitudinal control algorithms (i.e. the Equations 6.2 and 6.5, respectively) are modeled using **Physical Process** sub construct. In order for the AV to be safe, the probability of serious passenger injury should not be greater than zero, this condition is modeled by the **Safety Threshold** construct. The next subsection describes the application of generic safety verification algorithm to analyze the safety of passengers in this case study.

Safety Analysis

Based on the MCPS model, the safety verification logic (presented in Table 1) is used to compute the probability of serious injury to passengers. Figure 6.6 shows the mapping of different stages of the safety verification logic 1 to steps used for analyzing the safety of passengers.

We determine the speed of AV (an impacting parameter) using using Longitudinal Control Algorithm specified as part of Physical Process in **IROM** construct, (Step 3)). The speed is checked to determine if there is a skid by using equations specified in **Minimum Threshold** construct (Step 4). If the condition is satisfied we compute impact angle (an impacting parameter) and final velocity after the collision with guard rail. The final velocity is then used to compute probability of serious human injury (an impacted parameter). We finally check if the probability is greater than zero (a safety threshold) to declare the system

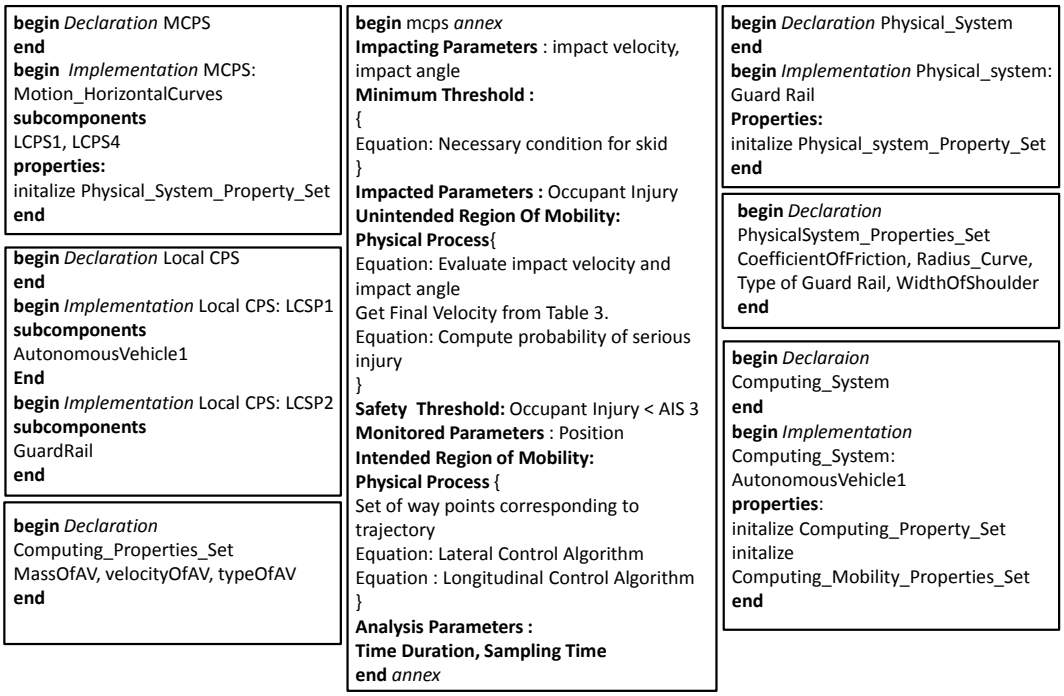


Figure 6.5: Modeling the motion of Autonomous Vehicle Along Horizontal Curves using AADL

Table 6.5: Relation between Final velocity, Impact Velocity and Impact Angle for Pick Up Truck computed using LS-Dyna

Impact Velocity (m/s)	Impact Angle (degrees)	Final Velocity (m/s)
27(60MPH)	8.4	24.7
26.5 (59MPH)	8.4	24.5
26.1 (58MPH)	8.4	23.8
25.7 (57MPH)	8.4	23.3
25.2(56MPH)	8.4	23.5
24.7 (55MPH)	8.4	22.7
24.3(54MPH)	8.4	22.5
23.8(53MPH)	8.4	22
23.4(52 MPH)	8.4	21.6
22.9 (51MPH)	8.4	22.5
22.5(50MPH)	8.4	21
22.0(49MPH)	8.4	20.2
21.6(48MPH)	8.4	21.6
21.1(47MPH)	8.4	21.15
20.7(46MPH)	8.4	20.7

