From Verified Models to Verified Code for Safe Medical Devices

Zhihao Jiang

A DISSERTATION

in

Computer and Infomation Science

Presented to the Faculties of the University of Pennsylvania
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

2016

Rahul Mangharam, Associate Professor of Electrical and Systems Engineering
Supervisor of Dissertation

Rajeev Alur, Zisman Professor of Computer and Information Science
Graduate Group Chairperson

Dissertation Committee:
Insup Lee, Cecilia Fitler Moore Professor of Computer and Information Science
Pieter J. Mosterman, Adjunct Professor at the School of Computer Science at McGill University
Richard Gray, Biomedical Engineer at FDA

From Verified Models to Verified Code for Safe Medical Devices

To the universe.

Acknowledgments

Thank you, God bless you; and may God bless the United States of America

ABSTRACT

From Verified Models to Verified Code for Safe Medical Devices

Zhihao Jiang
Rahul Mangharam

Contents

List of Tables

List of Figures

Chapter 1

Medical Devices: Current State and Challenges

The medical device market is worth $289 billion, of which $110 billion is from the US alone, with this number projected to reach $133 billion in 2016. Examples include everything from adhesive bandages, stents, artificial joints, drug infusion pumps to surgical robots, implantable cardiac pacemakers, and devices still undergoing basic research like the artificial pancreas. To take one example of the societal impact of medical devices, an estimated 3 million people worldwide have implanted cardiac pacemakers (a heart rate adjustment device), with 600,000 added annually. Clinical trials have presented evidence that patients implanted with cardiac defibrillators (another heart rate adjustment device) have a mortality rate reduced by up to 31%. Implanted cardiac pacemakers and defibrillators have approximately 80,000-100,000 lines of software code which essentially makes all sensing, control and actuation decisions autonomously within the human body, over the 5-7 year device lifetime [1]. With the increasing complexity of combining hardware and software in a large class of these life-saving technologies, there is an urgent need for approaches to rigorously validate the device and therapy to be safe and efficacious.

The US Food and Drug Administration defines a medical device as an instrument, apparatus, implement, machine, or implant which is:

- intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in humans or other animals, or
- intended to affect the structure or any function of the human body or other animals, and which does not achieve any of its primary intended purposes through chemical action and which is not dependent upon being metabolized for the achievement of any of its primary intended purposes."

[1] Paul L. Jones. Senior Systems/Software Engineer, Office of Science and Engineering Laboratories, U. S. FDA. Personal communication, 2010.

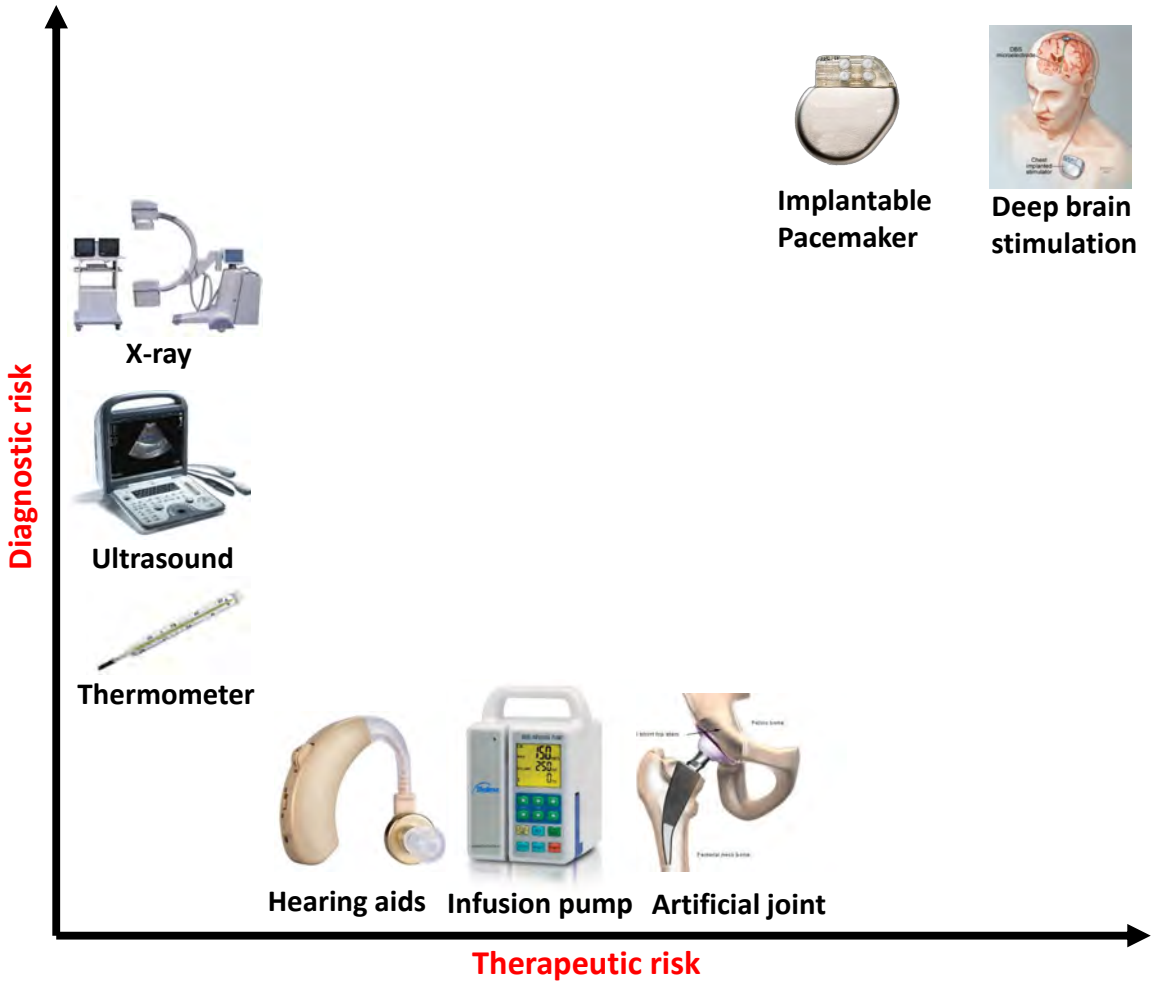


Figure 1.1: Current medical devices across a range of diagnostic and therapeutic risk. Implantable software-controlled devices such as the pacemaker and defibrillator which operate in a closed-loop of sensing, control and actuation are amongst the highest risk

In general, medical devices are categorized according to their risk factors - Class I, Class II and Class III, corresponding to low-risk, medium-risk and high-risk devices (Food and Administration [2014]). Fig. 1.1 gives an intuitive description of medical devices examples across a range of diagnostic and therapeutic risk.

1.1 Closing the Device-Patient Loop

Medical devices operate across a range of invasiveness and intervention with the patient in the loop. For diagnostic-only devices, like an X-ray machine, the physician operates the device to obtain patient data. Upon interpretation of the data, the physician performs diagnosis followed by delivery of proper therapy to the patient (Fig. 1.2.(a)). For therapy-only devices, e.g. a drug infusion pump, the physician

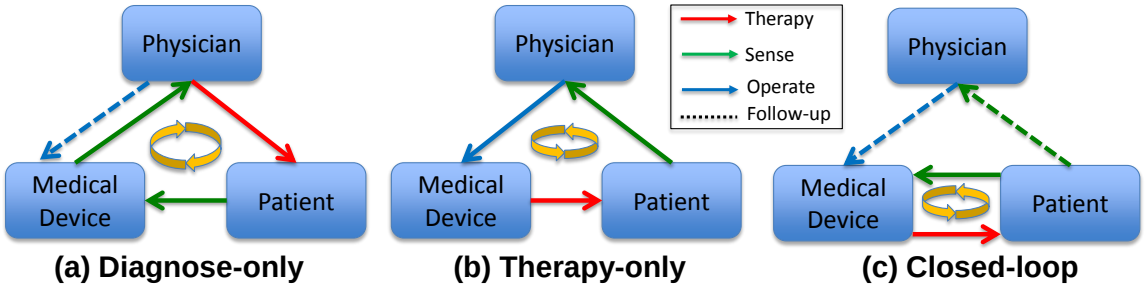


Figure 1.2: Diagnostic-only and therapy-only devices do not interact with the patient in direct closed-loop. The physician is responsible for the diagnostic and/or therapeutic decisions. However in closed-loop medical devices, the devices interact with the patient in closed-loop and have to make therapeutic decisions based on their own diagnosis.

configures the device infrequently based on prior diagnosis of the patient so the device executes the therapy on the patient (Fig. 1.2.(b)). We denote these devices as **Open-loop Medical Devices** as there is no direct feedback loop between the patient and the device. For open-loop devices, the device operates under the supervision of professionally-trained physicians. The device's safety is mostly determined by how accurately it provides information to the physicians or how faithfully it operates as instructed by the physicians.

There is a class of devices with both diagnostic and therapeutic functions, i.e. implantable cardiac devices to treat cardiac arrhythmia, deep brain stimulation devices (Coffey [2009]) to treat Parkinson's disease and artificial pancreas to treat Type-1 diabetes. These devices capture and diagnose the patient's physiological conditions from sensory data, *and* deliver therapy in response (Fig. 1.2.(c)). These devices usually operate (semi-) autonomously with very little human intervention. The benefits of closed-loop medical devices are timely therapy and better life-style. However, autonomy of these medical devices also arose safety concerns. Malfunctions or inappropriate therapies from these devices also cannot be corrected timely, which can cause serious adverse effects on patients' health. With open-loop medical devices, the diagnosis and therapy decisions are made by medical professionals, who have expert knowledge of human physiology. Therefore they are able to identify adverse health conditions and adjust the therapy accordingly. On the other hand, closed-loop medical devices have to make both the diagnosis and therapy decisions on their own. The domain expertise required to make those decisions has to be programmed into the device. It is impossible to encode all the knowledge of human physiology into the device. Therefore, for unanticipated physiological conditions, when the appropriate response has not been programmed into the device, the device may deliver inappropriate therapy which can have an adverse effect on patient's health. Therefore, these devices are usually classified into the highest risk category and undergo the most stringent regulation. We denote them as **Closed-loop Medical Devices**.

There are multiple challenges to develop safe and effective closed-loop medical devices:

1.1.1 Physiological Complexity:

Human physiology is complex and only partially understood. For instance, the functionality of the heart can be interpreted from *multiple perspectives*: from its electrical activity, mechanical contractions of the heart muscles, and dynamics of blood flow. The physiology of the heart can also be analyzed with *multiple scales*: from the molecular level to cellular level all the way to the organ and system level. It is impossible to encode all these contexts into the device, hence inappropriate diagnosis and therapy are observed due to the lack of physiological contexts Sandler et al. [2010], Hauser and Maron [2005].

1.1.2 Physiological Variability:

Physiological conditions and parameters demonstrate different levels of variability both within the individual at different times, levels of exertion and under the influence of medication and also across individuals. For instance, a segment of the population may have additional conduction pathways within their heart, which makes them vulnerable to certain heart diseases. Consequently, autonomous medical devices should be able to safely operate under a large variety of physiological conditions. This is difficult to guarantee, as the device designer must consider all possible physiological conditions (including rare conditions) during the design of the device.

1.1.3 Limited Observability:

Autonomous medical devices normally rely on minimally invasive measurement of the physiological parameters in order to allow the patients to live their normal life. For example, implantable pacemakers and defibrillators commonly have just two leads and therefore two points of observation for the whole heart. The limited observability inevitably leads to ambiguities as different physiological conditions can map to the same input sequence to the device, resulting in inaccurate diagnosis.

1.1.4 Software-related Medical Device Recalls

Due to the complexity of the diagnostic and therapeutic functions of the closed-loop devices, these functions are mostly controlled by their software components. Software embedded in a medical device, unlike electrical and mechanical components, does not fail due to corrosion, fatigue or have statistical failures of subcomponents. Software failures are uniquely sourced in the design and development of the system. According to the US Food and Drug Administration, in 1996, 10% of all medical device recalls were caused by software-related issues (Maisel et al. [2001]). This percentage rose to an average of 15% of recalls from 2008 to 2012 (Fig. 1.3). Malfunctions of closed-loop medical devices usually have severe consequences, which will be categorized as *Class I*, meaning there is a "reasonable prob-

	Software change control	Software Design	Software design manufacturing process	Sum	% of all CDRH recalls
2008	13	141	2	**156**	**18.3%**
2009	9	111	1	**121**	**15.4%**
2010	4	73	3	**80**	**8.9%**
2011	11	182	10	**203**	**15.8%**
2012	12	169	5	**186**	**15.5%**
Sum	**49**	**676**	**21**	**746**	**15.1%**

Figure 1.3: Medical device recalls due to software issues have risen from 10% in the 1990s to $\tilde{1}5\%$ in the past decade (Food and Administration [2012])

ability that use of these products will cause serious adverse health consequences or death." (Food and Administration [2006], Zhang et al. [2015], Sandler et al. [2010]).

1.2 Medical Device Regulation Efforts and Challenges

The medical device industry is regulated to ensure the safety of the patients and the public. In the United States, the FDA is the primary regulatory authority responsible for assuring the safety, efficacy and security of patients using medical devices.

Based on the rationale that 1) manufacturers know their devices better than the regulator, and 2) the variety of medical devices requires a variety of approaches, it is the device manufacturers' responsibility to demonstrate the safety and efficacy of the medical devices. Manufacturers are required to complete a pre-market submission before the devices can be released to the market. The level of requirements for the submission is determined by the safety classification of the devices. A set of general guidelines are recommended by the FDA (Food and Administration [1997, 2002, 2005]) which list the activities that need to be performed to ensure device safety.

In safety-critical industries such as automotive electronics, avionics and nuclear systems, international standards are **enforced** for software system development, evaluation, manufacturing and post-market changes (Fürst et al. [2009], Feiler et al. [2010]). This awareness is only beginning to enter the medical device industry as compliance with international standards are "recommended" in the aforementioned guidelines (Jetley et al. [2006]) but the burden of their interpretation and enforcement is on the device manufacturer. The basic rationale behind these standards is that: if all the risks/hazards of the device are identified and reasonably mitigated, and the device is developed with rigorous process, the device is *reasonably safe*.

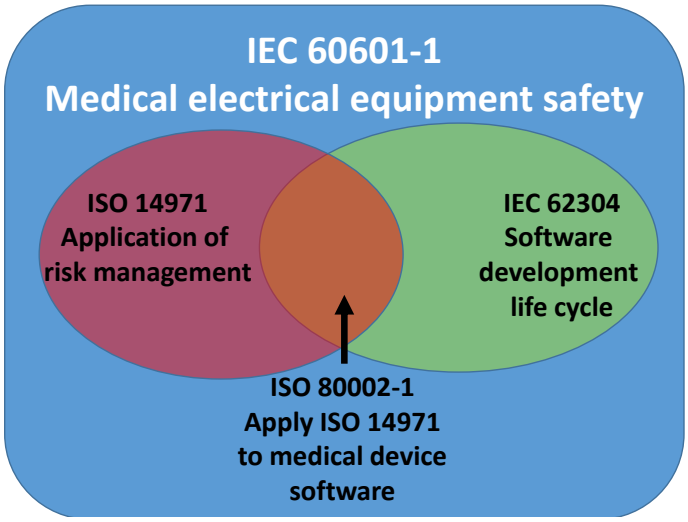


Figure 1.4: International standards for medical device safety. These standards define the required activities during the development process.

Fig. 1.4 describes the primary standards to ensure medical device safety and their relationships. The IEC 60601 Medical Electrical Equipment - General requirements for basic safety and essential performance is a product safety standard that all electronic medical devices must comply to. IEC 60324 specifies the processes and activities needed to perform during the software development life cycle to ensure software safety.

Risk management is a core activity throughout the software development life cycle. ISO 14971 is specified for the application of risk management to medical devices. In addition, for each risk management activity of ISO 14971, ISO 80002-1 provides additional guidelines for the software component, which highlights and explains approaches to assuring that software safety is adequately addressed.

1.2.1 Pre-Market Evaluation with Clinical Trials

Regardless of how rigorous the risk management and the device development process are, the devices have to be able to achieve their design goal on the real patient, which can only be evaluated within its physiological environment. Devices that have high risk factors, including the closed-loop medical devices, are required to submit clinical evidence for their safety and efficacy, often in form of clinical trials. In clinical trials, the devices are used on a preselected population of patients following carefully-designed protocols. The goal of a medical trial, in part, is to obtain unambiguous results for the primary question of the trial which can support the safety and/or efficacy of the devices. However, conducting clinical trials is very time consuming and expensive, and risks found during clinical trials are very expensive to fix (U. S. Food and Drug Administration [2013]).

To address this **safety gap** between ensuring the device satisfies its therapeutic

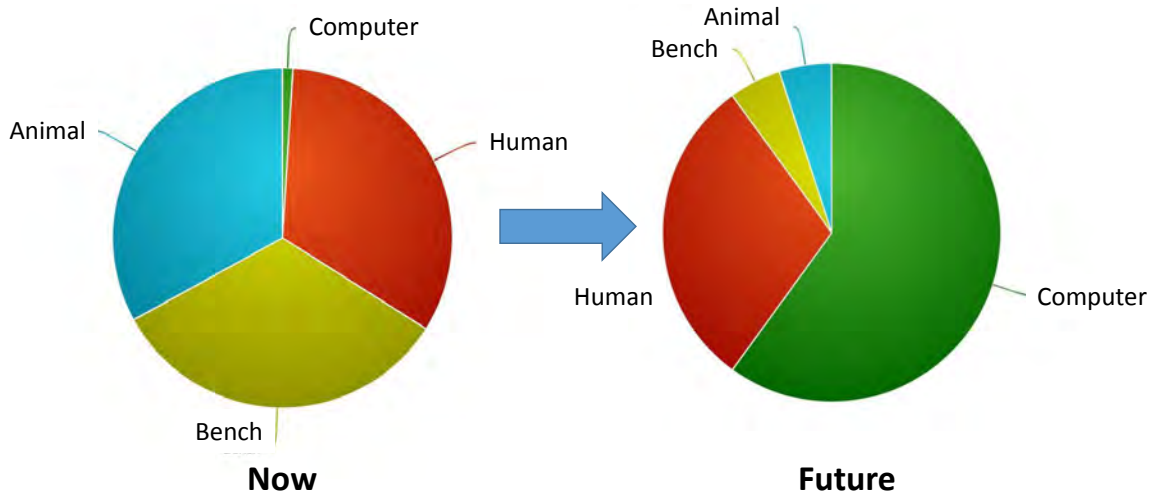


Figure 1.5: Percentage of computer simulation is expected to increase as safety and effectiveness evidence of medical devices

requirements with the patient-in-the-loop and testing its software specifications, new approaches for closed-loop validation of the device software within the physiological context are needed - this is the primary focus of this article.

1.3 Model-based design to improve medical device safety

With the deluge of software-based closed-loop medical devices in the coming years, relying on clinical trials as the only closed-loop evaluation method to identify risks rooted in device software is not scalable. Model-based design and virtual integration have been proposed and applied in other industries like automotive and avionics (Fürst et al. [2009], Feiler et al. [2010]), and can potentially help during the development process and provide extra confidence to the device before conducting clinical trials. However, unlike man-made systems like automobiles and aircrafts, physiological systems are less understood with larger variations for the type and degree of patient conditions. The lack of faithful models of physiological environment of the closed-loop medical devices is one of the reason that model-based design is not well-adopted in the medical device industry.

As computational models of human physiology are developed, they can be used to interact with closed-loop medical devices or their models. The FDA is starting to recognize in-silico modeling and simulation as regulatory-grade evidence for device safety and efficacy. For example, Ghorbani and Bogdan [2013] developed glucose-insulin models that can be used to evaluate control algorithms for artificial pancreas devices which can sense blood glucose and deliver insulin. Simulation results with the models have been recognized by FDA to replace animal trials, in part, which

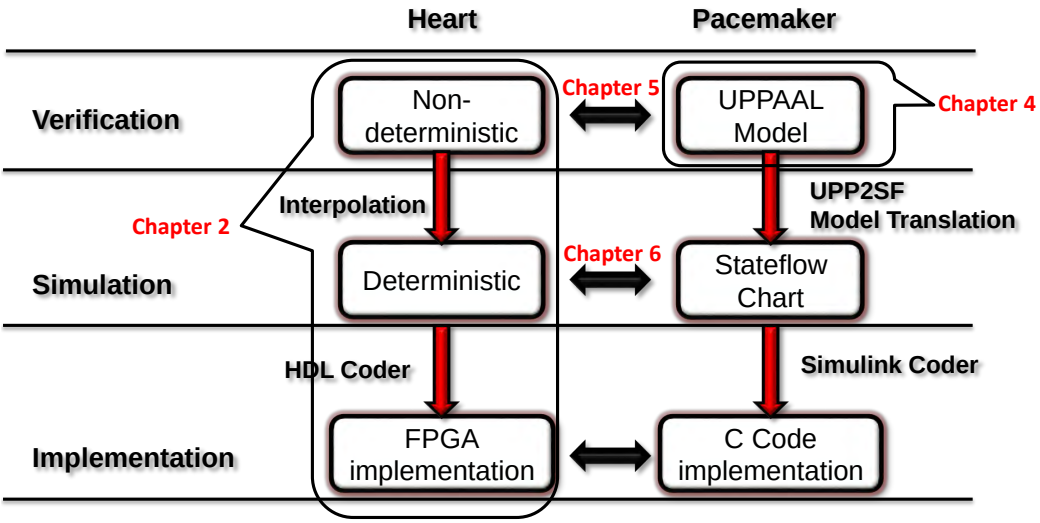


Figure 1.6: Model-driven design for verified models to verified code for the closed-loop heart and pacemaker system

significantly reduced cost (B. P. Kovatchev and M. Breton and C. Dalla Man and C. Cobelli [2009]). With the increasing interest and recognition from the regulators, computer models and simulations are expected to play bigger role as as "regulatory-grade evidence" evidence in the development of future closed-loop medical devices (Fig. 1.5).

1.4 Contributions

In this thesis, I use implantable cardiac devices as working examples to demonstrate how model-based approaches can help improve the safety and efficacy of autonomous medical device. We demonstrate the application of model-based approaches in several design and pre-regulation activities, from the perspective of the manufacturer's design validation team. We assume availability of design artifacts including pacemaker design and physiological requirements. By demonstrating the process of developing verified models to generate verified code, the results of our model-based closed-loop validation should be able to support the device's safety and efficacy requirements during the regulation process.

In this thesis, I propose a model-based design framework for closed-loop validation of medical devices, and use implantable cardiac devices as case study. Fig. **??** demonstrates our model-based design framework for implantable pacemaker software. The thesis can be broken down into 4 themes, which are illustrated in Fig. 1.6.