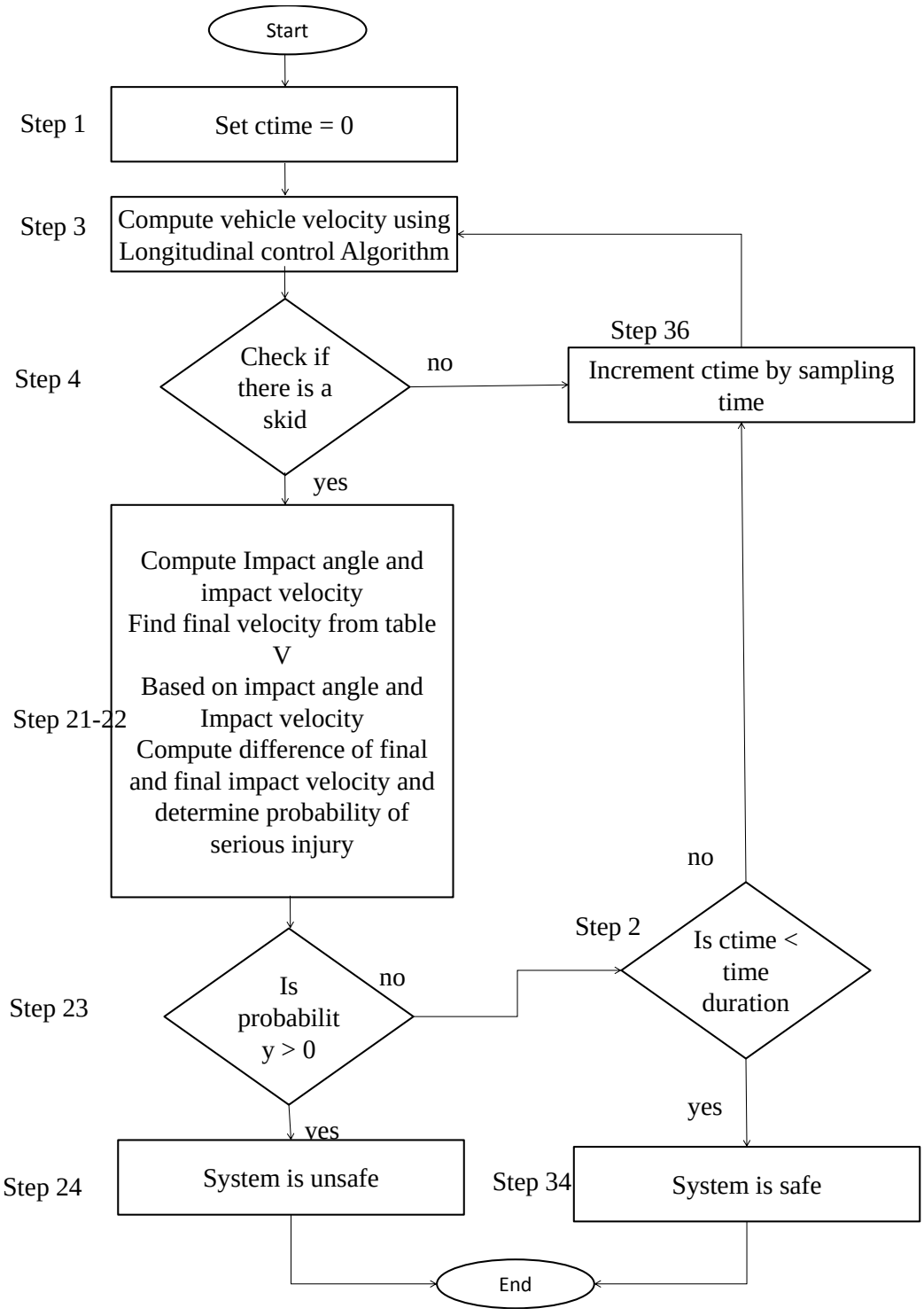


Figure 6.6: Analysis Steps in Case study

as unsafe (Step 24) or continue with the next iteration (Step 2).