The remaining thesis is arranged as follow: Chapter 2 discusses a dual chamber pacemaker desig n(Boston Scientific Corporation [2007b]) as the motivating example to illustrate the safety hazards that the device may pose to the patients. Chapter 3 discusses the physiological models that are necessary for closed-loop evaluation of

medical devices, and how to use those models to represent complex physiology with large variability Jiang et al. [2012a]. Chapter 4 discusses the use of model-checking techniques to evaluate the safety and efficacy of device design early in the device design stage, with focus on the abstraction and refinement of the heart models to cover large variety of physiological conditions (Jiang et al. [2014]). Chapter 5 discusses the rigorous translation from verified device model to device implementation which maintains the verified properties (Pajic et al. [2012]).. Chapter 6 discusses the use of physiological models to generate a virtual patient population and the device can go through model-based clinical trials which can provide useful insights for planning a clinical trial on real patient.

1.5 Useful terminologies for often misinterpreted terms

Ensuring the safety of complex medical devices has drawn interest not only from stakeholders like regulators and industries, but also medical professionals and academia. Different communities have different interpretations over certain terminologies, often causing misunderstandings. In this paper we adopt the terminologies from the regulation perspective, so that the results we have fit into the regulation framework. Most of the definitions are referred from the FDA guideline document General Principles of Software Validation (Food and Administration [2002]). Below are several terminologies that we use throughout the paper which worth clarifying.

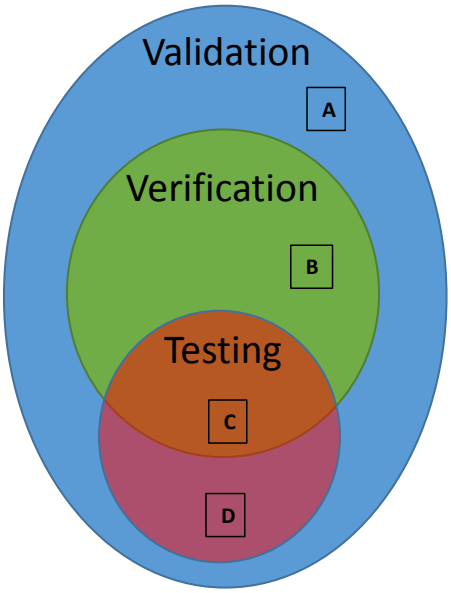
1.5.1 Requirements vs. Specifications

By the definition of FDA (Food and Administration [2005]), the requirements of a system describe **what** the system should achieve and the specifications of a system describe **how** the system is designed to satisfy the requirements. For example, a requirement for an autonomous car is "The car should not hit objects". The corresponding specification can be "brake if the speed of the car is greater than x and the distance to the object is less than y". We can see that a car satisfying its specification may not satisfy the requirement (e.g. when the car is driving too fast or the obstacle pops up right in front of the car). In this paper, we use the word requirement in particular to denote the intended uses of the medical devices to improve physiological conditions.

1.5.2 Validation vs. Verification vs. Testing

As defined in Food and Administration [2002], software validation is the confirmation by examination and provision of objective evidence that:

1. software specifications conform to user needs and intended uses, and



Validation activities	Zone
Planning	A
Requirements	A
Traceability	A,B
Change management	A
User site testing	D
Defect resolution	A,B
Risk management	A
Intended use	A
Evaluations	B
Design reviews	B
System testing	C
Regression testing	C,D

Figure 1.7: Validation activities during the software development life cycle (D A. Vogel [2011])

2. the particular requirements implemented through software can be consistently fulfilled

The first aspect ensures the device is safe and effective. The second aspect maintains the traceability of requirements throughout the development life cycle. Software verification fulfills the second aspect of software validation by "providing objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. "

Testing is the technique that can be used for validation and/or verification. Fig. 6.7 illustrates the relationship between validation, verification and testing, and different activities during the software development life cycle to ensure the safety and effectiveness of the software.

1.5.3 Closed-loop vs. Open-loop Evaluation

In open-loop evaluation, i.e. open-loop testing, input sequences are send to the system and system outputs are compared with expected outputs. In open-loop testing, the system outputs do not affect the inputs afterward. In closed-loop evaluation, the environment of the system is taken into account. System outputs affect the state of the environment and thus affect the input sequences. For closed-loop medical devices, clinical trials are currently the most common closed-loop evaluation method. Enable closed-loop evaluation at model level requires models of the environment, which is

human physiology for closed-loop medical devices.

Closed-loop evaluation accomplishes two goals in model-based design: 1) It enforces environmental constraints so that the test space is smaller and the test cases have physiological relevance. 2) Execution traces can be better interpreted as the physiological models encode domain knowledge.

Chapter 2

A Motivating Example: A Dual Chamber Pacemaker Design

In this chapter we introduce a dual chamber pacemaker specification. It is used throughout the thesis to demonstrate how different model-based techniques can be used to validate the safety and efficacy of the pacemaker.

2.1 Physiology Basis of the Heart and the Pacemaker

First, we use this small section to introduce the physiology basis of the heart and the application of implantable cardiac devices. Readers with knowledge of this subject can skip to the following sections.

2.1.1 Blood Circulation System

The heart is the "motor" for blood circulation within our body. The heart has two ventricles which pump the blood out of the heart, and two atria which gather blood from the body and pump them into the ventricles. (Fig. 2.1.(a)) There are two circulations through the heart: the *Pulmonary circulation* and the *Stemic circulation*. In the pulmonary circulation, the right atrium collects oxygen-depleted blood from all over the body and pumps it into the right ventricle. The right ventricle then pumps low-oxygen blood to the lungs. The blood gets oxygenated in the lungs and gathers into the left ventricle. In the stemic circulation, the oxygenated blood in the left atrium is pumped into the left ventricle. The left ventricle pumps the blood to the rest of the body and the heart itself. After the body extracts the oxygen from the blood and injects carbon dioxide, the oxygen-depleted blood then flows back to the right atrium.

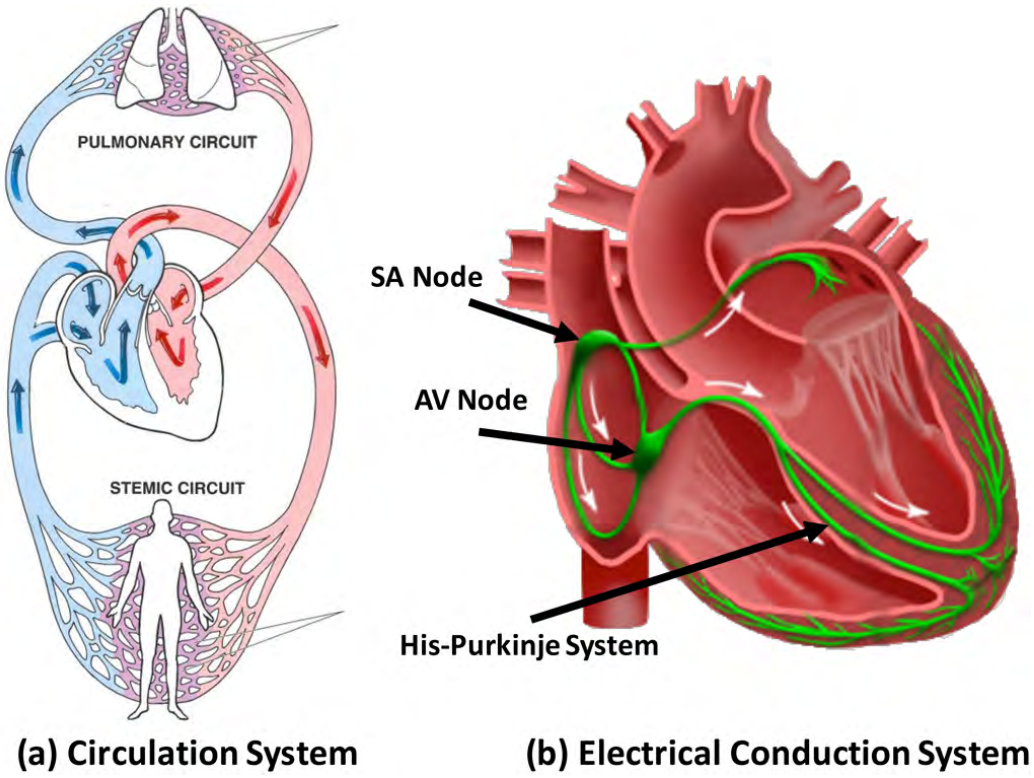


Figure 2.1: (a) The circulation system. (b) Electrical Conduction system of the heart

2.1.2 Electrical Conduction System of the Heart

The oxygen demand of the body changes during different activities. For example, the demand is higher while running and lower while sleeping. To satisfy these demands, the heart muscles in the atria and the ventricles have to contract with certain frequency and in accordance to optimize the *Cardiac Output*, which refers to the volume of blood pumped by the heart per minute (mL blood/min). The coordinated contractions of the heart muscles are governed by the electrical conduction system of the heart (Fig. 2.1.(b)) A *Normal Sinus Rhythm (NSR)* is the healthy heart rhythm which provides efficient blood flow. During a NSR, electrical signals are periodically generated by the *Sinoatrial (SA) node* in the upper right atrium, which acts as the intrinsic pacemaker of the heart. The signals conduct throughout both atria and trigger muscle contractions to push blood into the ventricles. After a long conduction delay at the *AV node* so that both ventricles are fully filled, the signals conduct through fast-conducting *His-Purkinje* system to trigger almost simultaneous contractions of the ventricles and pump blood out of the ventricles.

Derangement from NSR can result in insufficient cardiac output and thus insufficient oxygen supply to the body and/or the heart itself, which are referred to as *Arrythmia*. Arrhythmia impair the heart's ability to efficiently pump blood and compromise the patient's health. Arrhythmia are categorized into so-called *Tachycardia* and *Bradycardia*. Tachycardia features undesirable fast heart rate which can cause

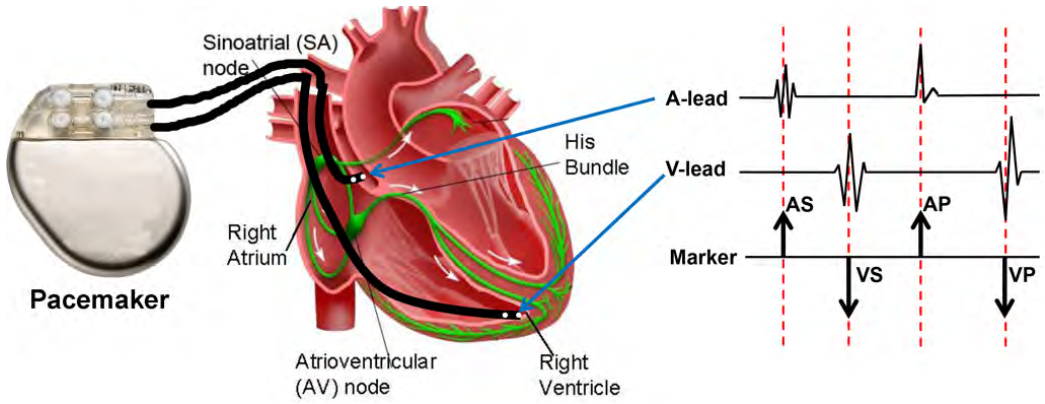


Figure 2.2: (a) Lead placement for a dual chamber pacemaker. (b) Electrogram (EGM) signals measured from pacemaker leads and corresponding internal pacemaker events

inefficient blood pumping. Bradycardia features slow heart rate which results in insufficient blood supply. Bradycardia are due to failure of impulse generation with anomalies in the SA node, or failure of impulse propagation where the conduction from atria to the ventricles is delayed or blocked.

2.1.3 Electrophysiology and Implantable Cardiac Devices

The electrical activities of the heart closely couple with the mechanical contractions thus the electrical activities of the heart can be monitored and used to diagnose arrhythmia. The most well-known method is Electrocardiogram (ECG), which measures the integration of electrical activities of the heart measured along different axis on the body surface. The electrical activities can also be directly measured by inserting electrodes through the vein into the heart. The electrodes are placed against the inside heart wall and localized electrical activities can be measured. Physicians can also deliver pacing sequence through the electrodes to explore the heart conditions. This procedure is referred to as Electrophysiological (EP) Testing (Josephson [2008]) and the signals are referred to as electrograms (EGMs) (Fig. 2.2.b). The timing and morphology of the ECG and EGM signals together are used to diagnose arrhythmia.

The implantable cardiac pacemakers are rhythm management devices designed to treat bradycardia. A typical dual chamber pacemaker has two leads inserted into the heart through the veins which can measure the local electrical activity of the right atrium and right ventricle respectively (Fig. 2.2.a). According to the timing between sensed impulses, the pacemaker may deliver electrical pacing to the corresponding chamber to maintain proper heart rhythm.

2.2 A Dual Chamber Pacemaker Specification

In our study, we focus on the implantable pacemaker, which is one of the simpler implantable cardiac devices. The functionality of a pacemaker is based on the timing of local electrical events, which can be intuitively modeled with timed automata. The specifications are based on the algorithm descriptions from Boston Scientific manuals (Boston Scientific Corporation [2007b]) and the functional description released as part of the Pacemaker Challenge (Boston Scientific Corporation [2007a]).

The pacemaker is designed for patients with bradycardia (i.e. slow heart rate). Two leads, one in the right atrium and one in the right ventricle, are inserted into the heart and fixed onto the inner wall of the heart. These two leads monitors the local activation of the atria and the ventricles, and generate corresponding sensed events (AS, VS) to its software. The software determines the heart condition by measuring time difference between events and delivers pacing events (AP, VP) to the analog circuit when necessary. The analog circuit then delivers pacing signals to the heart to maintain heart rate and A-V synchrony. In order to deal with different heart condition, pacemakers are able to operate in different modes. The modes are labeled using a three character system (e.g. xyz). The first position describes the pacing locations, the second location describes the sensing locations, and the third position describes how the pacemaker software responds to sensing. Here we introduce the widely used DDD mode pacemaker which is a dual chamber mode with sensing and pacing in both atrium and ventricle.

A DDD pacemaker has five basic timing cycles triggered by external and internal events, as shown in Fig. 2.3. We decomposed our pacemaker model into five com-

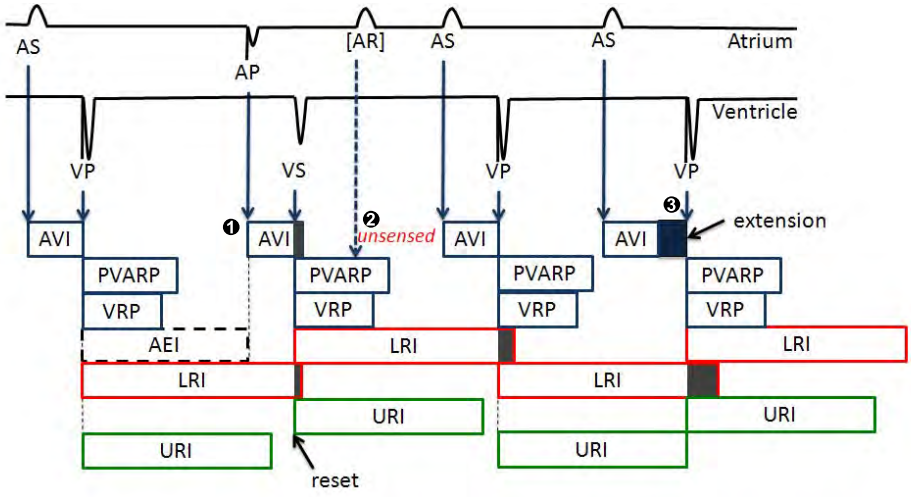


Figure 2.3: Basic 5 timing cycles for a dual chamber pacemaker which include the Lower Rate Interval (LRI), Atrio-Ventricular Interval (AVI), and Upper Rate Interval (URI). Also included are the blanking intervals, Post Ventricular Atrial Refractory Period (PVARP) and Ventricular Refractory Period (VRP), to inhabit action by the pacemaker.

ponents which correspond to the five timers. $P = LRI \| AVI \| URI \| PVARP \| VRP$. These components synchronize with each other using broadcast channels and shared variables (as shown in Fig. 4.1).

2.2.1 Lower Rate Interval (LRI)

The Lower Rate Interval (LRI) component is shown in Fig. 4.1(a). This component defines the longest interval allowed between two ventricular events, thus keeping the heart rate above a minimum value. In DDD mode, the LRI interval is divided into a V-A interval (TLRI-TAVI) and a A-V interval (TAVI). The LRI component maintains a maximum V-A delay while the AVI component maintains a maximum A-V delay so together they maintain the maximum V-V delay. In the LRI component, the clock is reset when a ventricular event (VS, VP) is received. If no atrial event has been sensed (AS), the component will deliver atrial pacing (AP) after TLRI-TAVI.

2.2.2 Atrio-Ventricular Interval (AVI) and Upper Rate Interval (URI)

The function of the AVI component defines the longest interval between an atrial event and a ventricular event. If there is no ventricular event (VS) within TAVI after an atrial event (AS, AP), and the time since the last ventricular event (VS, VP) is longer than TURI, the component will deliver ventricular pacing (VP). The URI limits the ventricular pacing rate by enforcing a lower bound on the times between consecutive ventricle events. The UPPAAL design of AVI and URI component is shown in Fig. 4.1(b) and (c).

2.2.3 Post Ventricular Atrial Refractory Period (PVARP) and Post Ventricular Atrial Blanking (PVAB)

Ventricular events, especially Ventricular Pace (VP) are sometimes so strong that the atrial lead can sense the activation as well. This signal may be falsely recognized as an atrial event and disrupt normal pacemaker function. This scenario is called crosstalk and was discussed in our previous work (Jiang and Mangharam [2011]). In order to prevent this undesired behavior, and filter potential noises, there is a blanking period (PVAB) followed by a refractory period (PVARP) for the atrial events after each ventricular event (VS, VP). Atrial events during PVAB are ignored and atrial events during PVARP trigger AR! events which can be used in some advanced diagnostic algorithms. The UPPAAL design of PVARP component is shown in Fig. 4.1(d).

2.2.4 Ventricular Refractory Period (VRP)

The VRP follows each ventricular event (VP, VS) to filter noise and early events in the ventricular channel which could otherwise cause undesired pacemaker behavior.

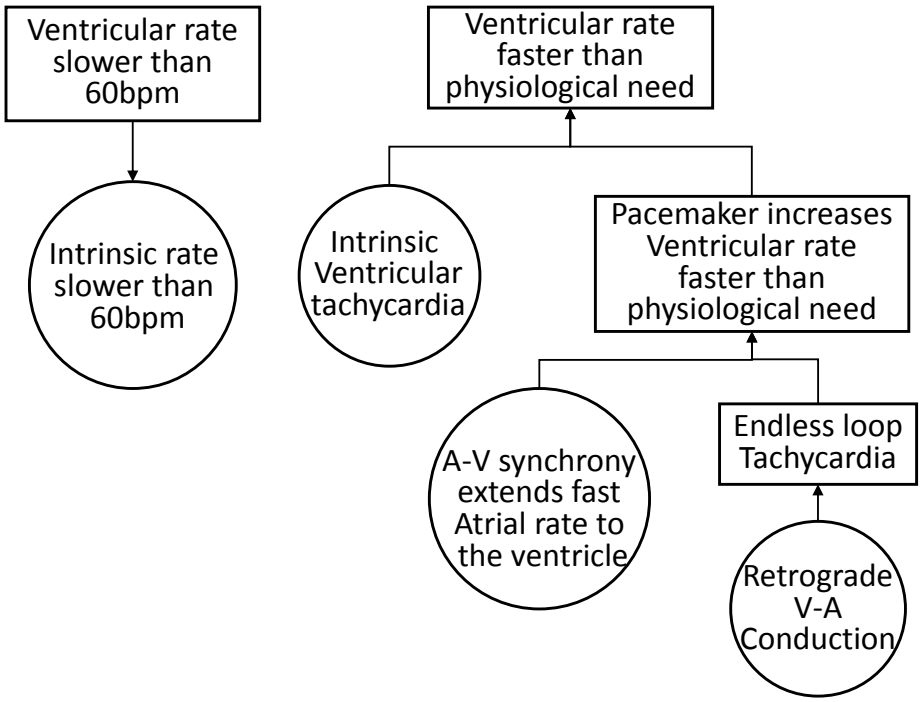


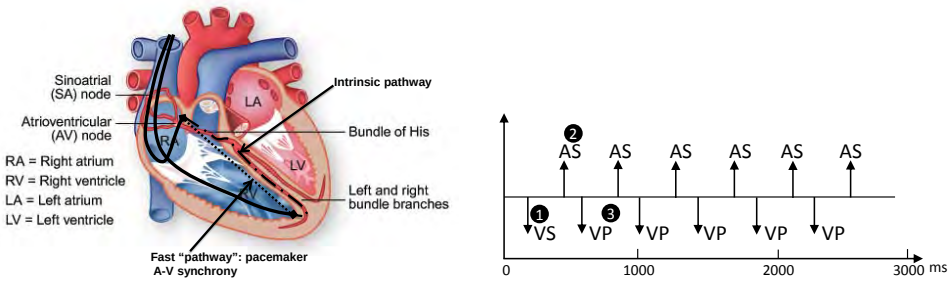
Figure 2.4: Sample Fault Tree Analysis of the physiological conditions leading to the lower rate limit and upper rate limits

2.3 Identify Safety Hazards in the Dual Chamber Pacemaker

Implantable pacemakers are designed to treat bradycardia by increasing the heart rate with external pacing. Therefore the heart rate should not only be increased to the minimum physiological need, but also should not be increased beyond physiological need. Fig. 2.4 demonstrates two Fault Tree Analysis (FTA) for these two top level hazards. In the remaining chapter we first specify hazards as properties, and use model checking to evaluate whether these hazards have been mitigated by the pacemaker. Then for one of the mitigation algorithm, we examine the mitigation effectiveness and the residue hazard.

2.4 Known Safety Hazards of Dual Chamber Pacemakers

In this section we introduce two well-studied safety hazards in a basic dual chamber pacemaker design. Device manufacturers have developed algorithms to mitigate the hazards. In the following chapters, I will use model-based techniques to identified these safety hazards in the early design stage, and evaluate the effectiveness of the mitigation algorithms developed by device manufacturers.