Validation

In order to evaluate the correctness of safety analysis algorithm we computed the probability of severe injury to passengers traveling in the autonomous pick up truck along MP 44 and compared it with the probability of having serious accident by a pick up truck due to speeding along MP44 based on the data provided in AZ-83 report [41]. META STATEMENT. We assume the speed of AV follows a normal distribution with mean of 55 MPH and standard deviation of 6.8. These values are based on data published by Arizona Department of transportation which indicates that speed distribution along the highways forms a normal distribution [1]. The report further states that the mean and standard deviation parameters of the distribution along I-10 Grant Road ATR, which has a posted speed limit of 55MPH, is 65 MPH and 10.0 respectively. Since most of vehicles going on Mile Post 44 of AZ-83 are speeding, we computed scaled mean and standard deviation values for speeding vehicles along I-10 and used it as the speed distribution of AV.

The validation has two steps:

- **Probability of serious injury computed by safety verification logic along MP44 on AZ-83:** We computed the probability of serious injury (column 4 in table 6.6) at each speed between 45-60MPH using safety verification logic and multiplied it with the probability distribution function (column 5 in table 6.6). We added up all the resulting values to determine the probability of serious injury along MP44.
- **Probability of serious injury as per [41]:** Authors in [41] report that the total number of accidents during 2002-2008 along MP 44 were 240. Out of these 10 per cent of accidents were due to pick up trucks, 14 percent of these accidents were serious in nature(incapacitated injury, fatal injury etc), 74 percent of these accidents were due to speeding. Multiplying all these values gives the total number of accidents due to speeding and by a pickup truck leading to serous passenger injury as 3.12, dividing this value by 240 gives the probability which is 0.013.

The probability of serious injury to passengers traveling in a pick up truck as computed by safety verification logic at MP44 on AZ-83 is 0.014, whereas the probability of serious injury

Table 6.6: Probability of serious injury at different speeds computed by Safety Verification Logic

Impact Velocity (m/s)	Final Velocity (m/s)	Difference of Final and Impact Velocity (m/s)	Probability of Serious Injury	Probability Of AV having this speed
27(60MPH)	24.7	2.3	0.023	0.044
26.5 (59MPH)	24.5	2.0	0.023	0.050
26.1 (58MPH)	23.8	2.3	0.023	0.053
25.7 (57MPH)	23.3	2.4	0.024	0.056
25.2(56MPH)	23.5	1.7	0.022	0.058
24.7 (55MPH)	22.7	2.0	0.023	0.058
24.3(54MPH)	22.5	1.8	0.022	0.058
23.8(53MPH)	22	1.85	0.022	0.056
23.4(52 MPH)	21.6	1.8	0.022	0.053
22.9 (51MPH)	22.5	0.4	0.020	0.049
22.5(50MPH)	21	1.5	0.021	0.044
22.0(49MPH)	20.2	1.85	0.022	0.040
21.6(48MPH)	21.6	0	0.017	0.034
21.1(47MPH)	21.15	0	0.017	0.024
20.7(46MPH)	20.7	0	0.017	0.020

to passengers given that there is an accident is 0.013 as per [41].

Chapter 7

Discussion

The modeling constructs proposed in this thesis have benefits such as i) application to complex scenarios, and ii)usability

- *Applicability to complex scenarios*: The modeling constructs can be used by system designers to model the behavior of heterogeneous AVs along various road segments (e.g., curve roads, and straight roads) and under various disturbances (e.g., presence of ice and high speed winds). This is achieved by allowing AV's behavior, physical world characteristics, and CPIs to be modeled in a modular manner (as shown in Figure 7.1). AVs are abstracted into different types (e.g., sedan, pickup-truck, coupe and trailer) based on their characteristics (e.g., safety attributes, sensor errors, and control algorithms). Road geometry and external disturbances are abstracted as part of the physical world. AV behavior is abstracted based on its type, physical world, and the CPIs (abstracted by the IROm and UIROm constructs). These abstractions can then be instantiated based on the scenario to be analyzed. For example, if the scenario includes fifty coupes, then fifty instances of coupe behavior can be created as part of the model (without requiring repeated definition of the vehicles). Further, combining simple scenarios can yield more complex ones. For example, CPS models of sedans on a curved road can be combined with coupe models moving on the straight highway to model a scenario where sedans are entering a highway using ramps and coupes are already moving on it.

- *Usability*: The CPS modeling constructs capture the semantics of planned and unplanned motions using intuitive and well defined constructs (i.e IROm and UIROM). As such, the modeling constructs and the verification algorithm can be applied by the system engineers to perform safety verification without requiring any specific analytical expertise. Further, the CPS modeling constructs are generic in nature and can be applied to specify different types of control behavior or even in different domains.