(a) Virtual circuit formed by the pacemaker and the heart

(b) Pacemaker trace for ELT initialized by a early ventricular signal

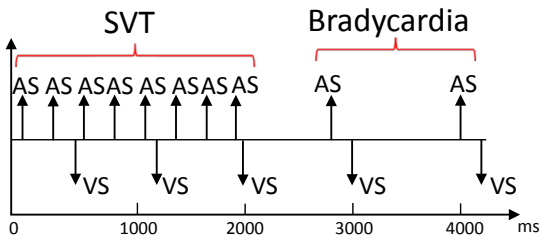
Figure 2.5: Endless Loop Tachycardia case study demonstrating the situation when the pacemaker drives the heart into an unsafe state Jiang et al. [2011]

2.4.1 Endless-Loop Tachycardia

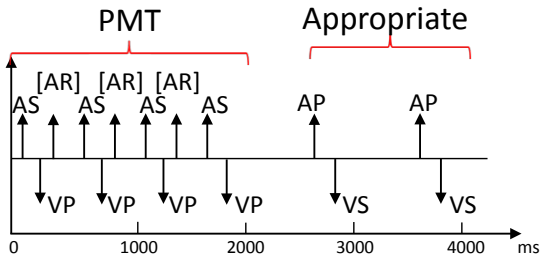
The AVI component of a dual-chamber pacemaker introduces a virtual A-V conduction pathway. This forms a timing loop with the intrinsic (physiological) A-V conduction pathway (see Fig. 2.5(a)). A Premature Ventricular Contraction (PVC), i.e. an early extra beat in the ventricles, may trigger another ventricular event (VS) and initiate a V-A conduction through the intrinsic pathway (Marker 1 in Fig. 2.5(b)). The pacemaker registers this signal as an Atrial Sense (AS) (Marker 2 in Fig. 2.5(b)). This event triggers a VP after TAVI, as if the signal conducts through the "virtual" A-V pathway (Marker 3 in Fig. 2.5(b)). We call it "virtual" pathway as the "conduction" delay is fulfilled by a timer in the pacemaker instead of a physical signal propagation along the heart tissue. The VP will trigger another V-A conduction and this VP-AS-VP-AS looping behavior will continue (see Fig. 2.5(b)). The interval between atrial events is TAVI plus the V-A conduction delay, which is normally shorter than the delay between intrinsic heart beats, thus driving the ventricular rate as high as the Upper Rate Limit. During ELT, the heart rate is not only high, but also fixed, which is an unsafe scenario.

2.4.2 Atrial Tachycardia Response

Supraventricular Tachycardia (SVT) is an arrhythmia with an abnormally fast atrial rate. Typically, in a heart without pacemaker, the AV node, which has a long refractory period, can filter most of the fast atrial activations during SVT, thus the ventricular rate remains relatively normal. Fig. 2.6(a) demonstrates a pacemaker event trace during SVT, with a pacemaker in ODO mode, which just sensing in both channels. As there is no pacing in ODO mode, the heart is in open-loop with the pacemaker. In this particular case, every 3 atrial events (AS) correspond to 1 ventricular event (VS) during SVT. As an arrhythmia, SVT is still considered a safe heart condition since the ventricles operate under normal rate and still maintain adequate cardiac output.



(a)



(b)

Figure 2.6: Benign open loop case: SVT without a pacemaker or with a pacemaker in sense-only mode (ODO) (b) Dangerous closed-loop-case SVT with DDD pacemaker which tries to match the fast atrial rate with a corresponding (and dangerous) fast ventricular rate.

However, in the closed loop case with the DDD pacemaker, the AVI component of a dual chamber pacemaker is equivalent to a virtual pathway in parallel to the intrinsic conduction pathway between the atria and the ventricles. The pacemaker tries to maintain 1:1 A-V conduction and thus increases the ventricular rate inappropriately to match the atrial rate. This is known as Pacemaker Mediated Tachycardia (PMT) as the heart would have been safe without the pacemaker and its virtual pathway. Fig. 2.6(b) shows the pacemaker trace of the same SVT case with DDD pacemaker. Although half of the fast atrial events are filtered by the PVARP period ([AR]s), the DDD pacemaker still drives the closed-loop system into 2:1 A-V conduction with faster ventricular rate. Maintaining A-V delay is less important than maintaining an appropriate ventricular rate. The DDD pacemaker violates a higher priority requirement in order to satisfy a lower priority requirement, which is inappropriate.

2.5 Discussion

Implantable cardiac devices such as implantable pacemakers are typical autonomous medical devices. Although the functionality is relatively simple, pacemakers illustrate the three challenges discussed in the introduction very well. In the following chapters, I will demonstrate the use of different model-based techniques to provide safety and

efficacy confidence to the pacemaker design.

Chapter 3

Theme 1: Modeling the Physiological Environment

Closed-loop medical devices such as the implantable cardiac pacemaker and defibrillator are designed to operate autonomously and interact with the human body to maintain and improve the physiological conditions of the patients. To evaluate the device within the closed-loop context of the human body, the knowledge of the physiological contexts (e.g. patient arrhythmia, physical activity) and the signals by which the device interacts with the organ(s) to manage the condition is essential. By constructing physiological models and encoding physiological requirements, our goal is to evaluate the safety and efficacy of the device therapy across a range of physiological conditions.

Consequently, it is important to model the physiological environment of the device such that details unrelated to the interaction between the device and the human are abstracted away, while essential information required to differentiate different patient conditions are maintained. As we will see, to validate the device operation across a range of physiological conditions and for a set of safety and efficacy properties, a family of models are needed which refine the closed-loop context to appropriately express the condition to be verified.

Models, especially models of the human physiology, which span a large spectrum of scale and complexity, should be designed in accordance with their respective applications. Each application of the environment model has a different focus and has distinct modeling requirements which influence the model complexity and model identifiability.

Model complexity: How much detail should the physiological models have, in order to unambiguously describe a physiological behavior? In particular, if the model checker returns an execution trace as counter-example, how much details should the physiological model have so that the execution traces can be interpreted by domain experts?

Model Identifiability is a metric for the feasibility of identifying model parameters from data. There are two methods for model construction: non-parametric

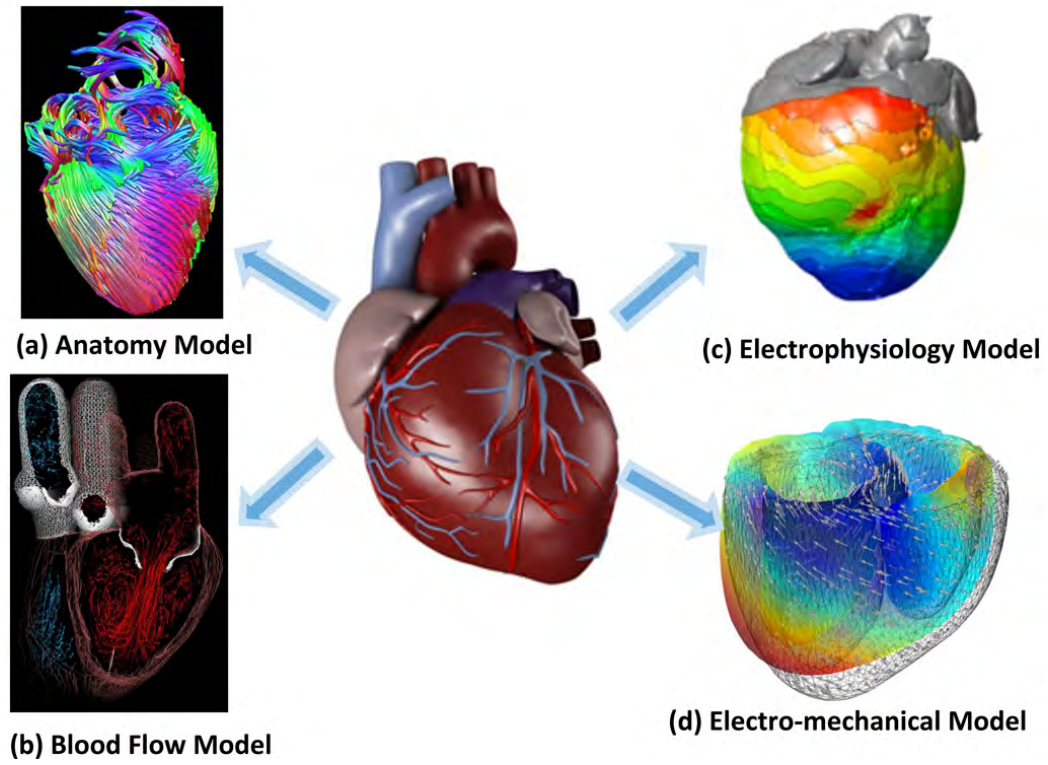


Figure 3.1: Physiological models of the heart from different perspectives

modeling in which no prior knowledge is assumed and the model construction is purely data-driven; and parametric modeling in which domain knowledge of the physiological conditions is taken into account. For example, to model the electrophysiological activity of the heart, there is abundant literature describing the phenomena of individual arrhythmia, which makes parametric modeling of the environment favorable.

In the following sections, we introduce the physiological contexts within which the implantable pacemaker operates, and proceed to construct a heart model structure for closed-loop validation of the pacemaker. Note that for two different applications, i.e. model checking of the device model in the loop and functional testing of the actual device in the loop, models are constructed differently as we address their respective requirements for environment models.

3.1 Physiological Models of the Heart

To study the mechanisms of heart diseases and their effects on cardiac output, different physiological models of the heart have been developed. Fig. 3.1 illustrates several aspects that these models capture. With the development of the imaging techniques like MRI, detailed anatomical structures of the heart can be modeled and studied (Schulte et al. [2001]). These models are fundamental in other modeling aspects as well, as the anatomy of the heart dictates the electrical and mechanical behaviors of

the heart. Fig. 3.1.(a) shows models for heart muscle fiber orientations by E.W. Hsu and C.S. Henriquez [2011]. With anatomy models the electrical and/ or mechanical properties of the heart can be studied. Fig. 3.1.(b) illustrate a model of blood flow within the ventricles (Peskin and McQueen [1989]). Electrical properties of the heart at cellular level has been modeled (Sachse et al. [2008]) and by combining these cellular models with the structural models, the electrical activities of the whole heart are studied, especially the mechanism of different arrhythmia (Trayanova and Boyle [2014], Grosu et al. [2011], Murthy et al. [2013]). Intrinsic heart rate variability has been modeled to synthesize optimal control of pacemaker pacing. (Bogdan et al. [2013]) Abstraction of the electrical cellular model has also been attempted by Islam et al. [2014] to reduce model complexity without sacrificing accuracy. The electrical properties and the mechanical properties of the heart are closely coupled. Models combining both of these aspects are also developed to study the effects of different arrhythmia on cardiac outputs (Trayanova and Boyle [2014], Rossi et al. [2011]).

3.2 EP Heart Model Structure for Closed-loop Validation of Implant-able Cardiac Devices

Models should be developed according to their applications. The aforementioned models of the heart are mostly used for understanding the mechanisms of different heart diseases. Physiological models developed for closed-loop evaluation of medical devices should have the following considerations:

C1. Interfacing with the device: The model should be able to generate physiological signals that the device sense from the real physiological entities. And the model should be able to take device output as input and change its states accordingly. Model complexity should also be adjusted according to the device interface to hide unnecessary details.

C2. Differentiate different physiological conditions: To evaluate the safety and effectiveness of the device, the device has to be evaluated under certain physiological conditions specified by the requirements. For example, the pacemaker is supposed to maintain proper heart rate during Bradycardia. The model should be expressive enough to be able to differentiate the physiological condition (Bradycardia in the example) from other conditions. Failing to do so may result in false-positives or false-negatives in the evaluation result.

C3. Physiological/logical interpretation of model states: In closed-loop evaluation we are checking the device safety and effectiveness against the physiological requirements. However, due to the limited interface (e..g two leads for a dual chamber pacemaker) it is always difficult to determine only from an execution trace that the therapy is safe and effective. Therefore, being able to provide physiological meanings to the states of the model also allows us to interpret the closed-loop execution more accurately, thus reducing the number of physiologically impossible

executions during the evaluation. To satisfy these requirements, the model structure of these physiological models should base on physiological or clinical first principles so that states and state transitions of the closed-loop executions can be explained with physiological language.

C4. Available patient data: In closed-loop evaluation, physiological models are developed to represent certain physiological condition across a population of patients or even a particular patient. The model parameters must be identified so that the behaviors of the models match the behaviors of the patients (groups). Due to the limited sensing capability of closed-loop medical devices, the obtained data is sparse. i.e. we can not put a sensor on every tissue region of the heart. Therefore the complexity of the model should be in accordance with the available data to avoid *over-fitting*, which occurs when a model has too many parameters relative to the number of observations, and this can introduce errors during prediction.

The electrophysiological models mentioned in the last section (Trayanova and Boyle [2014], Grosu et al. [2011]) satisfy C1-C3. However, the parameter space of these models are too large (10+ parameters for each cellular model multiplied by 10^5 of elements) which not only increase simulation complexity, but also impossible to identify due to lack of data. As introduced in Section 2.1.3, the pacemaker has only two leads at fixed locations and only use timing between local activation events for diagnosis. These models with high spatial fidelity possess details that can be abstracted without sacrificing the three considerations.

Electrophysiology testing (EP testing) has been an active clinical field to diagnose and treat arrhythmia with minimal-invasive procedures. During an EP testing procedure, the physicians diagnose heart conditions by examining the patterns and intervals of local electrical activations (temporal) measured from electrodes placed into different locations of the heart (spatial). EP testing is the perfect modeling level for closed-loop evaluation of implantable cardiac devices because: 1) it is the basis of implantable cardiac devices (C1), 2) physicians can use EP testing to diagnose most arrhythmia thus distinguish them (C2,C3), 3) there are abundant patient data available (C4).

In the remaining chapter we will introduce our heart modeling efforts based on EP testing, and model adaptation for two different applications of closed-loop evaluation of implantable cardiac devices.

3.2.1 Modeling Philosophy

As the pacemaker can only sense and actuate from two locations within the heart, only structures and parameters that affect inputs to the pacemaker are needed. Since the two leads are fixed, the accurate spatial locations of different heart anatomical structures are not necessary. Instead, the topology of the electrical conduction system of the heart is more important.

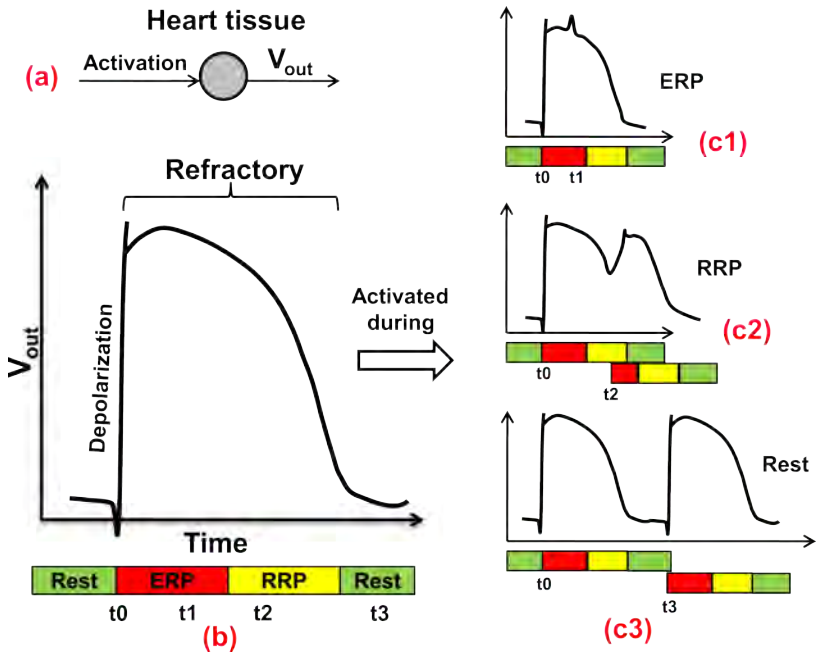


Figure 3.2: (a) The generation of Action potential; (b) Action potential; (c1) The second activation arrived during ERP; (c2) Arrived during RRP; (c3) Arrived after refractory.

3.2.2 Timing Behaviors of Cellular Electrophysiology

The contraction of heart muscles is triggered by external voltage applied to the tissue. After the activation, a transmembrane voltage change over time can be sensed due to ion channel activities, which is referred to as an Action Potential (Fig. 3.2(a)). The upstroke of the action potential is called depolarization, during which the muscle will contract. The voltage change caused by the depolarization will depolarize the tissue nearby, which causes an activation wave across the heart. After the depolarization there is a refractory period during which the tissue recovers to the pre-excitation state and the voltage drops down to the resting potential. The refractory period can be divided into *Effective Refractory Period (ERP)* and *Relative Refractory Period (RRP)* (Fig. 3.2(b)). During ERP, the tissue cannot be depolarized due to the lack of charge. As a result, the activation wave will be "blocked" at the tissue during ERP (Fig. 3.2(c1)). During RRP, the tissue is partially recovered and the tissue can be depolarized. However, the new action potential generated by the depolarization will have different morphology (e.g. attenuated in magnitude and duration), thus affecting the refractory periods of the tissue and conduction delay of the activation wave (Fig. 3.2(c2)). Fig. 3.2(c1)-(c3) show the action potential shape change and corresponding timing change in refractory periods when the tissue is activated at time stamp $t1$, $t2$, $t3$ after the initial activation $t0$.

3.2.3 Heart Model Components

We introduce the model components that can be used to configure heart models corresponding to different heart conditions. As discussed earlier, the action potential of a heart tissue has 3 timing periods during which the tissue responds to external electrical stimuli differently. We use an extended timed-automata formulation (Alur and Dill [1994]) to model the timing behaviors of a heart tissue during each cycle.

Node Automata: We refer to the tissue model as *node automaton* and Fig. 3.3.(a) shows the structure of a node automaton i. 3 states correspond to the timing periods of the action potential. From Rest state, the node can either self-activate or get activated by external stimuli (Act_node) and go to ERP state. During ERP state the node does not respond to external stimuli (blocked). During RRP state, the node can still be activated and go to ERP state, however the ERP period and the conduction delay of the tissue are affected by the "earliness" of the activation arrived during the RRP period, which is tracked by a shared variable $C(i)$. The new ERP period is determined by a function over clock value $g(f(t))$ which mimics the beat-to-beat dynamics described in Josephson [2008]. The function g and f are given by:

$$f(t) = 1 - t/Trrp \tag{3.1}$$

and

$$g(x) = \begin{cases} T_{min} + (1 - (1-x)^3) \cdot (T_{max} - T_{min}), i = AV \\ T_{min} + (1 - x^3) \cdot (T_{max} - T_{min}), i \neq AV \end{cases} \tag{3.2}$$

where T_{min} and T_{max} are the minimum and maximum value for *Terp* of the tissue.

Due to the limited number of observable points within the heart, modeling the electrophysiological behavior of every tissue of the heart and its full anatomy is unnecessary and unfeasible. In our heart models, only self-activating tissue and key hubs of the electrical conduction system are modeled as node automata.

Path Automata: The electrical conduction through the tissue between nodes are abstracted using *path automata*. The path automata can be used to represent structural or topological (functional) electrical connections between nodes. Fig. 3.3.(b) shows a path automaton connecting node a and b.

The initial state of a path automaton is *Idle*, which corresponds to no conduction. The states corresponding to the two conduction directions are named after the physiological terms: Antegrade (Ante) and Retrograde (Retro). These states can be intuitively described as forward and backward conductions. If path actuation *Act_path* event is received from one of the nodes connected to it, there is a transition to *Ante* or *Retro* state based on the activation source in the path automaton. At the same time, the clock invariant of the state is modified according to the shared variable *C(a/b)*. This corresponds to the change of the conduction delay that is caused by the early activation. Similar to node automaton, the changing trend is extracted from clinical data and the function h is defined as:

$$h(c) = \begin{cases} path\_len/v \cdot (1 + 3c), i = AV \\ path\_len/v \cdot (1 + 3c^2), i \neq AV \end{cases} \tag{3.3}$$

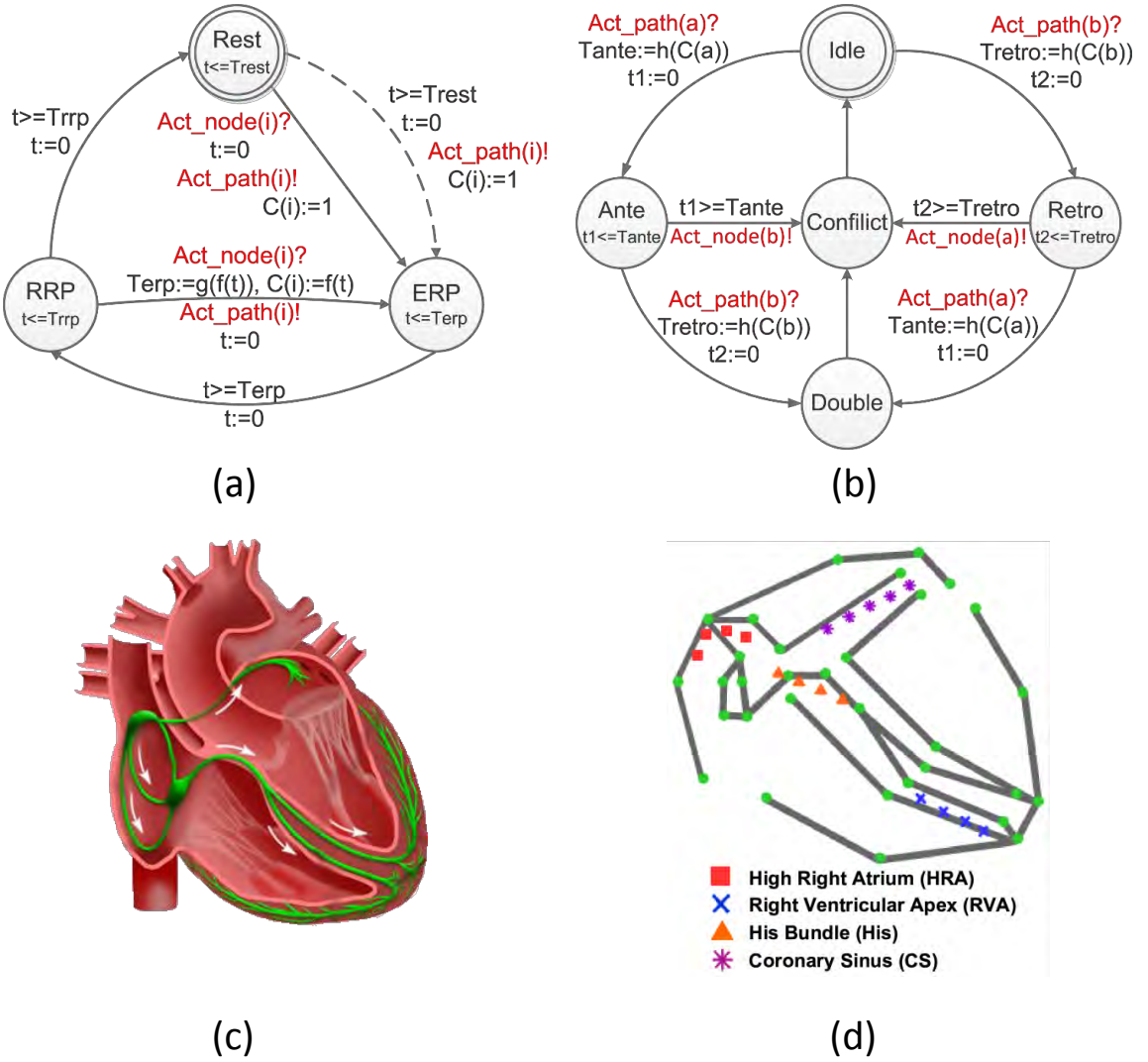


Figure 3.3: (a) Node automaton: The dotted transition is only valid for tissue (like SA node) that can be activated by an external trigger; (b) Path automaton modeling the electric conduction and propagation between two node automata; (c) Electrical conduction system of the heart; (d) Model of the electrical conduction system of the heart using a network of node & path automata Jiang et al. [2012a].

where *path_len* denotes the length of the path and v is the conduction velocity.

After *Tante* or *Tretro* time expires, the path automaton sends out *Act_node(b)* or *Act_node(a)* respectively. A transition to *Conflict* state occurs followed by the transition to *Idle* state. The intermediate state *Conflict* is designed to prevent back-flow, where the path is activated by the node b it has just activated. If during *Ante* or *Retro* state another *Act_path* event is received from the other node connected to the path automaton, a transition to *Double* state will occur, corresponding to the two-way conduction. In this case, the activation signals eventually cancel each other

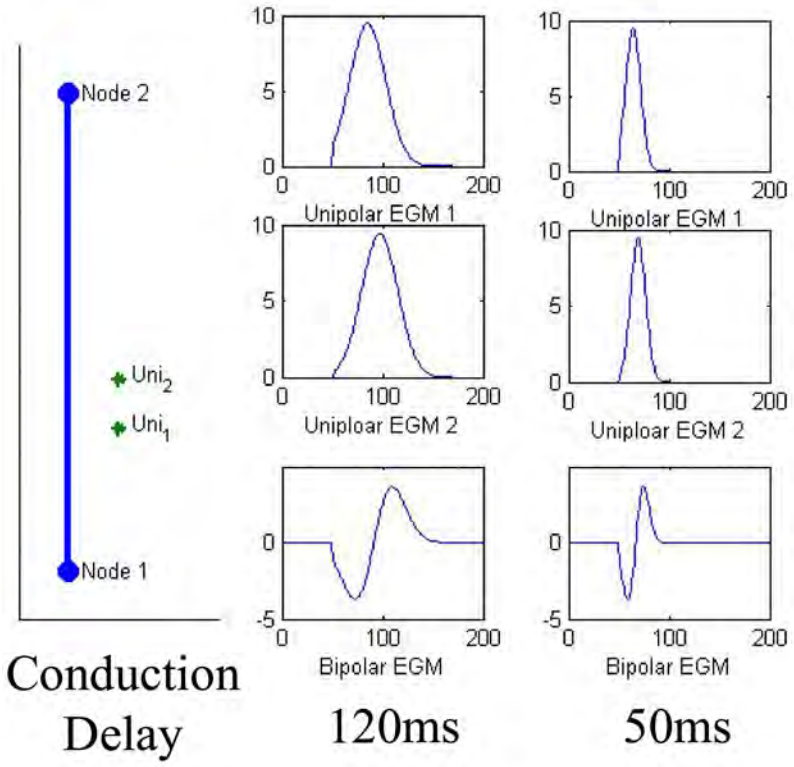


Figure 3.4: The influence of conduction velocity and probe configuration on the EGM morphology. The left columns show the placement of probes in relation to the path; the right columns show the functional EGM.

and the transition to *Idle* state is taken.

3.2.4 Modeling the Heart's Electrical Conduction System

The node and path automata are the basic building blocks for EP heart modeling. Hearts with different conditions are modeled by using different conduction topologies with appropriate timing parameters for each node and path automata. Fig. 3.3.(d) shows one such topology of a network of node and path automata.

3.3 Interaction with the Heart Model

In this section, we first introduce a probe model we developed to generate synthetic EGM signals from the EP heart model. We then use two case study to demonstrate that the probe model enables the EP heart model to evaluate device malfunctions due to sensing errors.

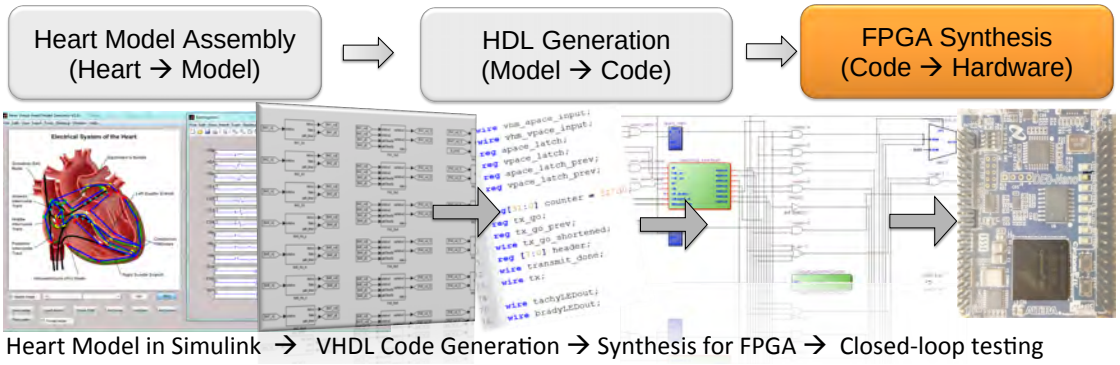


Figure 3.5: The heart model was developed in Matlab/Simulink and code was automatically generated to operate on an FPGA platform for platform-level testing.

3.3.1 Probe Model for Synthetic EGM Generation

In EP testing and during pacemaker implantation, the local electrical activities, measured as electrogram (EGM) signals, are used to diagnose heart conditions. During heart model construction, we can assign a node automaton at electrode locations and the transitions to the ERP state can be used to represent the local activation events. In a more general setup where electrodes are assigned anywhere within the heart model, a probe model is designed to generate synthetic EGM signals using spatio-temporal information from the proximity to the network of node and path automata. According to Stevenson and Soejima [2005], a potential difference is generated when the activation wavefront passes by the electrode. The locations of the activation wavefronts are calculated from the locations of the path automata and their current timer values. The amplitude of EGM decreases when the activation wavefront moves away from the probe. We assume the decrease factor is a function related to the distance between the activation wavefront and the probe. The potential difference caused by an activation wavefront to a probe is the signal strength of the path multiplied by the decrease factor. The amplitude of EGM from a probe is the sum of potential differences caused by all activation wavefronts. The bipolar EGM is the subtraction between two unipolar EGMs. Fig. 3.4 shows that this probe model captures timing properties of EGM and the functional shape of the EGM impulses. The probes can be placed anywhere within the heart model and generate clinically-relevant EGMs.

With the sensing model, the heart model structure can be used to identify safety hazards caused by sensing errors.

3.3.2 Pacemaker Oversensing and Crosstalk

Oversensing is a general term for inappropriate sensing caused by noise or far-field signals. It's very common among pacemaker malfunctions and it may result in failure to pace (Beaumont et al. [1995], Fuertes and Toquero [2003]), competitive pacing and inappropriate therapy. Crosstalk is a special case for oversensing which occurs when

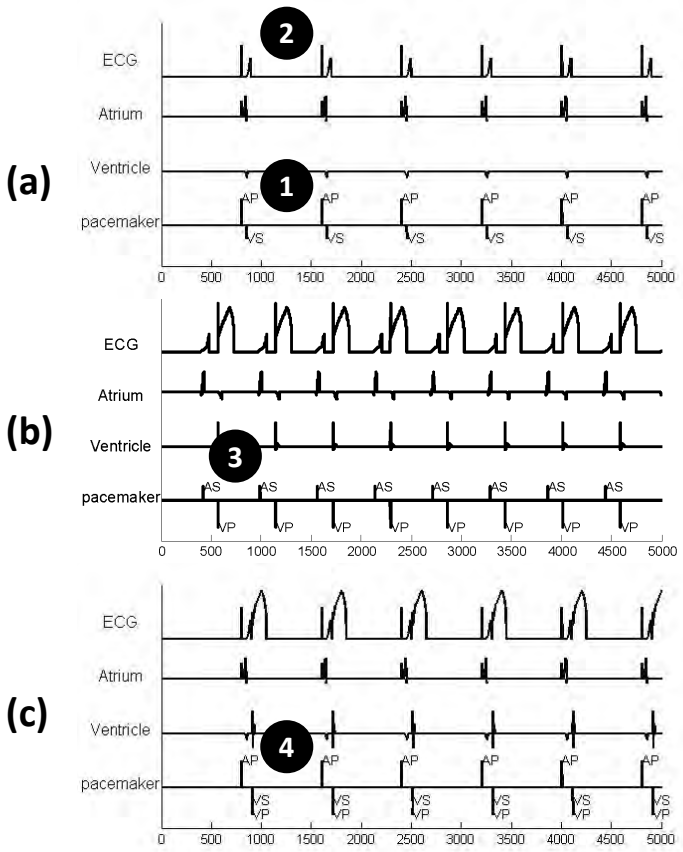


Figure 3.6: Crosstalk between pacemaker leads with high sensitivity in the ventricle, adjusted sensitivity and ventricular safety pacing

the pacemaker stimulus in one chamber is sensed in the other chamber. It happens when two leads are close to each other or pacing signal in the other chamber is too strong. It is common that the ventricular lead is placed in the right ventricle outflow tract, which is close to the atrium (Saxonhouse et al. [2005]). Fig. 3.6(a) shows simulated EGMs from a patient with bradycardia and complete heart block. During atrial pacing (AP), the pacing signal is sensed by the ventricular lead 53 ms after the AP. (Marker 1) It is treated as ventricular sense (VS) signal and thus inhibits the subsequent ventricular pacing (VP). This is indicated by no QRS-wave in the ECG channel. (Marker 2) For a patient with complete heart block this will cause dangerous ventricular asystole, meaning a long time without ventricular events.

Increasing the sensing threshold of the ventricular channel can prevent false sensing. In Fig. 3.6(b), the small signals in ventricular EGM are ignored and ventricular pacing are successfully delivered.

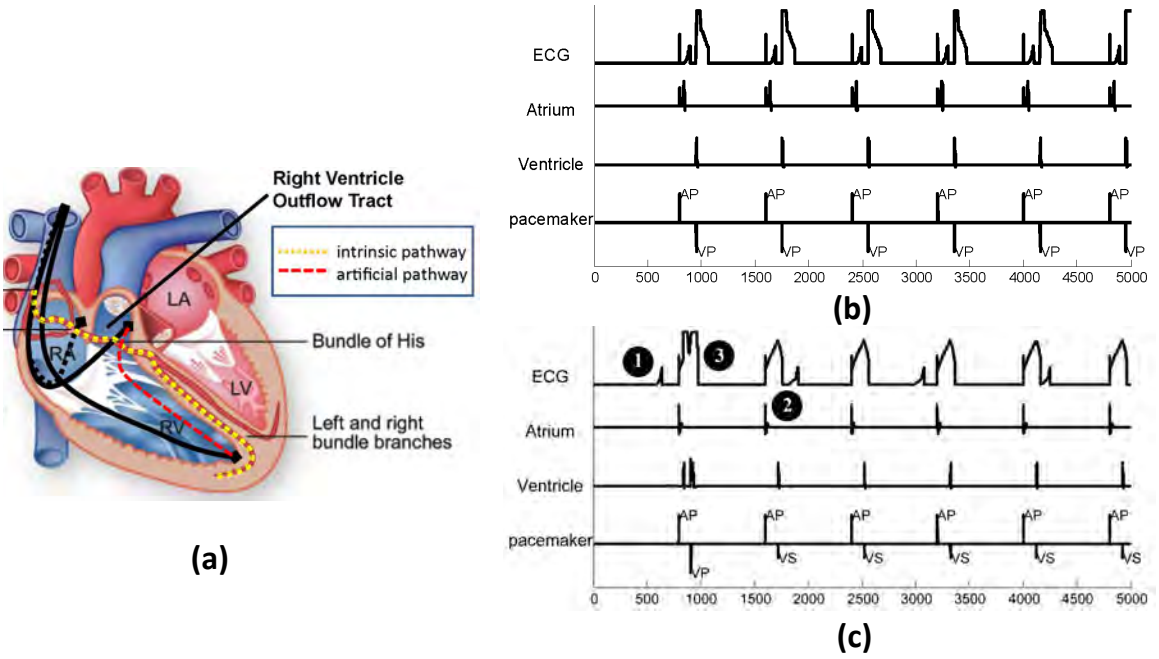


Figure 3.7: (a) Dotted line shows the location where the atrial lead should be (b) Pacemaker function before lead dislodge. (b) Pacemaker function after lead dislodge

3.3.3 Lead Displacement

Lead displacement affects many patients and can result in inappropriate or ineffective therapy. Fig. 3.7. (b) shows the simulation result for the pacemaker function when the leads are in their designated location. From the figure we can observe: 1) Each P-wave is initialized by an Atrial Pace signal. 2) Each QRS complex is initiated by a ventricular pacing signal. 3) The interval between AP and VP is 150 ms, which matches the programed AVI period.

One common case for lead dislodge is shown in Fig. 3.7.(a), where the atrial lead has fallen into the right ventricle outflow tract. In this case the atrial lead senses from the ventricle rather than atrium and atrial pacing will initiate a ventricular event. Fig. 3.7.(c) shows the simulated EGMs in this case. The figure reveals several facts: 1) No P wave is sensed or tracked (Marker 1). 2) Atrial Pace initiates an abnormal, wide QRS which is then sensed by the ventricle lead (Marker 2). 3) Intermittent appearance of VP on QRS 110 ms after the AP. The ventricular lead can receive signal from: 1) pacing signal sent from the atrial lead, 2) the intrinsic A-V conduction path. The two paths are shown in Fig. 3.7.(a) and form a timing race condition. When the signal from the atrial lead arrives the ventricular lead first, it will trigger VS. If the intrinsic signal arrives the ventricular lead during the VSP sensing window (defined in previous section), it will trigger VSP. Although the pacing is 'safe' because the pacing is early enough to avoid the vulnerable refractory period, the damage caused by pacing on depolarized tissue is currently a matter of much investigation.

3.4 Heart-on-a-Chip Platform

Platform testing remains the primary means to verify and validate device software. Currently testing is performed by feeding recorded open-loop heart signals to the device and evaluating the device output. Consequently, the change in the state of the heart condition, in response to device output, is not taken into account. Thus, device malfunctions involving state changes due to multiple closed-loop interactions will not be captured during open-loop testing.

To this effect, the heart model described above is also implemented on hardware platform (Fig. 3.5) for closed-loop testing. Since each heart model is a network of node and path automata running concurrently, we implemented the heart model on an FPGA, so that increasing in the number of nodes and paths would not affect real-time constraints. The second generation heart model implementation has been implemented on a lower cost fast micro-controller platform. The fast clock ensures that executions of all nodes and paths can be finished within 1ms. The Heart-on-a-Chip platform includes a heart model implementation which is able to represent common heart conditions such as bradycardia, tachycardia, heart block, etc (for mode details refer to Jiang et al. [2012a]). The parameters of the heart model can be changed at run-time by either switching among pre-defined parameter sets, or sending values directly to the model through a user interface in Matlab. A monitoring system observes logical interactions between heart model and the pacemaker and checks them against safety invariants at run-time.

As shown in (Fig. 3.8), with an analog interface the heart model can interact with a commercial pacemaker in real time. Our analog interface uses an optical isolation circuit to separate the pacemaker circuit and the heart implementation. Signals generated from the heart are attenuated to the appropriate level to interact with a Boston Scientific pacemaker and analog pacing signals are converted to pacing events received by the heart model.

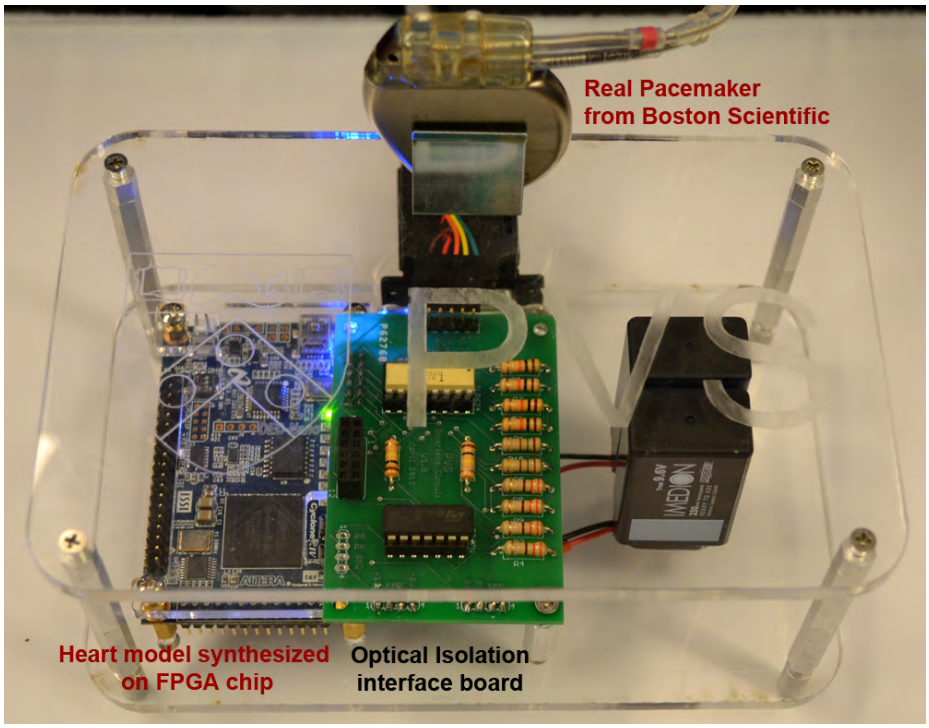


Figure 3.8: Heart-on-a-Chip testbed for real-time closed-loop testing of the pacemaker or model of the pacemaker with the heart model on the hardware platform

3.5 EP Heart Model Validation

Since models are approximations of the actual environment, there are always discrepancies between the model and the actual patient (group). The challenge is to evaluate the confidence in the safety guarantees that model-based closed-loop verification can provide. The metrics and process to validate the environment model is different for the two applications of heart modeling: in closed-loop model checking, the model's **coverage** on environmental behaviors is more important, while in closed-loop simulation, the **accuracy** of the model is more important.

In this chapter, we aim to answer the following questions and use our heart models as examples to demonstrate different validation procedures which improve the fidelity of the environment model.

- What are the different methods to validate physiological models?
- How much confidence is sufficient from the model validation process?

3.5.1 Validating Models for Closed-loop Simulation

A physiological model is considered valid for closed-loop simulation if (a) it is capable of generating the same output as the patient, for the same input; and (b) it is general enough to represent other patients with similar conditions by adjusting its parameters. The second point is to ensure that the model successfully captures the underlying mechanism instead of over-fitting the data. In the following example we validate the capability of our heart models to represent certain heart conditions according to the mechanisms described in physiological literature, and output the correct responses across a range of inputs.

Quantitative Heart Model Validation: During an EP testing procedure, the physician places catheters inside the patient's heart to observe local electrical activity from different locations of the heart. The His bundle catheter (HBE) is particularly important when evaluating the atria-to-ventricle conduction path (Fig. 3.9). For each A to V conduction there are 3 impulses which correspond to atrial contraction (A), His bundle activation (H) and ventricular activation (V). In this case study, two pacing signals $a1$ and $a2$ are delivered to the heart from the high right atrial catheter (HRA). By gradually decreasing the pacing interval in each test, certain tissue along the A-V conduction path will be activated during its refractory period, thus affecting the conduction delay further down the conduction path and change the intervals between the impulses. Fig. 3.10(a) shows the relation between pacing interval ($a1$-$a2$) and corresponding intervals between A, H and V impulses. On the left side it shows that interval $H_1 - H_2$ and $V_1 - V_2$ decrease but remain equal as the pacing interval decreases, indicating the tissue with the longest refractory period along the path is not between the His Bundle and the ventricles. When the pacing interval decreases to 350ms both intervals increases, indicating that the RRP of certain tissue has been reached and

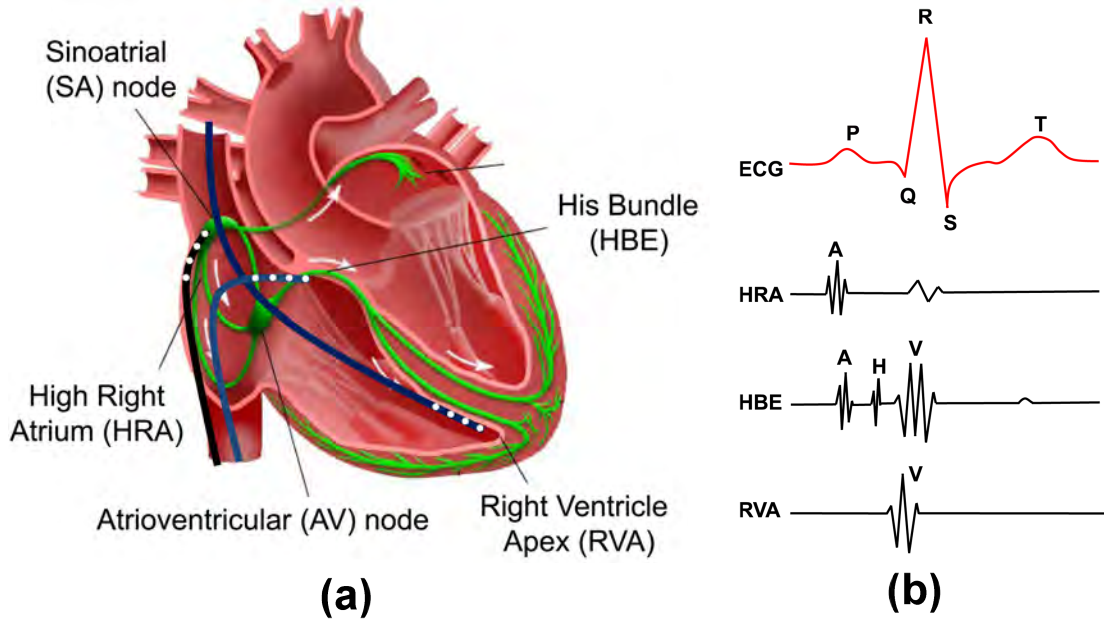


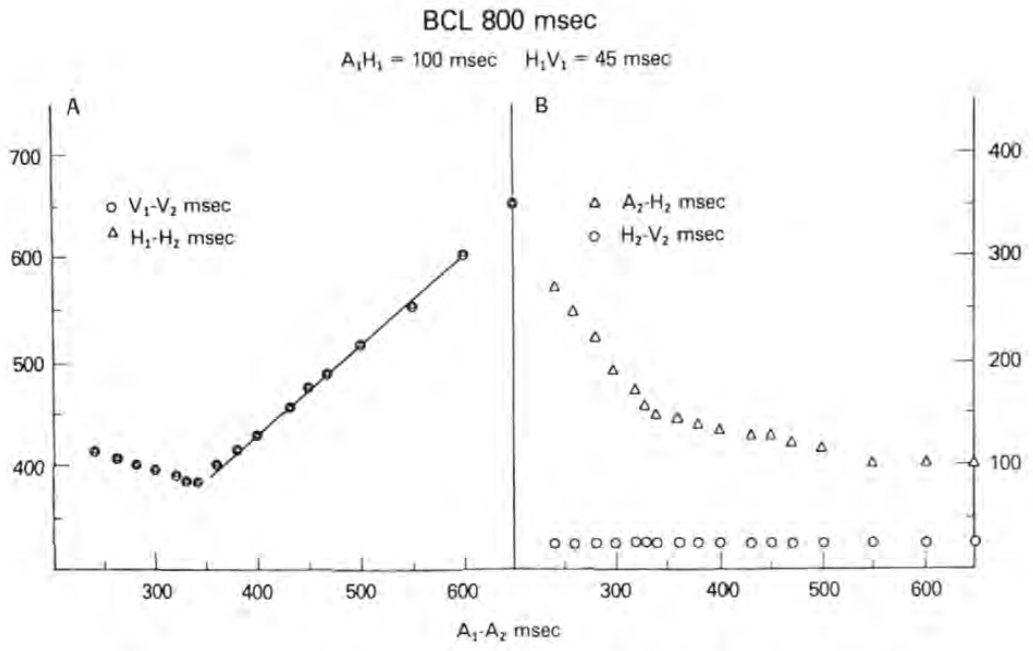
Figure 3.9: (a) Probe locations for a general EP testing procedure. (b) EGM signals measured from the probes at the high right atrial (HRA), His bundle (HBE) and right ventricular apex (RVA) standard catheter positions

the tissue is between the atria and the His bundle. On the right it shows that the $A_2 - H_2$ interval increases as the pacing interval decreases, which further proves the hypothesis that the AV node, which is between the atria and the His bundle, has the longest refractory period along the A-V conduction path. We configured our heart model such that the AV node has the longest refractory period and performed the same study by decreasing the pacing interval. The heart model shows the same trend as that of the real patient (Fig. 3.10(b)).