In this thesis, CPS modeling constructs have been applied to BSN and AV's for doing safety verification. The modeling abstractions and safety verification algorithm are

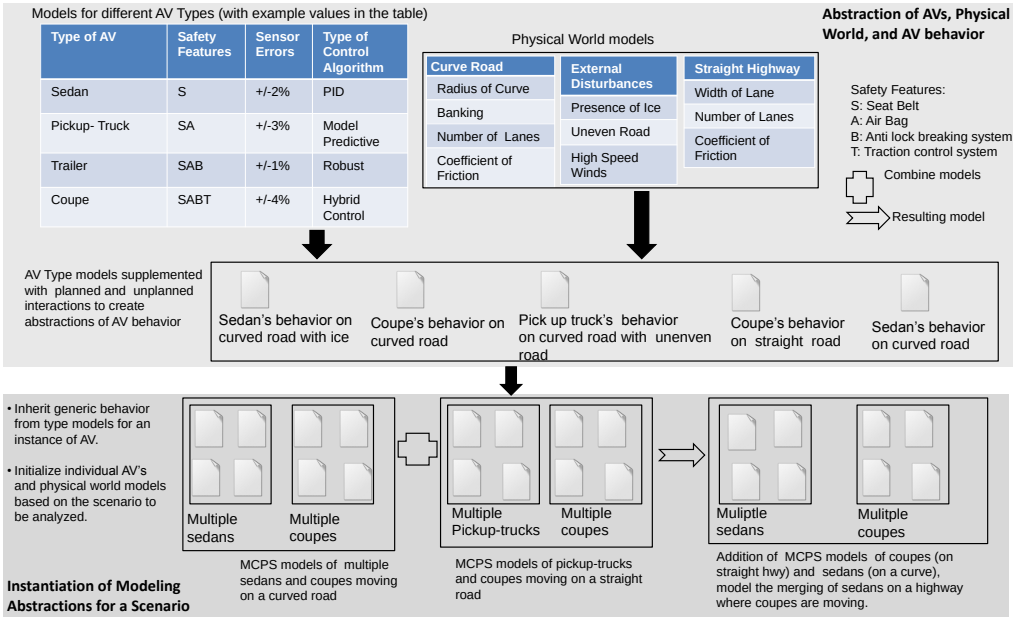


Figure 7.1: Modeling abstractions and their instantiation for specific scenario.

however generic and can be applied to diverse domains such as first responder applications, e.g., a building under fire. In such a scenario, first responders use mobile ad hoc network infrastructure to communicate among themselves as well as use location information provided by fire sensors to identify severe damages. Due to extreme temperatures inside the building, there can be localization errors [17] as a result, fire fighters can often be redirected to incorrect locations thereby delaying the help to disaster victims. We can model the incorrect locations and the behavior of sensor under extreme temperatures using **Unintended Region Of Mobility (UIRom)** and **Physical Process** constructs. The safety criteria will be a threshold based on victim's physiological state under smoke asphyxiation.

Chapter 8

Conclusion and Future Work

In this thesis, i have proposed modeling constructs and safety verification algorithm that allows system engineers to model the behavior of a sytem from CPS perspective and analyze the safety of physical system. The modeling constructs allows specification of system architectural abstraction of various components in CPSs and the inter-dependencies of these components. These abstractions are further modular in nature that can capture the semantics of intended, unintended, planned and unplanned interactions from the computing units (e.g., BSN and AVs) to the physical world. Instantiation of the modularized abstractions for specific scenarios yields representation of corresponding architectural models. The abstraction along with its instantiation allows for modeling complex scenarios with heterogeneous computing units. We demonstrated the use of modeling constructs and verification algorithm for an AV moving along a horizontal curve on AZ-83 highway and BSN performing computing and communication operations.