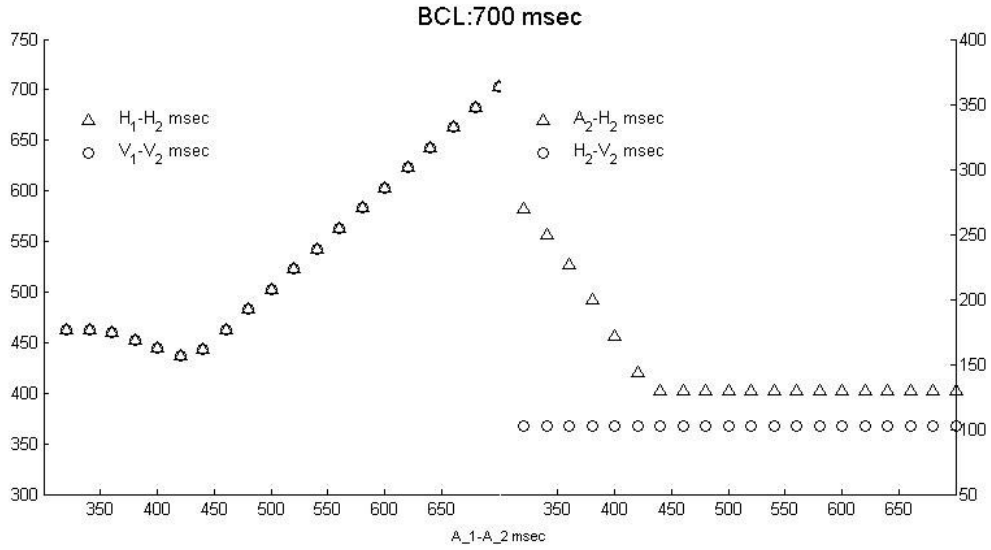
Validation by comparison to real patients: This heart condition can also show Wenckebach type A-V nodal response. In this case, a sequence of pacing signals with a short coupling interval ($A_1 - A_2 <= AV.Terp + AV.Trrp$) is delivered in the atrium. This results in a gradual increase in the AV nodal conduction delay and then a dropped beat occurs in the ventricle due to the increased ERP period of the AV node. The EGMs for a real patient with Wenckebach type A-V nodal response are shown in Fig. 3.11(a). With the VHM, we observe similar behavior, and the gradually increasing ERP and conduction delay are visualized in Fig. 3.11(b).

3.5.2 Validating Models for Closed-loop Model Checking

In model checking, a lot of complex dynamics of the environment are abstracted so that the environment model covers a larger number of environmental behaviors using non-determinism. The validity of the model is obtained by a valid initial model and a rigorous abstraction processes. In Jiang et al. [2014], we started with a valid detailed



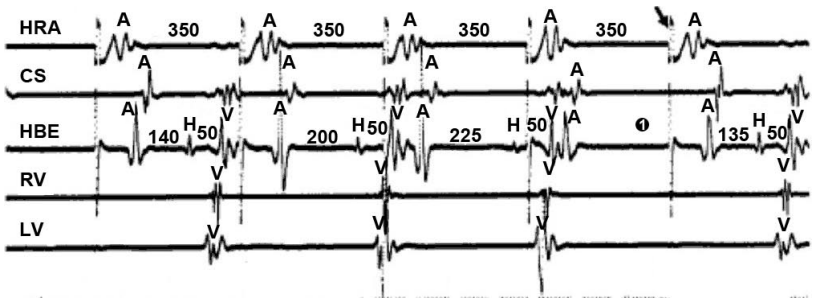
(a)



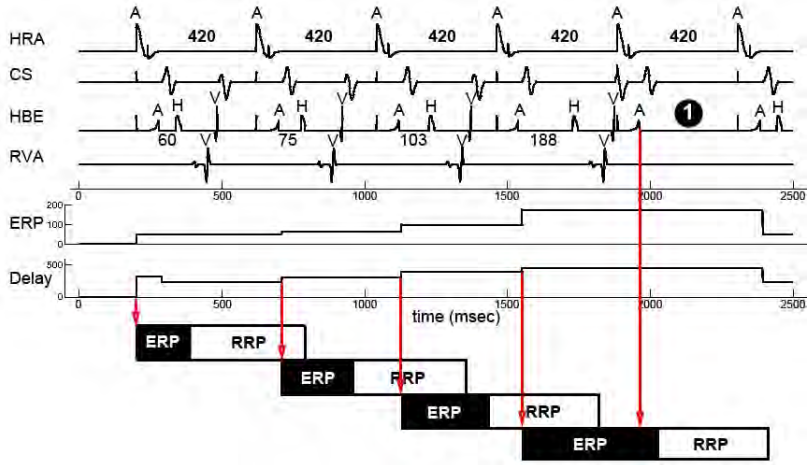
(b)

Figure 3.10: Key interval values when the coupling interval shortens for (a) a real patient (Josephson [2008]) and (b) in heart model simulation (Jiang et al. [2010b]).

deterministic model (as described above) and by applying different abstraction steps we were able to generate a series of non-deterministic heart models. Between each abstraction step, the heart models satisfy a timed simulation relationship (Yamane



(a) Real patient's electrograms



(b) Heart model's electrograms

Figure 3.11: (a) Electrograms of induced Wenckebach block in a patient. (b) Electrograms of induced Wenckebach block in the heart model with a basic cycle length of 420 msec. The heart model also displays lengthening in the A-H interval and block in A-V node (Marker 1). Rows 5 and 6 show the increase in the ERP and conduction delay of the A-V node.

[2006]) which is described below. The timed simulation guarantees all behaviors are covered in the more abstract model.

For two timed automata $T^1 = \langle S^1, S_0^1, \Sigma^1, X^1, inv^1, E^1 \rangle$ and $T^2 = \langle S^2, S_0^2, \Sigma^2, X^2, inv^2, E^2 \rangle$, a timed simulation relation is a binary relation $\mathsf{sim} \subseteq \Omega^1 \times \Omega^2$ where Ω^1 and Ω^2 are sets of states of T^1 and T^2. We say T^2 time simulates T^1 ($T^1 \preceq_t T^2$) if the following conditions holds:

- Initial states correspondence: $(\langle s_0^1, \mathbf{0} \rangle, \langle s_0^2, \mathbf{0} \rangle) \in \mathsf{sim}$
- Timed transition: For every $(\langle s_1, v_1 \rangle, \langle s_2, v_2 \rangle) \in \mathsf{sim}$, if $\langle s_1, v_1 \rangle \xrightarrow{\delta} \langle s_1, v_1 + \delta \rangle$, there exists $\langle s_2, v_2 + \delta \rangle$ such that $\langle s_2, v_2 \rangle \xrightarrow{\delta} \langle s_2, v_2 + \delta \rangle$ and $(\langle s_1, v_1 + \delta \rangle, \langle s_2, v_2 + \delta \rangle) \in \mathsf{sim}$.
- Discrete transition: For every $(\langle s_1, v_1 \rangle, \langle s_2, v_2 \rangle) \in \mathsf{sim}$, if $\langle s_1, v_1 \rangle \xrightarrow{\sigma} \langle s_1', v_1' \rangle$, there exists $\langle s_2', v_2' \rangle$ such that $\langle s_2, v_2 \rangle \xrightarrow{\sigma} \langle s_2', v_2' \rangle$ and $(\langle s_1', v_1' \rangle, \langle s_2', v_2' \rangle) \in \mathsf{sim}$.

As shown in the later chapters, these validated heart models can then be used for closed-loop verification of implantable pacemaker. Both the identification and validation of the heart models can be used to provide more confidence to the verification results, which would be helpful during the medical device certification process.

3.6 EP Heart Model Identification

Physiological models are developed to represent certain clinical conditions common across a population of patients, or the conditions of a specific patient. Consequently, the structure of the model and corresponding parameters have to be identified. This information can be obtained from electrogram data collected during medical procedures and from physiological literature in which population data has been analyzed and summarized. Due to limited interactions with the patient (e.g. during a device implantation procedure or an ablation procedure), currently the quality and quantity of patient-specific physiological data is sparse as there is generally not enough information to identify all the parameters in the heart model. A model with the spatio-temporal structure that is similar to the conduction patterns in the heart helps simplify the process of identifying the model parameters. A rigorous procedure for the model identification step is an important contributor to the model validation step. In this chapter, we first aim to answer the following questions:

- What is the importance of model identification for closed-loop simulation and model checking?
- How are models identified from patient data and patient population parameters?

In the following section, we briefly discuss our model identification effort for heart models used in two closed-loop verification applications, and their corresponding challenges. This is followed by the procedures to validate the heart models before they are used for closed-loop verification and testing of the pacemaker.

3.6.1 Heart Model Identification for Closed-loop Testing

In closed-loop simulation, a deterministic heart model should be identified to represent a specific patient under a certain heart condition. The constraints for model parameters can be obtained from patient data with *Electrophysiological (EP) Testing*. During EP testing, the physician delivers electrical pacing sequences from electrodes placed inside the patient's heart to instigate responses along fast and slow conduction pathways (Fig. 3.12). The observed patterns and timing of electrical events are used to extract conduction and propagation properties of different tissue regions across the myocardium. Since the goal for any EP testing procedure is not to determine all the timing parameters for a patient, the number of parameters that can be identified from the patient data is limited.

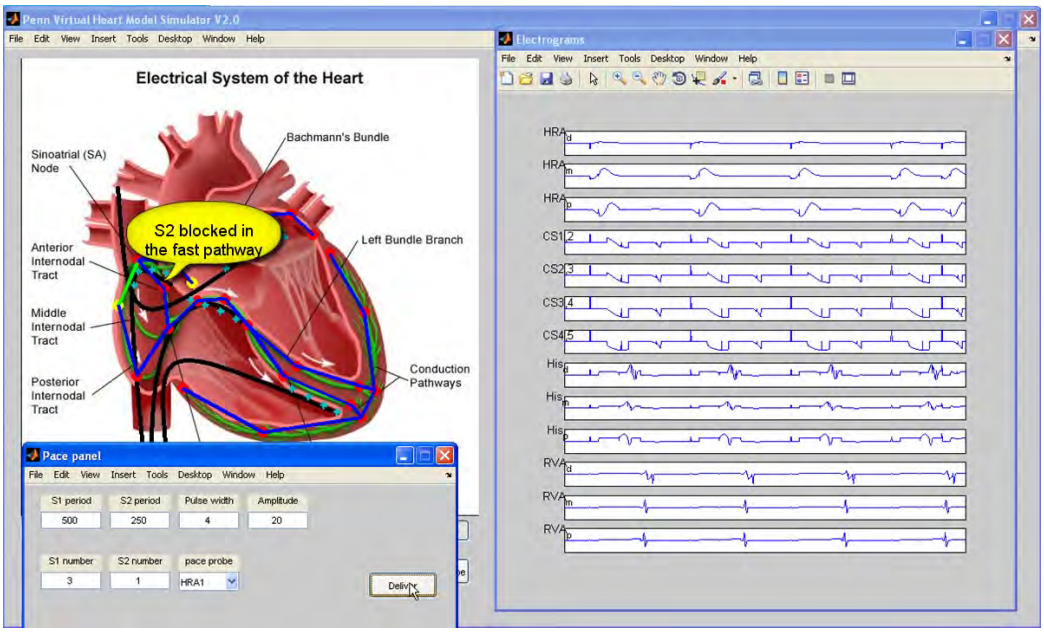


Figure 3.12: Simulation model of the heart showing the conduction pathways (left) with electrogram signals from different probe locations (right) and an interactive pacing panel (bottom left). In this case, the heart was paced four times at an interval of 500ms, followed by a pacing at a shorter (250ms) interval. This EP Testing procedure is employed to trigger conduction along alternative pathways and check for the existence of a reentry circuit.

Fig. 3.13 illustrates how timing parameters can be extracted during an EP testing procedure. Fig. 3.13(a) shows a setup with two electrodes placed in the right atrium and right ventricle of the heart respectively. EGM signals can be measured from these two electrodes (Fig. 3.13(b)). The physician delivers a series of long interval pacing sequences followed by one or more short interval pacing through the electrodes. This

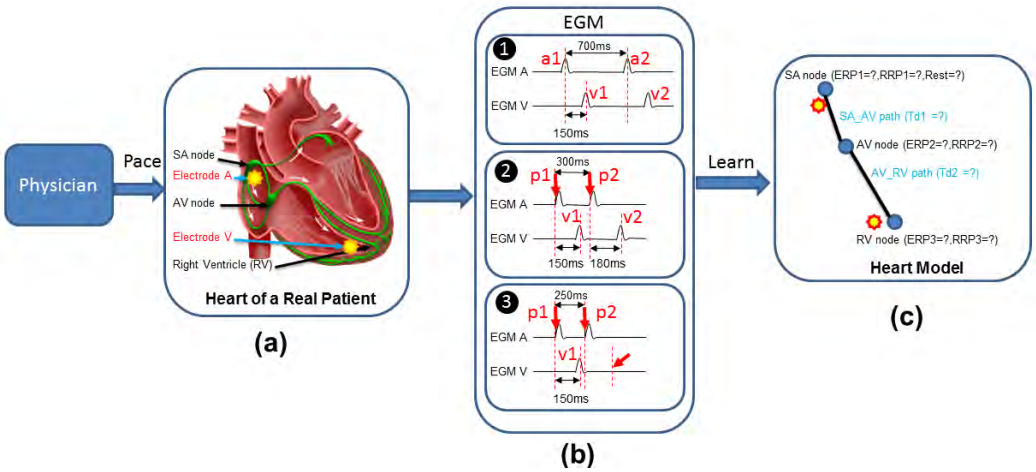


Figure 3.13: (a) The illustration of the probe locations. (b) Multiple pacing sequences with different timing outcomes. (c) The heart model with undecided parameters

may trigger different responses along primary and alternate conduction pathways from the patient's heart. Fig. 3.13(c) shows a heart model structure with unknown parameter values. By analyzing the *timing* and *pattern* of the EGM signals we extract constraints on the heart model parameters. In EGM sequence 1, the interval between two intrinsic activations $a1$ and $a2$ in EGM A is 700ms, so we have:

$$ERP1 + RRP1 + Rest = 700ms$$

The interval between $a1$ and $v1$ is 150ms, so we have:

$$Td1 + Td2 = 150ms$$

In EGM sequence 2, the pacing interval from Electrode A is 300ms. By observing that the interval between $p1 - v1$ is less than the interval between $p2 - v2$, we know that $p2$ arrives during the RRP period of the AV node. So we have:

$$ERP1 + RRP1 \leq 300ms$$

In EGM sequence 3, the pacing interval is further reduced to 250ms. There is no $v2$ corresponding to $p2$, indicating $p2$ arrives during the ERP period of the AV node. So we have:

$$ERP1 \leq 250ms$$

Each experiment provides additional time constraints for model parameters. By systematically conducting experiments certain model parameters can be uniquely identified within a relatively tight range. However, even with simplified model structure like the one in the example, not all model parameters can be uniquely identified due to limited number of electrodes and limited number of experiments during a real procedure.

Heart Model Identification in Closed-loop Model Checking

In model checking, the heart models have simpler structure and fewer parameters due to non-deterministic abstraction. The placement and connectivity of nodes and paths in the heart models are developed to be consistent with EP practice. This way, each node and path automata and their timing parameters have physiological correspondence to parameters found in literature (Fig. 3.14). The range for non-deterministic parameters directly corresponds to the range for possible values of the respective physiological parameters. Therefore, model identification for model checking is much simpler and requires less EP testing data. It is important to note here that model checking of abstract models of the closed-loop system and testing of the device in the loop are complementary approaches for validating the safety and efficacy of the overall system.

TABLE 2-1 Normal Conduction Intervals in Adults

Laboratory	P-A	A-H	H-V	H
Narula (2,5)	25–60	50–120	35–45	25
Damato (1,3,18,28)	24–45	60–140	30–55	10–15
Castellanos (6)	20–50	50–120	25–55	
Schuilenburg (23,24)	85–150	35–55		
Peuch (4,14)	30–55	45–100	35–55	
Bekheit (25,26)	10–50	50–125	35–45	15–25
Rosen (27)	9–45	54–130	31–55	
Author	60–125	35–55	10–25	

Figure 3.14: Timing intervals measured during clinical studies Josephson [2008]

3.7 Discussion

In this chapter, we use heart modeling as example to demonstrate how to develop physiological models for closed-loop evaluation of closed-loop medical devices. We emphasized that models should be developed according to their applications. We developed heart models based on clinical Electrophysiological Testing, and modeled the topological and temporal behaviors of the heart with timed-automata formulation. In the next chapter we will demonstrate the identification and validation of the heart models.

Chapter 4

Theme 2: Closed-loop Model Checking for Implantable Pacemaker

Model checking is a technique in which the state space of the model under investigation is automatically and exhaustively explored to identify executions or states that violate specified properties. Violations of the properties are returned by the model checkers as *counter-examples*, which can be used by designers to revise the design. In this chapter, model checking is used to evaluate an early design of a dual chamber pacemaker. More specifically, model checking is used to identify known and unknown physiological hazards induced by implantable pacemakers (e.g. when the pacemaker provides inappropriate therapy which drives the heart to an unsafe state).

The chapter is organized as follow: first the basis for timed-automata formalism is introduced; the dual chamber pacemaker specification introduced in Chapter 2 is implemented in UPPAAL; then the heart models used during closed-loop model checking of the pacemaker model is introduced, as well as the abstraction refinement framework for the heart models to capture heart behaviors for different applications; finally we use the heart models developed to check safety and efficacy properties of the pacemaker model, as well as the effectiveness of additional algorithms developed to mitigate two known safety hazards. We demonstrated that closed-loop model checking is capable of finding violations of safety and/or efficacy properties.

4.1 Timed Automata

Timed automata (Alur and Dill [1994]) is an extension of a finite automaton with a finite set of real-valued clocks. It has been used for modeling and verifying systems which are triggered by events and have timing constraints between events. UPPAAL is a standard tool for modeling and verification of real-time systems, based on networks of timed automata. The graphical and text-based interface makes modeling more

intuitive. Requirements can be specified using Computational Tree Logic (CTL), as described in Clarke and Emerson [1982], and violations can be visualized in the simulation environment.

Syntax of Timed Automata

A timed automaton $\mathbf{G}$ is a tuple $\langle S, S_0, \Sigma, X, inv, E \rangle$, where

- S is a finite set of locations.
- $S_0 \in S$ is the set of initial locations.
- Σ is the set of events.
- X is the set of clocks.
- inv is the set of invariants for clock constraints at each location.
- E is the set of edges. Each edge is a tuple $\langle s, \sigma, \Psi, \lambda, s' \rangle$ which consists of a source location s, an event $\sigma \in \Sigma$, clock constraints Ψ, λ as a set of clocks to be reset and the target location s'.

For the clock variables X, the clock constraints $\Psi \in \Psi^X$ can be inductively defined by $\Psi := x \perp c \| \Psi_1 \wedge \Psi_2$, where $\perp \in \{\leq, =, \geq\}$, and $c \in \mathbb{N}$.

Semantics of Timed Automata

A state of a timed automaton is a pair $\langle s, v \rangle$ which contains the location $s \in S$ and the valuation v for all clocks. The set of all states is Ω. For all $\lambda \in X$, $v[\lambda := 0]$ denotes the valuation which sets all clocks $x \in \lambda$ as zero and the rest of the clocks unchanged. For all $t \in \mathbf{R}$, $v + t$ denotes the valuation which increase all the clock value by t. There are two kinds of transitions between states. The discrete transition happens when the condition of an edge has been met. So we have:

$$\langle s, \sigma, \Psi, \lambda, s' \rangle \in E, v \models \Psi, v[\lambda := 0] \models inv(s')$$

$$\Rightarrow (s, v) \xrightarrow{\sigma} (s', v[\lambda := 0])$$

The timed transition happens when the timed automaton can stay in the same location for certain amount of time. We have:

$$\delta \in R, \forall \delta' \leq \delta, v + \delta' \models inv(s)$$

$$\Rightarrow (s, v) \xrightarrow{\delta} (s, v + \delta)$$

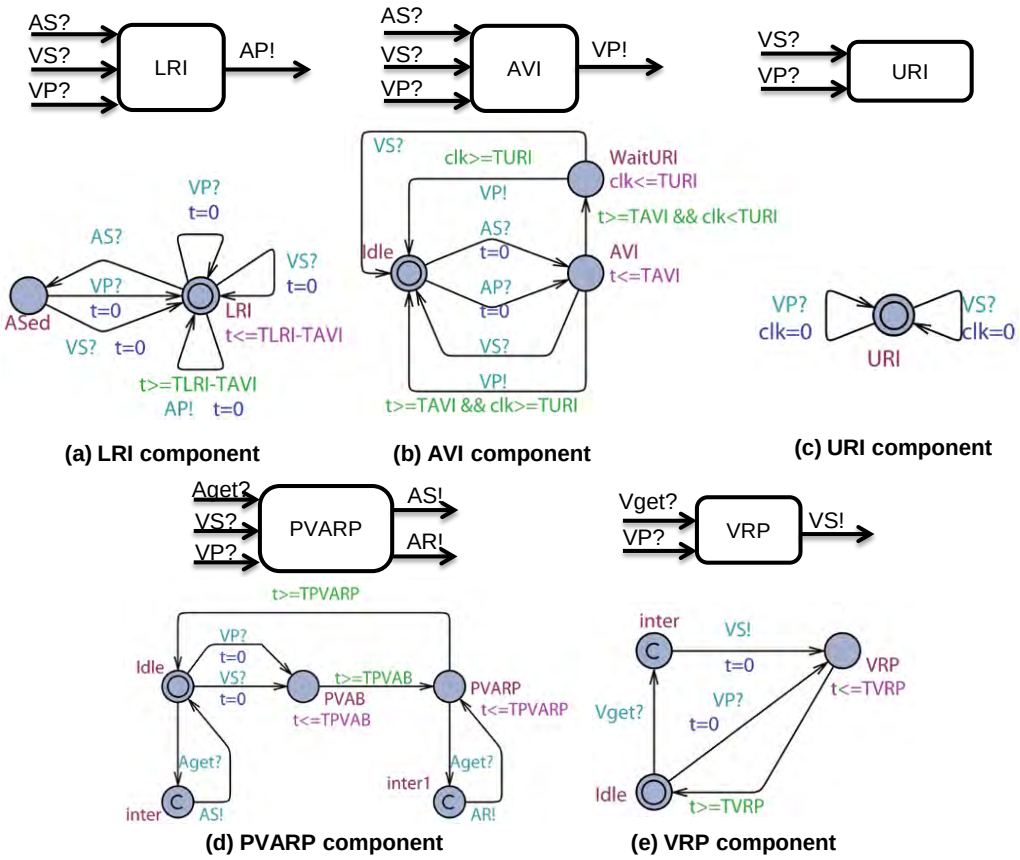


Figure 4.1: Five basic timing cycles for a dual chamber pacemaker, which include the Lower Rate Interval (LRI), Atrio-Ventricular Interval (AVI), and Upper Rate Interval (URI). Also included are the blanking intervals, Post Ventricular Atrial Refractory Period (PVARP) and Ventricular Refractory Period (VRP), to inhabit action by the pacemaker.

4.1.1 UPPAAL Model of a Dual Chamber Pacemaker

The five timing cycles introduced in Chapter 2 can be modeled as timed automata in UPPAAL. (Fig. 4.1)

4.2 Heart Models for Closed-loop Model Checking

During closed-loop model checking, the device model is verified against safety and efficacy properties under physiological conditions covered by the human physiology. The ideal physiological model should be: (1) general enough to cover possible physiological behaviors, and (2) expressive enough to distinguish specific physiological conditions from other conditions. It is obvious that no single model can achieve both properties. A rigorous framework should be adapted so that models with the appropriate level of details are selected.

4.2.1 Modeling Philosophy

Model Coverage: Pacemakers are designed to treat bradycardia by maintaining the appropriate heart rate when the intrinsic rate is low. However, at the same time, a pacemaker should not adversely affect other heart conditions such as supraventricular tachycardia (Zhang et al. [2015]). Even for the same patient, the heart condition changes over time and must be addressable by the modeling effort. In order to achieve safety across all possible heart conditions, the heart models used during closed-loop verification should be able to cover all possible heart behaviors, more precisely, their mapping to pacemaker inputs. Over-approximation (Clarke et al. [2003]) with nondeterminism can be used to simplify model structure while covering larger variety of environmental behaviors. Techniques like model-checking can then be used to examine the whole closed-loop state space for property violations.

Ambiguity due to Limited Sensing Capability: The spatial sensing resolution of pacemakers is low, in terms of the number of sensing location (2-3 leads), as well as the information obtained from each sensing location (binary events with no analog morphology). Through the process of abstraction, if different heart conditions are able to generate exactly the same input sequence to the pacemaker, there will be ambiguities in concretizing abstract closed-loop executions. For certain conditional requirements, it is important to differentiate all possible concrete executions corresponding to an abstract execution. As the result, the heart model(s) should have the capability to differentiate these heart conditions when verifying certain properties. Thus, a single heart model will not suffice and a family of heart models are required.

Information Loss during Abstraction: While over-approximation achieves simplicity and coverage, it also inevitably introduces invalid behaviors (e.g. not clinically feasible or relevant) into the model which can cause false-negatives and false-positives during model checking. To solve this problem, refined models of the heart should be available which can resolve spurious counterexamples and eliminate invalid executions when necessary to avoid false-positives.

The most challenging aspect during closed-loop model checking is the abstraction and refinement of the environment model. In Jiang et al. [2014] we developed a series of heart model abstractions at various abstraction levels. The models are abstracted using abstraction rules derived from physiological knowledge, thus ensuring that each abstraction step covers more physiological conditions. The models in adjacent abstraction levels also satisfy timed-simulation relationship (Yamane [2006]) to ensure complete coverage in the more abstract model. In the remainder of this section, we briefly discuss this multi-scale modeling process and the domain knowledge used.

Timed Simulation

For two timed automata $T^1 = \langle S^1, S_0^1, \Sigma^1, X^1, inv^1, E^1 \rangle$ and $T^2 = \langle S^2, S_0^2, \Sigma^2, X^2, inv^2, E^2 \rangle$, a timed simulation relation is a binary relation $\mathsf{sim} \subseteq \Omega^1 \times \Omega^2$ where Ω^1 and Ω^2 are sets of states of T^1 and T^2. We say T^2 time simulates T^1 ($T^1 \preceq_t T^2$) if the following conditions holds:

- Initial states correspondence: $(\langle s_0^1, \mathbf{0} \rangle, \langle s_0^2, \mathbf{0} \rangle) \in \mathsf{sim}$
- Timed transition: For every $(\langle s_1, v_1 \rangle, \langle s_2, v_2 \rangle) \in \mathsf{sim}$, if $\langle s_1, v_1 \rangle \xrightarrow{\delta} \langle s_1, v_1 + \delta \rangle$, there exists $\langle s_2, v_2 + \delta \rangle$ such that $\langle s_2, v_2 \rangle \xrightarrow{\delta} \langle s_2, v_2 + \delta \rangle$ and $(\langle s_1, v_1 + \delta \rangle, \langle s_2, v_2 + \delta \rangle) \in \mathsf{sim}$.
- Discrete transition: For every $(\langle s_1, v_1 \rangle, \langle s_2, v_2 \rangle) \in \mathsf{sim}$, if $\langle s_1, v_1 \rangle \xrightarrow{\sigma} \langle s_1', v_1' \rangle$, there exists $\langle s_2', v_2' \rangle$ such that $\langle s_2, v_2 \rangle \xrightarrow{\sigma} \langle s_2', v_2' \rangle$ and $(\langle s_1', v_1' \rangle, \langle s_2', v_2' \rangle) \in \mathsf{sim}$.

Certain properties are preserved for timed simulation relation. For $\varphi \in ATCTL$, if $M \preceq_t M'$, we have $M' \models \varphi \Rightarrow M \models \varphi$.Yamane [2006] However, $M' \not\models \varphi \Rightarrow M \not\models \varphi$ does not hold. Violations of $ATCTL$ yield counter-examples and the validity of which need to be checked on more refined model.

It is known that timed simulation relation is also closed under composition Yamane [2006]. So when we have two heart models $H_1 \preceq_t H_2$ we will have $H_1 \| P \preceq_t H_2 \| P$ where P is the timed-automata model of the pacemaker. For $\varphi \in ATCTL$, we have $H_2 \| P \models \varphi \Rightarrow H_1 \| P \models \varphi$. With this property we can verify the pacemaker model with abstract heart model. In the rest of the section, we will describe how we develop our initial heart model from the physiological perspective and abstract the model step by step so that the complexity of the model is reduced for verification. Given two heart models H_1, H_2 and a timed simulation mapping $\mathsf{sim}{=}\Omega_1 \times \Omega_2$, there are no automated methods to check $H_1 \preceq_t H_2$. In the Appendix, we show the manual proof for the timed simulation relation between two heart models H_2 and H_3. Other timed simulation relations can be proved similarly.

4.2.2 Counter-Example-Guided Abstraction Refinement

In closed-loop model checking, there is only one device model. However there can be a large number of environmental conditions which require different models to represent them. For instance, a heart with atrial flutter has an additional conduction pathway that is not present in a healthy heart, causing fast atrial rate. The timing and structural differences of different heart conditions should be distinguished in corresponding heart models. A set of initial models of the environment can be constructed but the set is inherently incomplete because of the large number of environment conditions and their combinations. As a result, performing model checking using every model in the set cannot ensure full coverage of the environmental conditions.

In this paper, domain-specific over-approximation rules are developed that produce abstract models that not only cover explicitly modeled environment conditions,

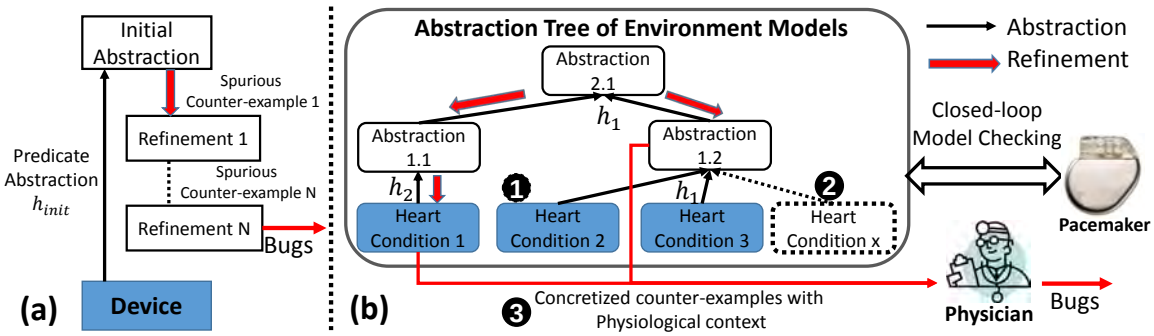


Figure 4.2: (a) Device modeling with CEGAR framework (b) Closed-loop model checking with environment abstraction tree.

but also cover timing behaviors and conditions not modeled in the set of initial models. The abstract models can be then used for closed-loop model checking of the device model. If the closed-loop system satisfies a requirement, the device under verification satisfies the requirement under environment conditions covered by the abstract models. However, if the requirement is not satisfied, the model checker returns a counter-example. In device modeling, the counter-example is considered *spurious* if it can not be produced by the device (as shown in (Fig. 4.2(a)). However in environment modeling, even if the counter-example can not be produced by any of the initial environment models, it might still be a physiologically valid behavior. Thus the validity of a counter-example cannot be determined by refining the environment model, but can ultimately only be determined by domain experts.

Counter-examples returned from abstract models can be difficult to interpret by domain experts. One abstract counter-example could be produced by multiple physiologically valid conditions, which causes ambiguity. Thus, a rigorous framework is necessary to balance the need to cover a wide range of environmental conditions and the need to provide counter-examples to the physicians within their physiological context.

Another challenge for closed-loop model checking of medical devices is the amount of domain expertise needed during: 1) physiological modeling, 2) model abstraction and refinement, and 3) checking the validity of counter-examples. Thus the framework must also allow non-domain experts to perform verification (item 2 above), and establish 'hand-off' points where the results of verification can be handed back to the experts for interpretation.

4.2.3 Abstraction Tree for Heart Model Abstraction Refinement

The ideal heart model for closed-loop model checking of an implantable pacemaker not only covers all possible inputs to the pacemaker, but also has physiological explanations to all known heart conditions. However, no **single** heart model can satisfy

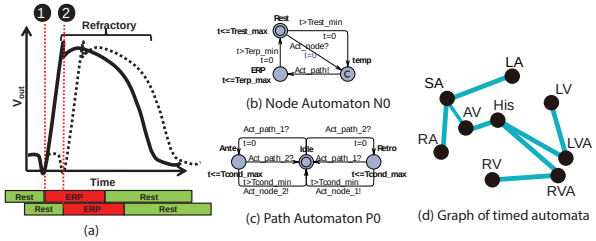


Figure 4.3: Node and Path Automata which models the timing properties of the heart tissue. A network of node and path automata models the generation and conduction of electrical activities of a heart

both requirements. Therefore, a set of heart models must be employed where the different abstraction levels of the models strike a balance between coverage and expressiveness. More importantly, the heart models should have rigorous relationships among each other to provide formal guarantees.

In this section we present the abstraction tree framework that maintains formal *Timed Simulation* relationships between heart models and enables automated closed-loop model checking of implantable pacemaker.

Initial Set of Heart Models

The heart model structure discussed in Chapter 3 is implemented in UPPAAL as shown in Fig. 4.3. Dynamic changes of the ERP periods and conduction delays are abstracted as ranges using *non-determinism* in timed-automata. This enables the heart model structure to capture behaviors of the heart models with timing variability. This heart model structure is based on clinical electrophysiology, with state variables and parameters directly corresponding to physiological parameters. Therefore, domain experts from clinical electrophysiology can construct models of different heart conditions with their domain expertise and literature.

An example set of initial heart models is shown in Fig. 4.4. The different topologies of node and path automata represent the mechanism of different heart diseases. These heart models represent the current knowledge for heart condition variability, thus the set is inherintly incomplete, meaning there is no guarantee for 100% safety even if a property is satisfied in all of these models. These models are mostly used for providing physiological contexts for counter-examples returned by the model checker. Domain experts can always expand the set with knowledge of new heart conditions.

Interaction With the Pacemaker

The interactions between the heart and the pacemaker are modeled by using binary event channels. For the atrial lead, we have: $N_A.Act\_path! \rightarrow \mathsf{AS}!$, and for ventricular lead we have $N_V.Act\_path! \rightarrow \mathsf{VS}!$.
The pacemaker accordingly generates atrial or ventricular pacing actions $\mathsf{AP}! \rightarrow N_A.Act\_node!$ and $\mathsf{VP}! \rightarrow N_V.Act\_node!$.

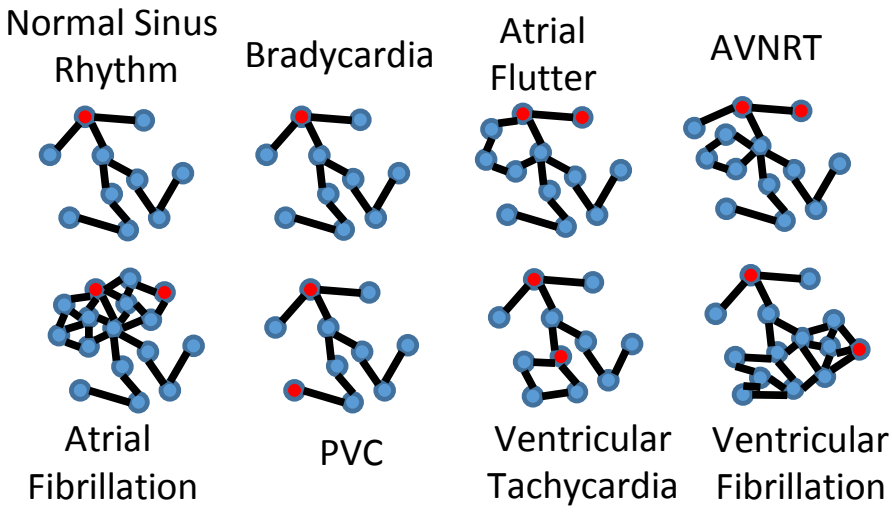


Figure 4.4: Examples of the initial set of heart models. The models are different in node and path topology and/or timing parameters.

Physiological Abstraction Rules

The initial set of heart models only represents a subset of all possible conditions. There always exists conditions that beyond our knowledge or that are combinations of known conditions. By using *over-approximation*, heart models can be created that cover the observable behaviors of the initial set and that introduce behaviors that were not captured in the initial set. Inevitably, some of the introduced behaviors will be physiologically invalid. This problem can be alleviated by carefully designing the abstraction rules so that behaviors introduced are mostly physiologically valid. The physiologically invalid behaviors can be eliminated during a validity check in the abstraction tree.

Physiological abstraction rules are developed to cover observable behaviors of heart models. Applying one abstraction rule to heart model(s) $H_1, H_i \ldots H_n, n \geq 1$ yields an abstract heart model H' such that all observable behaviors of H_i are covered by H'. For each heart model H_i, H' is a *timed simulation* of H_i. To illustrate, a subset of abstraction rules is described intuitively. The complete set of abstraction rules and the proofs of timed simulation relationship can be found in the tech report Jiang et al. [2015].

Rule R1: Convert Reentry Circuits to Activation Nodes

Within the conduction network of the heart, there can be multiple pathways between two locations, forming conduction loops. If the timing parameters of the tissue along the loop satisfy certain properties, there can be scenarios in which a depolarization wave circling along the circuit. The circuits are referred to as *Reentry Circuits*. Since the time interval for an activation wave to circle a reentry circuit is usually less

than the intrinsic heart cycle length, the heart rate will be "'hijacked"' by the reentry circuit once the cycling is triggered, causing tachycardia. Reentry is the most common mechanism for tachycardia, which can be captured by our heart models that are used in Jiang et al. [2010a].

The effect of reentry tachycardia is that activation signals coming out of the circuit with a given cycle length equal the sum of conduction delays along the circuit. It is therefore reasonable to model a reentry circuit as a self-activation node with the self-activation range equal to the sum of conduction delays.

Applicable Condition: The rule only affects the topology of the model, and therefore can be applied without preliminaries.

Output model: The "essential structure" of a heart model is the shortest paths (in terms of conduction delay) connecting self-activation nodes and/or sensing nodes. First detect all circles in the input graph. For each circle with nodes $N_i, i \in [1 \dots n]$ and paths $P_j, j \in [1 \dots m]$, remove all "non-essential" nodes and paths, create a node automaton N_s and connect to the nearest sensing node with a path automaton P_s.

Effect on parameters: For the new node automaton N_s, the minimum of the Trest parameter is set to the minimum of the sum of the conduction delays within the reentry circuit, and the maximum is set to infinity

Effect on behaviors: The new model captures the behavior of the original model when the reentry circuit is active and inactive. Additionally, the new model captures the behaviors of other heart conditions in which the rate of the reentry circuit is lower.

Fig. 4.5 shows an example in which a circle is replaced by a self-activation node.

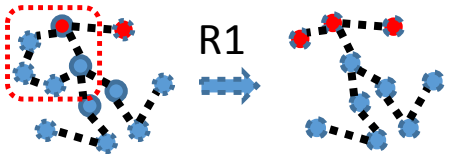


Figure 4.5: Rule 1: Remove reentry circuits from the model

Rule R2: Remove Non-essential Structures

After the circles within the topology are removed, the topology of the heart model is in the form of a tree. Since the "non-essential" structures do not affect the activation signals from and/or to the sensing nodes, all the "non-essential" structures can be removed.

Applicable Conditions: The rule can only be applied after Rule 1 has been applied.

Output model: Trimmed topology with only the essential structure remaining.

Effects on parameters: There are no effects on parameters of the node and path automata.

Effects on behaviors: Applying this rule does not affect the observable behaviors of the model.

Fig. 4.6 shows an example in which non-essential structures are removed.

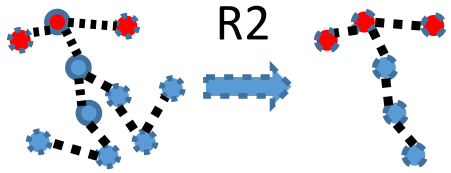


Figure 4.6: Rule 2: Remove non-essential structures

Rule 3: Removing Unnecessary Non-self-activation Nodes

The effect of non-self-activation nodes is blocking electrical events with interval shorter than its ERP period. If the self-activation nodes at both ends of a core path have self-activation interval longer than the maximum ERP period of nodes along the core path, the nodes can be removed.

For a core path from a self-activation node N_1 to another core node N_2, for any structure $P_1 - N_n - P_2$ which N_n is a non-self-activation node, if $N_n.ERP_{max} < min(N_1.Rest_{min}, N_2.Rest_{min})$, replace $P_1 - N_n - P_2$ with P_3 so that:

$$P_3.cond_{min} = P_1.cond_{min} + P_2.cond_{min}$$

$$P_3.cond_{max} = P_1.cond_{max} + P_2.cond_{max}$$

Rule R4: Merge Parameter Ranges

Timing periods of heart tissue, such as Rest and ERP, are modeled as locations in the node and path automata. The minimum and maximum time an automaton can remain in a location is governed by the parameters in the guards and invariants. By merging and expanding these periods, new behaviors are introduced where a heart model may remain longer in Rest, activate or self-activate a node faster, andsoforth.
Applicable Conditions: This rule applies to heart models with the same node and path topology but possibly with different parameters.
Output Model: The abstract model has the same topology as the original models.
Effects on parameters: The parameter ranges in the new model are a super-set of the parameter ranges in the old models.
Effects on behaviors: The abstract model captures all behaviors of the original models. In addition, heart conditions with parameters outside of the ranges of the original models are covered.

Rule 5: Merge Self-activation Nodes with Interaction Nodes

The effect of self-activation nodes on the interaction of the pacemaker is triggering sensing events within certain delay. In this rule we merge all the self-activation nodes

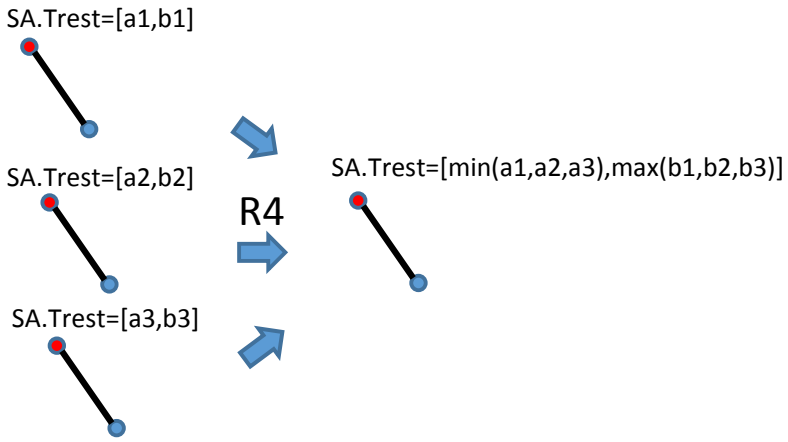


Figure 4.7: Rule 4: Merging parameter ranges

to their neariest interaction nodes. If there exists multiple self-activation nodes merging to the same interaction node, the parameters of the new model are determined following Rule 3.