The current work can be extended in multiple ways. Firstly, the formal semantics of the constructs needs to be developed. Formal semantics can be primarily specified in three ways, i) denotational semantics, ii) axiomatic semantics, and iii) structural operational semantics (SOS). In denotational semantics, information about a construct is passed as arguments to a function which returns it's semantic value on execution. In axiomatic semantics, formal semantics are defined using *Hoare triples* [27] on language constructs. In SOS, formal semantics are defined using using *Labelled Transition Systems (LTS)*. An LTS consists of a set of states and a transition relation between them. States intuitively represent language constructs whereas transitions are based on the abstract syntax of the language. More information on applying SOS to modeling languages can be found at [46] and references with in it. A brief survey of several other techniques published in literature for specifying syntax and semantics of modeling languages can be found at [30]. Secondly, the modeling constructs and safety verification logic needs to be applied to other domains such as mobile ad hoc networks to determine the generality of the constructs. Thirdly, model transformation techniques for converting architectural models specified in AADL to formal models in Spatio-Temporal hybrid automata needs to be developed. This transformation

will help in performing formal safety verification using reachability analysis [25].

REFERENCES

[1]Actual speeds on the road compared to posted speed limits, final report 551, october 2004.

[2]Architecture analysis and design language. http://www.aadl.info/aadl/currentsite/.

[3]Engineers handbook. http://www.engineershandbook.com/Tables/frictioncoefficients.htm.

[4]Fluent, a cfd simulation tool. http://www.fluent.com/.

[5]Lidar. http://wiki.gis.com/wiki/index.php/LIDAR.

[6]Lsdyna. http://www.lstc.com/lsdyna.htm.

[7]Modelica. http://www.modelica.org/.

[8]The montana toolset for formal analysis of aadl specifications. http://aadl.sei.cmu.edu/aadlinfosite/LinkedDocuments/20050127CharonFremont.pdf.

[9]National crash analysis center at george washington university. http://www.ncac.gwu.edu/vml/models.html.

[10]Shoulder width published by us department of federal highway administration. http://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_shoulderwidth.htm.

[11]Simulink. http://www.mathworks.com/products/simulink/.

[12]Sysml,a general purpose modeling language for systems engineering applications. http://www.sysml.org/.

[13] The uml profile for marte: Modeling and analysis of real-time and embedded systems. http://www.omgmarte.org/.

[14] Volume 7:NCHRP Report 500, A Guide for reducing collisions on horizontal curves, 2007.

[15] M. Althoff, D. Althoff, D. Wollherr, and M. Buss. Safety verification of autonomous vehicles for coordinated evasive maneuvers. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1078 –1083, june 2010.

[16] D. Avresky, F. Lombardi, K. Grosspietsch, and B. Johnson. Fault-tolerant embedded systems. *Micro, IEEE*, 21(5):12 –15, sep. 2001.

[17] K. Bannister, G. Giogetti, and S. K. Gupta. less Sensor Networking for Hot Applications: Effects of Temperature on Signal Strength, Data Collection and Localization. In *Hotmnets '08*, 2008.

[18] A. M. Bedford and W. Fowler. *Engineering Mechanics: Statics and Dynamics(5th Edition)*. Prentice Hall, 2008.

[19] B. Berthomieu, J.-P. Bodeveix, S. Dal Zilio, P. Dissaux, M. Filali, S. Heim, P. Gaufillet, and F. Vernadat. Formal Verification of AADL models with Fiacre and Tina. In *ERTSS 2010 – 5th International Congress and Exhibition on Embedded Real-Time Software and Systems*, may 2010.

[20] M. Biehl, C. DeJiu, and M. Törngren. Integrating safety analysis into the model-based development toolchain of automotive embedded systems. In *LCTES '10: Proceedings of the ACM SIGPLAN/SIGBED 2010 conference on Languages, compilers, and tools for embedded systems*, pages 125–132, New York, NY, USA, 2010. ACM.

[21] J. Cannon. The one-dimensional heat equation.

[22] D. G. M. Greenhalgh, M. B. R. Lawless, B. B. Chew, W. A. Crone, M. E. Fein, and T. L. M. Palmieri. Temperature threshold for burn injury: An oximeter safety study. *Journal of Burn Care and Rehabilitation*, 25(5):411–415, 2004.

[23] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Procs. of IEEE HICSS '00-Volume 8*, page 8020, Washington, DC, USA.

[24] F. C. J. Henriques and A. R. Moritz. Studies of thermal injury: I. the conduction of heat to and through skin and the temperatures attained therein. a theoretical and an experimental investigation. In *Am J Pathol.*, pages 530–549, July. 1947.

[25] T. A. Henzinger and V. Rusu. Reachability verification for hybrid automata. Technical Report UCB/ERL M98/19, EECS Department, University of California, Berkeley, 1998.