Rule R6: Replace Blocking With Non-deterministic Conduction

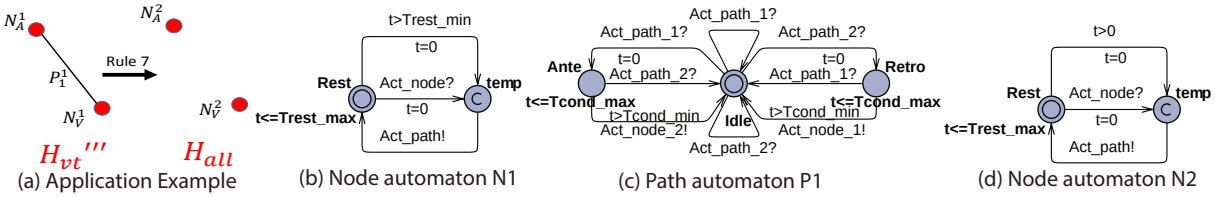
Consider the structure $N_1P_1N_2P_2N_3$ with three nodes and two paths, where N_2 is a passive node (i.e. not self-activating). If N_2 blocks an activation signal from N_1 to N_3, this is equivalent to the paths P_1 or P_2 not conducting. In this rule, the structure $P_1N_2P_2$ is replaced by a path P whose automaton can take a self loop when it receives an activation signal, thus effectively stopping the conduction. This is shown in Fig. 4.8: the extra transitions are marked Act_path_1? and Act_path_2?. Because the blocking effect of nodes is now incorporated into the paths, the node automata of self-activating nodes can be modified to the one shown in Fig. 4.8, which doesn't have the (now useless) ERP period.

Subgraph to which it applies. Line graphs with 3 vertices $N_1P_1N_2P_2N_3$, and self-activating nodes.

Applicability conditions. N_2 is a passive node.

Output subgraph. $N_1'P'N_3'$ A path P' whose path automaton is as shown in Fig. 4.8.b. The self-activating nodes N are replaced by nodes N' with automata shown in Fig. 4.8.a.

Effect on parameters For the new path, $P.cond_{min} = P_1.cond_{min} + P_2.cond_{min}$ and $P.cond_{max} = P_1.cond_{max} + P_2.cond_{max}$ For the new nodes, $N'.Trest_{min} = N.Terp_{min} + N.Trest_{min}$ and $N'.Trest_{max} = N.Terp_{max} + N.Trest_{max}$.



(a) Application Example (b) Node automaton N1 (c) Path automaton P1 (d) Node automaton N2

Figure 4.8: (a) Rule 7 application example; (b)(c) Node and path automata used in H_{vt}'''; (d) Node automata used in H_{all}

Rule R7: Replace Conduction With Self-activation

We describe Rule R7 as it illustrates both effects of an abstraction rule: structure change and modifications to the automata. The effect of a conduction path is to conduct electrical activity from a node. Since the pacemaker cannot distinguish self-activation of the node and activation triggered by path conduction, we can use self-activation to replace path conduction. If all self-activation nodes are allowed at any time by setting their minimum Rest period to 0, all the conduction paths can be removed, while preserving the original behaviors (where the Rest period was constrained to a finite interval).

Applicability conditions. This rule can only be applied after Rule 5 and Rule 6 have been applied.

Output graph. All edges are deleted: $G' = (V(G), \emptyset)$. The node automata are replaced with the one shown in Fig. 4.8.d.

Effect on parameters For every node automaton N in G', $N.Trest_{min} = 0$.

Now we use R7 as example to demonstrate the timed-simulation relationship between heart models before and after the application of R7. Consider Fig. 4.8.(a) showing an application of R7, $H_{vt}''' = N_A^1 P^1 N_V^1$ is abstracted to $H_{all} = N_A^2 N_V^2$. Here we prove that $H_{vt}''' \preceq_t H_{all}$ with observable events $\Sigma_o = \{N_A.Act\_path, N_V.act\_path\}$. The state of H_{vt}''' is represented by $(N_A^1.loc, P_1^1.loc, N_V^1.loc, N_A^1.t, P_1^1.t, N_V^1.t)$ and the state of H_{all} is represented by $(N_A^2.loc, N_V^2.loc, N_A^2.t, N_V^2.t)$. Due to space limit, only one transition from each category is presented:

Initial state: First for the initial state we have:

$$\langle (Rest, Idle, Rest, 0, 0, 0), (Rest, Rest, 0, 0) \rangle \in sim_o$$

Timed transitions: Consider a timed transition in H_{vt}'''

$$(Rest, Idle, Rest, t_1, t_2, t_3) \xrightarrow{\tau} (Rest, Idle, Rest, t_1 + \tau, t_2 + \tau, t_3 + \tau)$$

in which $(\tau \in \mathbb{R}) \wedge (t_1 + \tau \leq N_A^1.Trest\_max) \wedge (t_3 + \tau \leq N_V^1.Trest\_max)$. For a state in H_{all} such that $\langle (Rest, Idle, Rest, t_1, t_2, t_3), (Rest, Rest, t_1, t_3) \rangle \in sim_o$, there is a timed transition:

$$(Rest, Rest, t_1, t_3) \xrightarrow{\tau} (Rest, Rest, t_1 + \tau, t_3 + \tau)$$

and $\langle (Rest, Idle, Rest, t_1+\tau, t_2+\tau, t_3+\tau), (Rest, Rest, t_1+\tau, t_3+\tau)\rangle \in sim_o$.
Discrete transitions: Consider a discrete transition in H'''_{vt}

$$(Rest, Ante, Rest, t_1, t_2, t_3) \xrightarrow[t_2 \in [P_1^1.Tcond\_min, P_1^1.Tcond\_max)]{N_V^1.Act\_path!} (Rest, Idle, Rest, t_1, t_2, 0)$$

in which $N_V^1.Act\_path! \in \Sigma_o$.
For a state in H_{all} such that $\langle (Rest, Idle, Rest, t_1, t_2, t_3), (Rest, Rest, t_1, t_3)\rangle \in sim_o$, there is a discrete transition:

$$(Rest, Rest, t_1, t_3) \xrightarrow[t_3 \in [0, N_V^2.Trest\_max)]{N_V^2.Act\_path!} (Rest, Rest, t_1, 0)$$

and $\langle ((Rest, Idle, Rest, t_1, t_2, 0)), (Rest, Rest, t_1, 0)\rangle \in sim_o$. Basically activation due to conduction is replaced by self-activation of the corresponding node automata.
Additional behaviors: The timed-simulation also allows additional behaviors into H_{all}. Consider a discrete transition in H_{all}

$$(Rest, Rest, t_1, t_3) \xrightarrow[t_3 \in [0, N_V^2.Trest\_min)]{N_V^2.Act\_path!} (Rest, Rest, t_1, 0)$$

However, for a state in H'''_{vt} such that $\langle (Rest, Idle, Rest, t_1, t_2, t_3), (Rest, Rest, t_1, t_3)\rangle \in sim_o$, when $t_3 \in [0, N_V^1.Trest\_min]$ there is no available discrete transitions. Physiologically, these implicitly included behaviors correspond to fast heart rate, premature heart events and even noise.

Abstraction Tree

By applying the abstraction rules to the initial set of heart models, an abstraction tree is created. Fig. 4.9 shows an example of an abstraction tree with the root model capturing all possible input sequences to the pacemaker. Self-activating nodes are marked as red and the Trest parameters are specified next to them. Note that this abstraction tree is not unique. With a different initial set of heart models and/or different rule application orders the abstraction tree can be very different. The abstraction tree can also be extended at any time if new heart conditions are specified. The following section demonstrates the use of this abstraction tree during the closed-loop model checking of the pacemaker design.

4.3 Efficacy Validation for Implantable Pacemaker

The most essential function for the pacemaker is to treat bradycardia by maintaining the ventricular rate above a certain threshold. We define the region where the ventricular rate is slow, as unsafe. The monitor PLRI_test is designed to measure intervals between ventricular events and is shown in Fig. 4.10(a). For property

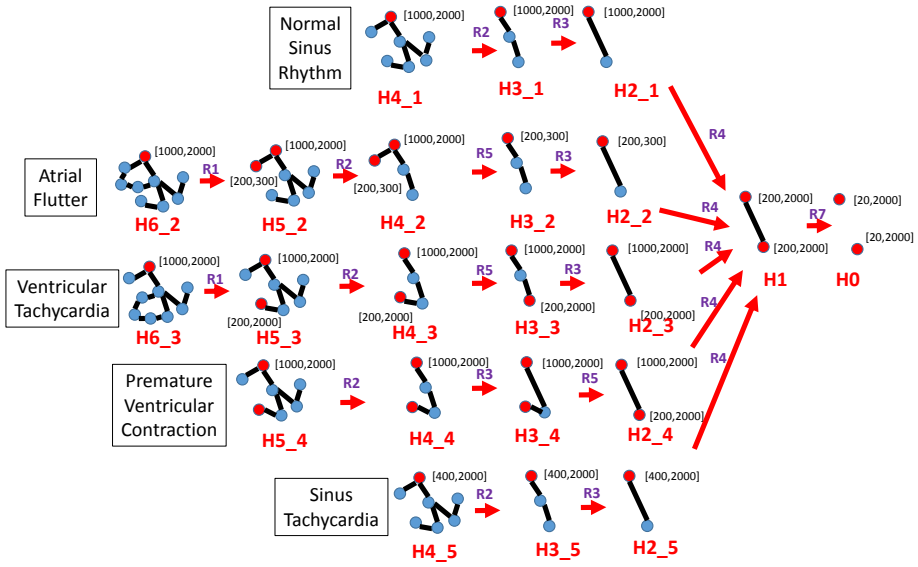
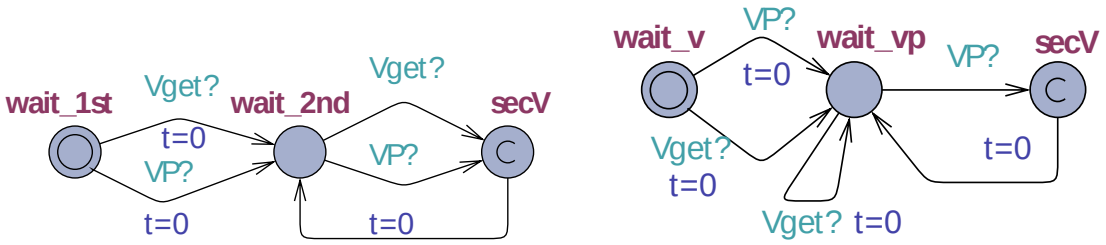


Figure 4.9: One example of abstraction tree of heart models



(a) Monitor PLRI_test

(b) Monitor PURI_test

Figure 4.10: (a) Monitor for LRL: Interval between two ventricular events should be less than TLRI, (b) Monitor for URL: Interval between a ventricular event and a VP should be longer than TURI

$$\varphi_{LRI} = \text{A[] (PLRI\_test.secV imply PLRI\_test.t} \leq \text{TLRI)}$$

we have a closed-loop system with heart model H_0 :

$$H_0 \| P \| PLRI\_test \models \varphi_{LRI}$$

The pacemaker is not designed to treat tachycardia so it can only pace the heart to increase its rate and cannot slow it down. To mitigate the hazard that the pacemaker may increase the heart rate above physiological need, an Upper Rate Interval (URI) is specified such that the pacemaker can increase the ventricular rate up to this limit.

We require that a ventricle pace (VP) can only occur at least $TURI$ after a ventricle event (VS, VP). The monitor PURI_test is shown in Fig. 4.10(b). For the property

$$\varphi_{URI} = \text{A[] (PURI\_test.secV imply PURI\_test.t} \geq \text{TURI)}$$

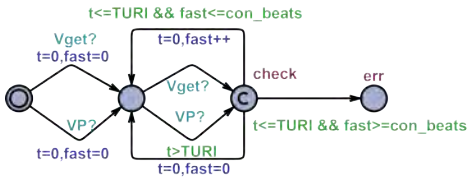


Figure 4.11: UPPAAL monitor for Property 1

we have:

$$H_0 \| P \| PURI\_test \models \varphi_{URI}$$

As we saw in the above examples, the efficacy requirements are satisfied with the most abstract heart model. Therefore no heart model refinements are necessary and the requirements are satisfied under all possible heart conditions.

4.4 Safety Validation for Implantable Pacemaker

A dual chamber pacemaker is designed to increase the heart rate during bradycardia, but it should also not increase the heart rate inappropriately. Inappropriate increase of heart rate by the pacemaker is referred to as *Pacemaker Mediated Tachycardia (PMT)*. Previous work **?** used model checking to identify two known PMT conditions. However, in order to avoid ambiguities in the counter-examples, the properties for the two PMTs were specified very specifically, which abandoned the advantage of model checking to find unknown safety violations.

With the abstraction tree approach, the ambiguities can potentially be resolved since the abstraction tree considers all known heart conditions. Therefore the property for PMTs can be specified more generally. In this paper, we specify a property such that

φ_{PMT}: The interval between ventricular events (Vget, VP) should not be shorter than TURI for 30 consecutive beats

which means that the ventricular rate should not be faster than the upper rate interval (TURI) for too long, either intrinsically or because of pacemaker interaction. Counter-example because of intrinsic fast ventricular rates can be removed from results after analysis from the abstraction tree.

A UPPAAL monitor M_{con} for the property is shown in Fig. **??**, and the TCTL property is specified as:

$$A[]\ not\ M_{con}.err$$

The model checker UPPAALLarsen et al. [1997] was used to check Property 1 on the pacemaker model using the abstraction tree of heart models. The property is violated in $H0||PM$, thus thus the abstraction tree is followed to select pair $H1$ with the pacemaker model and Property 1 is verified again. The process continues till either the leaves of the tree are reached of the property is satisfied. The result is

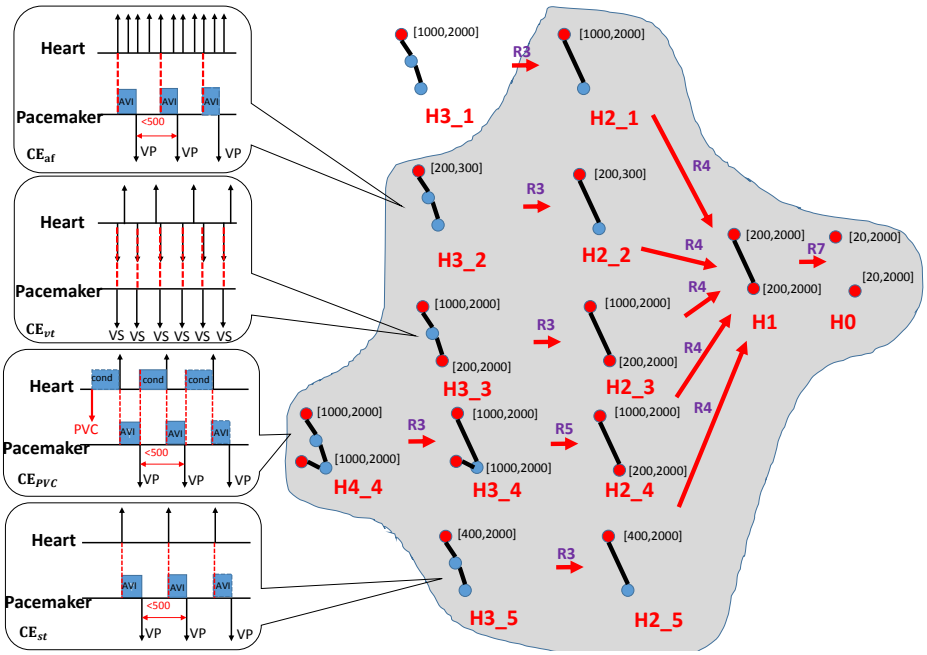


Figure 4.12: Four different physiological conditions in which Property 1 is violated. In CE_{af} the pacemaker extends a fast atrial rate to a dangerously fast ventricular rate; in CE_{vt} the ventricular rate is intrinsically fast; in CE_{st} the pacemaker appropriately maintained A-V conduction delay; in CE_{pvc} the pacemaker inappropriately increased the ventricular rate, causing Endless Loop Tachycardia

shown in Fig. 4.12, which demonstrates 5 different scenarios that can happen when using the abstraction tree. The shaded area marks the heart models with counter-examples. **Case 1:** Property 1 is violated in $H2\_1||PM$ but is satisfied in its children $H3\_1||PM$. Careful examination of the counterexample finds it to be spurious and so it is successfully eliminated by model refinement.

Case 2: CE_{af} is returned by $H3\_2||PM$ and corresponds to intrinsic atrial tachycardia with fast atrial rate, which is a sub-optimal but non-lethal condition. The AV node of the heart will block a subset of the electrical events and maintain a normal ventricular rate. However, despite the filters in the pacemaker, the pacemaker still paces the ventricle for every 3 atrial activations, which extends fast atrial rate to more dangerous fast ventricular rate. The condition is referred to as Atrial Tachycardia Response in which the ventricular rate is increased inappropriately, thus requires revision of the pacemaker design.

Case 3: CE_{vt} is returned by $H3\_3||PM$ and corresponds to intrinsic ventricular tachycardia with fast ventricular rate. The counter-example is physiologically valid but the fast ventricular rate is due to the heart itself and is beyond pacemaker functionality. Therefore this scenario demands no revision of the pacemaker design.

Case 4: CE_{st} is returned by $H3\_5||PM$ and corresponds to sinus tachycardia, for instance, when the patient is exercising. The pacemaker improved the open-loop

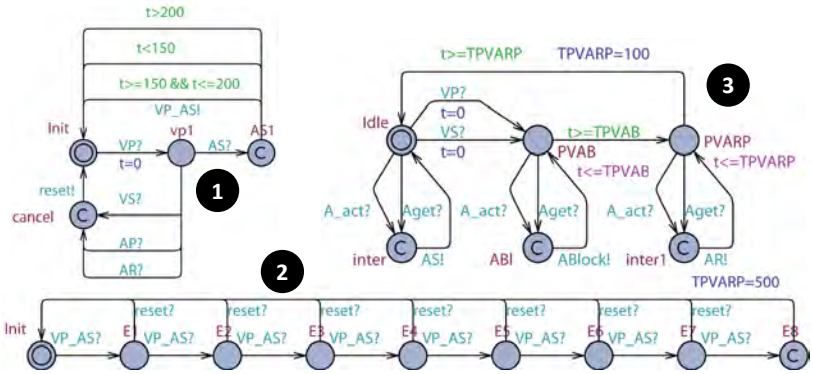


Figure 4.13: (1) The component PVAS sends VP_AS! event when a VP-AS pattern with delay between [150,200] is detected; (2) Component ELTct. After 8 VP-AS pattern, the algorithm increase TPVARP to 500ms. (3) Modified PVARP' component. TPVARP can only be set to 500 for one timing cycle.

heart condition by pacing the ventricles AVI after each atrial event, which is a correct operation of the pacemaker despite the requirement violation.

Case 5: CE_{pvc} is returned by $H4\_4||PM$ and has a very similar input-output relationship to CE_n. However, the activations of the atrial node are triggered by retrograde conduction from ventricular paces (marker cond). The atrial activations trigger another ventricular pace after AVI, which will trigger another retrograde conduction. In this case the heart rate is inappropriately high, which corresponds to a dangerous closed-loop behavior referred to as *Endless Loop Tachycardia*.

From the above result, it can be seen that abstraction tree is able to 1) refine a heart model to eliminate spurious counter-examples as in Case 1; 2) provide (multiple) physiological explanations to a counter-example as in Case 2-5; and 3) resolve ambiguities caused by abstraction as in Case 4 and 5.

Device manufacturers have developed algorithms aiming to eliminate behaviors showed in Case 2 and Case 5. In the following section we demonstrate the use of abstraction tree to evaluate the effectiveness of these algorithms.

4.4.1 Terminating Endless Loop Tachycardia

Due to the limited information the pacemaker has about the heart, the pacemaker cannot distinguish a retrograde atrial event from an intrinsic atrial event which is triggered by the SA node. From the pacemaker's point of view, the pacemaker paces the ventricles as specified for every AS. That is why open-loop testing is unable to detect this closed-loop behavior.

Modern pacemakers are equipped with anti-ELT algorithms to identify and terminate potential ELT. One common algorithm identifies ELT by the ELT pattern and terminates ELT by increasing TPVARP time once to block the AS caused by the V-A conduction. By increasing the blocking interval after a ventricular event, the pacemaker effectively ignores the early atrial signal detected due to the PVC.

ELT termination algorithm

The ELT persists without intervention and the patient's heart is forced to beat at a fast rate approaching the Upper Rate Limit. Thus, device manufacturers require a way to identify ELT and terminate it despite the limited information the pacemaker can get. The ELT detection algorithm by Boston Scientific Boston Scientific Corporation [2007a] utilizes these three features:

- Ventricular rate at Upper Rate Limit
- VP-AS pattern
- Fixed V-A conduction delay

The pacemaker first monitors VP-AS pattern with ventricular rate at upper rate limit. Then it compares the VP-AS interval with previous intervals. ELT is confirmed if the difference between the current VP-AS interval and the first VP-AS interval is within ±32ms for 8 consecutive times. Then the pacemaker increases the PVARP period to 500ms once so that the next AS will be blocked and will not trigger a VP, terminating ELT. As the V-A conduction delays are patient-specific, the algorithm compares the VP-AS interval to a previously sensed value instead of an absolute value. Since we can not store past clock values in UPPAAL, we can not explicitly model this ELT detection algorithm. However, since the conduction delay in our heart model is within a known range, we can compare the VP-AS interval with this range. The VP-AS pattern detection module $VPAS$ for our anti-ELT algorithm is shown in Fig. 4.13 (1). It detects the VP-AS pattern with ventricular rate at the Upper Rate Limit and sends out a VP_AS event if the interval qualifies.

A counter $ELTct$ counts the number of qualified VP-AS patterns. It increases the PVARP period to 500ms if eight consecutive VP-AS patterns are detected. (Fig. 4.13 (2)) The PVARP component is also modified so that the PVARP period can only be changed once by the anti-ELT algorithm. (Fig. 4.13 (3))

Verification of the algorithm

With the new pacemaker model
$P_1 = LRI \| AVI \| URI \| PVARP' \| VRP \| ELTct \| VPAS$
we first check whether the two efficacy properties still hold when the anti-ELT algorithm is introduced. We have

$$H_0 \| P_1 \| PLRI\_test \models \varphi_{LRI}$$

$$H_0 \| P_1 \| PURI\_test \models \varphi_{URI}$$

Indicating the efficacy properties still hold after introducing the anti-ELT algorithm.