[26] S. T. Hiroaki, H. Mori, and S. Kato. A lateral control algorithm for vision-based vehicles with a moving target in the field of view. In *in the Field of View,Ť in IEEE International Conference on Intelligent Vehicles*, pages 41–45, 1998.

[27] C. A. R. Hoare. An axiomatic basis for computer programming. *COMMUNICATIONS OF THE ACM*, 12(10):576–580, 1969.

[28] J. Hwang, Y. Gu, T. He, and Y. Kim. Realistic sensing area modeling. pages 2421 –2425, may. 2007.

[29] M. P. V. Jeremy Gillula, Haomiao Huang and C. J. Tomlin. Design and analysis of hybrid systems with applications to robotic aerial vehicle. In *In the Proceedings of the 14th International Symposium of Robotics Research, Lucerne, Switzerland*, 2009.

[30] A. Kamandi and J. Habibi. A survey of syntax and semantics frameworks of modeling languages. In *Computer Science and its Applications, 2009. CSA '09. 2nd International Conference on*, pages 1 –6, dec. 2009.

[31] G. Karsai and J. Sztipanovits. Model-integrated development of cyber-physical systems. In U. Brinkschulte, T. Givargis, and S. Russo, editors, *Software Technologies for Embedded and Ubiquitous Systems*, volume 5287 of *Lecture Notes in Computer Science*, pages 46–54. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-87785-1_5.

[32] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii. Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *Intelligent Transportation Systems, IEEE Transactions on*, 3(3):155 – 161, sep. 2002.

[33] G.-Z. Liu, L. Wang, and Y.-T. Zhang. A robust closed-loop control algorithm for mean arterial blood pressure regulation. *Wearable and Implantable Body Sensor Networks, International Workshop on*, 0:77–81, 2009.

[34] G. Madl and S. Abdelwahed. Model-based analysis of distributed real-time embedded system composition. In *EMSOFT '05: Proceedings of the 5th ACM international conference on Embedded software*, pages 371–374, New York, NY, USA, 2005. ACM.

[35] T. Mukherjee and K. S. Gupta. Cret: a crisis response evaluation tool to improve crisis preparednesss. In *In IEEE International Conference on Technologies for Homeland Security (HST) (To Appear). IEEE Computer*. Society Press, 2009.

[36] V. Prasad, T. Yan, P. Jayachandran, Z. Li, S. Son, J. Stankovic, J. Hansson, and T. Abdelzaher. Andes: An analysis-based design tool for wireless sensor networks. pages 203 –213, dec. 2007.

[37] A. Rajhans, S.-W. Cheng, B. R. Schmerl, D. Garlan, B. H. Krogh, C. Agbi, and A. Bhave. An architectural approach to the design and analysis of cyber-physical systems. *ECEASST*, 21, 2009.

[38] A.-E. Rugina, K. Kanoun, and M. Kaâniche. A system dependability modeling framework using aadl and gspns. pages 14–38, 2007.

[39] L. Sha and J. Meseguer. Design of complex cyber physical systems with formalized architectural patterns. pages 92–100, 2008.

[40] Q. Tang, N. Tummala, S. Gupta, and L. Schwiebert. Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue. *Biomedical Engineering, IEEE Transactions on*, 52(7):1285–1294, July 2005.

[41] T. Tech. Az-83 roadway assessment report,rosemont copper project, 2009.

[42] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, 23(9):661–692, 2006.

[43] K. Venkatasubramanian, A. Banerjee, and S. Gupta. Ekg-based key agreement in body sensor networks. In *In Proc. of 2nd Mission Critical Networks Workshop, IEEE Infocom Workshops*, Apr. 2008.

[44] A. Wardzinski. Safety assurance strategies for autonomous vehicles. In M. Harrison and M.-A. Sujan, editors, *Computer Safety, Reliability, and Security*, volume 5219 of *Lecture Notes in Computer Science*, pages 277–290. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-87698-4_24.

[45] S. Weininger, J. Pfefer, and I. Chang. Factors to consider in a risk analysis for safe surface temperature. In *IEEE Symposium on Product Safety Engineering, 2005*, pages 83–91, Oct.

[46] T. Wolterink. Operational semantics appliedto model driven engineering, 2009.

[47] X. Zhao, J. Guo, H. Yu, and H. Hu. Communication range of virtual node based on cooperative communication. pages 1395 –1400, aug. 2007.