Then property φ_{PMT} is checked, we have

$$H2\_4 \| P_1 \| PELT\_det \| Pvv \not\models \varphi_{PMT}$$

which indicates the algorithm successfully eliminated all ELT executions.

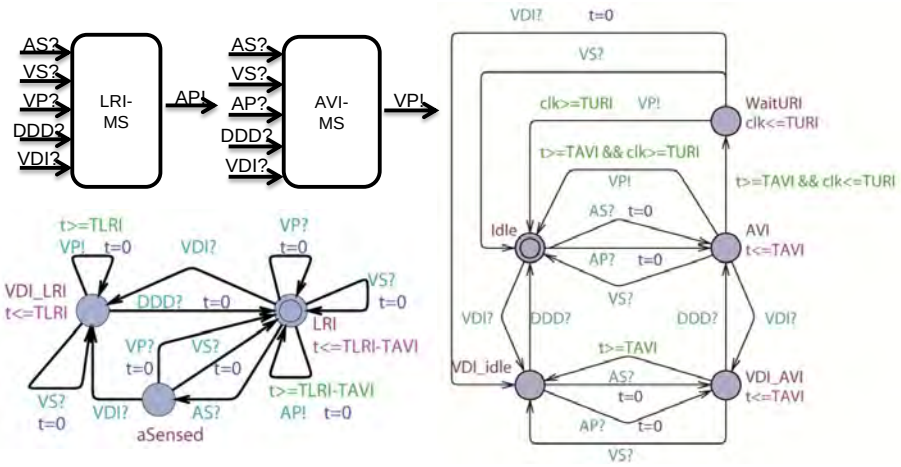


Figure 4.14: (a) After switching to VDI mode, the new LRI component LRI' maintains a minimum V-V interval; (b) After switching to VDI mode, the new AVI component AVI' keeps track of the time after each atrial events.

4.4.2 Mode Switch Operation: Atrial Tachycardia Response

Pacemaker manufacturers have designed algorithms to detect and terminate behaviors as in CE_{af}. Intuitively, the mode-switch algorithm first detects SVT. After confirmed detection, it switches the pacemaker from a dual-chamber mode to a single-chamber mode. During the single-chamber mode, the A-V synchrony function of the pacemaker is deactivated thus the ventricular rate is decoupled from the fast atrial rate. After the algorithm determines the end of SVT, it will switch the pacemaker back to the dual chamber mode. The mode-switch algorithm (also known as atrial tachycardia response) specification we use is similar to the one described in the Boston Scientific pacemakers' manual (Boston Scientific Corporation [2007b]). The algorithm first measures the interval between atrial events outside the blanking period (AS, AR). The interval is considered as *fast* if it is above a threshold (*Trigger Rate*) and *slow* otherwise. In our UPPAAL model we model it as INT (see Fig. 4.15 (1)). A counter CNT increments for *fast* events and decrements for *slow* events (see Fig. 4.15 (2)). After the counter value reaches the *Entry Count*, the algorithm will start a *Duration* (DUR) ,which is a time interval used to confirm the detection of SVT (see Fig. 4.15 (3)). In the *Duration*, the counter keeps counting. If the counter value is still positive after the *Duration*, the pacemaker will switch to the VDI mode (*Fallback mode*). In the VDI mode, the pacemaker only senses and paces the ventricle. At any time if the counter reaches zero, the *Duration* will terminate and the pacemaker is switched back to DDD mode. In our UPPAAL model of the mode-switch algorithm, we use nominal parameter values from the clinical setting. We define *trigger rate* at 170bpm (350ms), *entry count* at 8, *duration* for 8 ventricular events and *fallback mode* as VDI.

In order to model both DDD and VDI modes and the switching between them, we made modifications to the AVI and LRI components. In each component two copies for both modes are modeled, and switch between each other when switching events

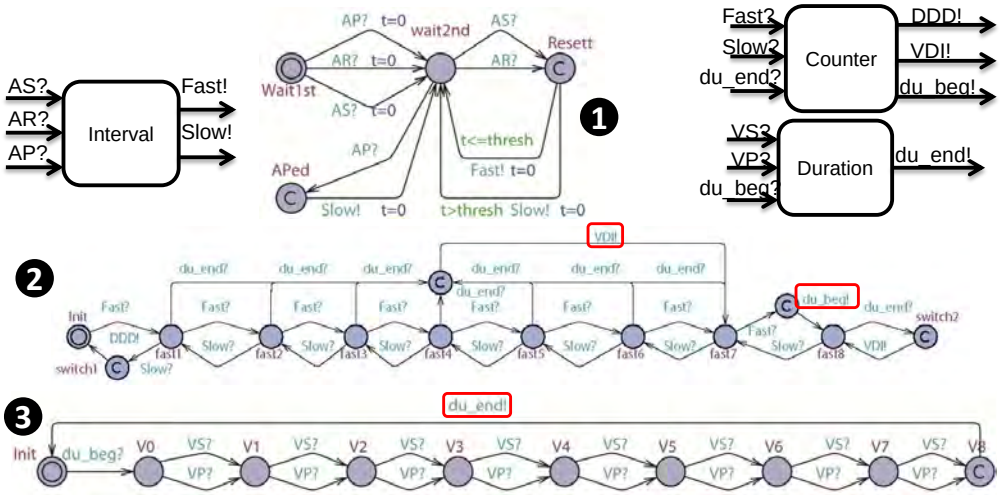


Figure 4.15: (1) Component INT: An atrial event (AS,AR) arrives before thresh after the previous atrial event is regarded as a fast event. Atrial event arrives after thresh and AP are regarded as slow event; (2) Component CNT: After 8 fast event the algorithm will start a duration by sending du_beg and will switch to VDI mode when the duration ends (du_end); (3) Component DUR :The duration length is 8 ventricular events (VS,VP)

(DDD, VDI) are received. During VDI mode, VP is delivered by the LRI component instead of the AVI component. The clock values are shared between both copies in order to preserve essential intervals even after switching. The modified AVI (AVI') and LRI (LRI')components are shown in Fig. 4.14.

Verification of the Mode-Switch Algorithm

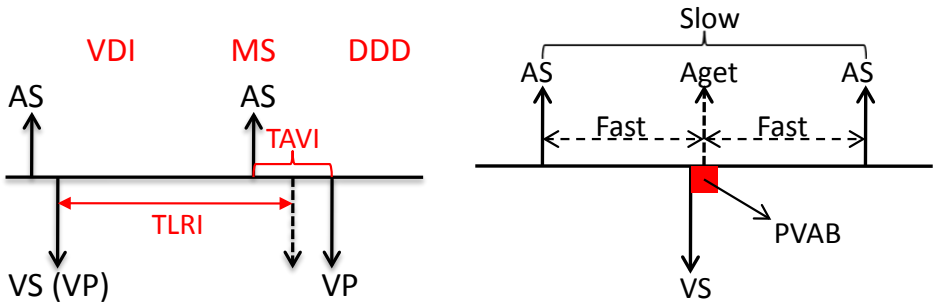
With the new pacemaker model
$P_2 = LRI' \| AVI' \| URI \| PVARP' \| VRP \| INT \| CNT \| DUR$
we first check whether the two efficacy properties still hold when the anti-ELT algorithm is introduced. We have

$$H_0 \| P_2 \| PLRI\_test \not\models \varphi_{LRI}$$

$$H_0 \| P_2 \| PURI\_test \models \varphi_{URI}$$

By analyzing the counter-example we found that when the pacemaker is switching from VDI mode to DDD mode, the responsibility to deliver VP switched from LRI component to AVI component. Since the clock reference is different (Ventricular events in LRI component and Atrial events in AVI component), the clock value for delivering the next VP is not preserved. As a result, if an atrial event which triggered the mode-switch from VDI to DDD happens within [TLRI-TAVI, TLRI) after the last ventricular event, the next ventricular pacing will be delayed by at most TAVI time, which violates the Lower Rate Limit property (Fig. 4.16(a)). Then property φ_{PMT} is checked, we have

$$H3\_2 \| P_1 \| PELT\_det \| Pvv \not\models \varphi_{PMT}$$



(a) (b)

Figure 4.16: (a) Safety Violation: VP is delayed due to the reset of timer during mode-switch, (b) Correctness Violation: The blocking period may block some atrial events, turning two *Fast* events to one *Slow* event (Jiang et al. [2012b]
)

Indicating the mode-switch algorithm failed to eliminate the PMT scenario. The counter-example trace returned by UPPAAL shows that a subset of atrial events fall into the blanking period after a ventricular event (see Fig. 4.16(b)). As a result, two fast events are reduced to one slow event and mode switch may never happen. Therefore the mode-switch algorithm in our pacemaker model can not terminate all PMT behaviors as specified as certain mild PMT events are admissible.

4.5 Related Work

Chen et. al Chen et al. [2014] extended our verification work Jiang et al. [2012b]. They developed a hybrid heart model which is able to simulate action potential at tissue level. The model is a more refined model than our Virtual Heart Model Jiang et al. [2012a], with linear dynamics on each state of the heart tissue. They also developed a probability model to simulate natural pacemaker function. They then used the combined heart model for quantitative verification of the pacemaker. However, since the pacemaker only sense the timing of the heart tissue activation, their hybrid extension for action potential does not bring much benefit but increased model complexity dramatically. As a result, they have to use bounded model checking thus sacrificed accuracy.

Jee et. al present a safety assured development approach of real-time software using pacemaker as their case study in Jee et al. [2010]. They formally model and verify a single chamber VVI pacemaker using UPPAAL and then implement it and check the preservation of properties transferred from model to implementation code.

Tuan et. al propose an RTS formal model for pacemaker and its environment and verified it against number of safety properties and timed constraints using the PAT model checker Tuan et al. [2010]. They have modeled the pacemaker for all 18 operating modes as described in Boston scientific, but their work lacks specification

and analysis of complex behaviors of the pacemaker, such as mode-switch.

Wiggelinkhuizen uses mCRL2 and UPPAAL to formally model the pacemaker from the firmware design of Vitatron's DA+ pacemaker Wiggelinkhuizen [2007]. Two main approaches have been used to investigate the feasibility of applying formal model checking to the design of device firmware. The main approach consists of verifying the firmware model in context of a formal heart model and a formal model of a hardware module which fails for high heart rates because of the state explosion. Another approach is to verify a part of firmware design which was feasible and was able to detect a known deadlock rather soon.

Macedo et. al have developed a concurrent and distributed real-time model for a cardiac pacemaker through a pragmatic incremental approach D. et al. [2008]. The models are expressed using the VDM and are validated primarily by scenario-based test, where test scenarios are defined to model interesting situations such as the absence of input pulses. The models cover 8 modes of pacemaker operation.

Gomes et. al present a formal specification of pacemaker system using the Z notation in Gomes and Oliveira [2009]. They have also tried to validate that the formal specification satisfies the informal requirements of Boston Scientific by using a theorem prover, ProofPower-Z. They have partially checked the consistency of their specification through reasoning. No validation experiment regarding safety conditions were performed yet.

Mery et. al in Mery and Singh [2009], formally model all operational modes of a single electrode pacemaker system using event-B and prove them. They use an incremental proof-based approach to refine the basic abstract model of the system and add more functional and timing properties. They use the ProB tool to validate their models in different situations such as absence of input pulses.

4.6 Discussion

Model checking is not widely use in industry, in part, due to scalability issues and also because domain expertise must be a skill possessed by the verification engineer. However, with rigorous abstraction of the system and its environment, model checking can be used to identify known and even unknown mechanisms to induce hazards. In this chapter, we use a model of a dual chamber pacemaker as an example to demonstrate the use of model checking during risk analysis. During the process we identified the need to refine the heart models to eliminate false-positives introduced during the abstraction, and demonstrated the difficulty to do so manually. The abstraction tree approach is then proposed to reduce the effort needed for both the developers and the domain experts, which makes model checking a viable approach for providing safety and effectiveness evidence.

Chapter 5

Theme 3: Verified Model to Verified Code

Model checking is performed on abstract models of the system, which is at an early stage in the development process. The verified system model during model checking is then translated into a Stateflow model, which is a step towards simulation-based testing and subsequently to code generation. Closed-loop simulation/testing are performed on more refined deterministic models, and on the actual system, complement model checking in terms of resolving ambiguities within abstract counterexamples. We aim to answer the following questions here:

- How are abstract models translated to deterministic models for simulation-based testing?
- What system level issues are best tested at the platform level?

In this chapter, we first describe an approach to automatically translate formal models that are verified in UPPAAL to Stateflow charts for simulation-based testing, and then code generated to run on an embedded platform. Following this, we demonstrate two examples that cannot be explicitly modeled using abstract semantics and must be tested.

5.1 UPPAAL to Stateflow Automated Model Translation

A model translation tool, UPP2SF (Pajic et al. [2014]) was developed to translate UPPAAL models to Stateflow (see Fig. 5.1(a)). Consider an UPPAAL model with automata $P_1, ..., P_n$. After UPP2SF translation, a two-level Stateflow chart is generated as in Fig. 5.1(b). The chart consists of parallel states $P_1, ..., P_n$ (referred to as the *parent* states) derived from the automata, parallel states $Gc\_x_1, ..., Gc\_x_m$ (referred to as *clock states*) that model all global clocks $x_1, ..., x_m$ from the UPPAAL model,

and the state *Eng* that is used as the chart's control execution engine. Moreover, the chart has predefined global data variables (and constants) with appropriate ranges and initial values derived from the UPPAAL model. Since all automata in UPPAAL are active simultaneously, the obtained Stateflow chart is a collection of parallel states with unique execution orders. Also, in every UPPAAL automaton exactly one location is active at a time. Thus, each of the parent states is a collection of exclusive states, extracted from locations in the corresponding UPPAAL automaton.

In Pajic et al. [2014], we showed that for a large class of UPPAAL models, the generated Stateflow models generated by UPP2SF preserve behaviors of the initial UPPAAL models. The translation tool can be used to estimate the worst case execution time (WCET) during modeling and model checking stage in UPPAAL, and facilitates development of modular code from timed-automata based models.

Fig. 5.2 demonstrates the Stateflow chart generated from the UPPAAL model of the DDD pacemaker model in Fig. 4.1 using the UPP2SF tool. We generated C code from the pacemaker Stateflow chart using the Simulink Coder. The code structure is shown in Fig. 5.3. The code was generated for the general embedded real-time target and as a result we obtained the main procedure, `rt_OneStep`, which processes the three input events, $VinB$, $AinB$ and clk. To ensure that the model semantics are preserved (modulo the execution time), clk input events should be created every 1ms, followed by the procedure's activation. This makes it suitable for implementation on top of a real-time operating system (RTOS).

The pacemaker code generated by the Simulink Real-Time Workshop's Embedded Coder was executed on nanoRK (Nano-RK [2007]), a fixed-priority preemptive RTOS that runs on a variety of resource constrained platforms. We tested the im-

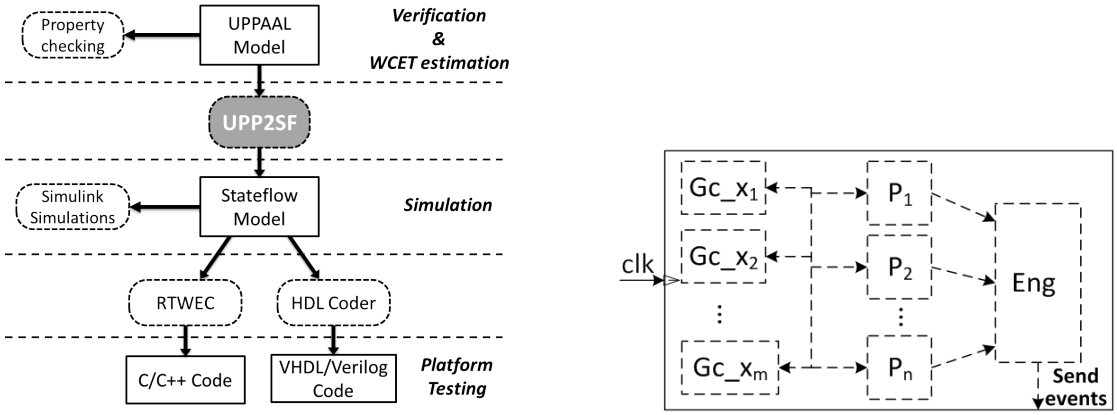


Figure 5.1: (a) Model Driven Design framework: From UPPAAL to Stateflow to generated code – covering model verification, simulation-based testing and platform testing. (b) Structure of Stateflow charts of the pacemaker's five basic timing cycles (from Fig. 4.1) derived by the UPP2SF model translator. Parent states $P_1, ..., P_n$ are derived from automata, while the *clock* states $Gc\_x_1, ..., Gc\_x_m$ model all global clocks $x_1, ..., x_m$ from the UPPAAL model. The state *Eng* is used to control execution of the chart.

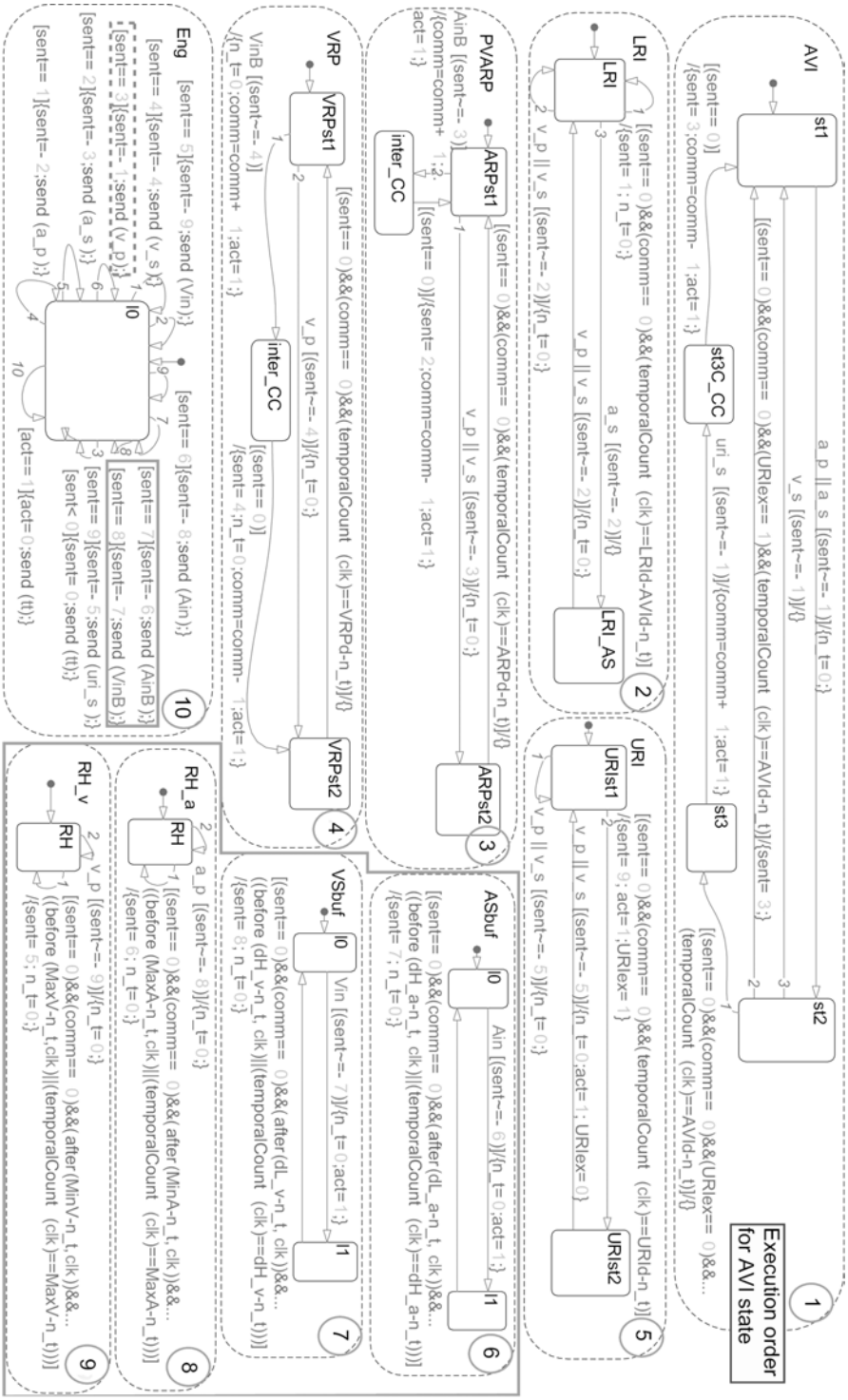


Figure 5.2: Pacemaker Stateflow chart converted from the UPPAAL model in Fig. 4.1 using UPP2SF; the heart and buffer models are highlighted.

plementation on the TI MSP-EXP430F5438 Experimenter Board interfaced with a

Listing 1. bitsForTID0 definition

```
struct {
   uint_T is_AVI:3;
   uint_T is_LRI:2;
   uint_T is_PVARP:2;
   uint_T is_VRP:2;
   uint_T is_URI:2;
   uint_T is_active_AVI:1;
   uint_T is_active_LRI:1;
   uint_T is_active_PVARP:1;
   uint_T is_active_VRP:1;
   uint_T is_active_URI:1;
   uint_T is_active_Eng:1;
   uint_T is_Eng:1;
   uint_T URI_ex:1;
 } bitsForTID0;
```

Listing 2. Rt_OneStep procedure

```
detect active inputs;
for each of the input events {
  if EventName is active  {
    sf_previousEvent = _sfEvent_;
    _sfEvent_ = EventName;
    c1_ChartName()
    _sfEvent_ = sf_previousEvent;
  }
}
update the outputs;
update the input events states;
```

Listing 3. c1_ChartName() procedure

```
increase counters for _sfEvent_;
for each parallel state {
 processState();
}
```

Listing 4. processState() procedure

```
if (rtDWork.bitsForTID0.is_active_NAME != 0) {
  switch (rtDWork.bitsForTID0.is_NAME) {
    case SubStateName1:
      /* the loop below is - checkTrans();*/
      for all transitions in ex. order {
       if transition enabled {
         execution transition actions;
         reset corresponding temporal counters;
         update rtDWork.bitsForTID0.is_NAME;

         break for
      }   }
      break;

    case SubStateName2:
      checkTrans();
      break;
    ...
    default:
      rtDWork.bitsForTID0.is_NAME=NoActiveChild;
      break;
  }  }
```

Listing 5. broadcast_tt() procedure

```
static void broadcast_tt(void) {
  int16_T sf_previousEvent;

  sf_previousEvent = _sfEvent_;
  _sfEvent_ = event_tt;
  c1_ChartName();
  _sfEvent_ = sf_previousEvent;
}
```

Figure 5.3: Structure of the pacemaker code obtained from the Stateflow chart shown in Fig. 5.2.

signal generator that provides inputs for the pacemaker code (Fig. 5.4). More details regarding UPP2SF translation and platform testing can be found in Pajic et al. [2014].

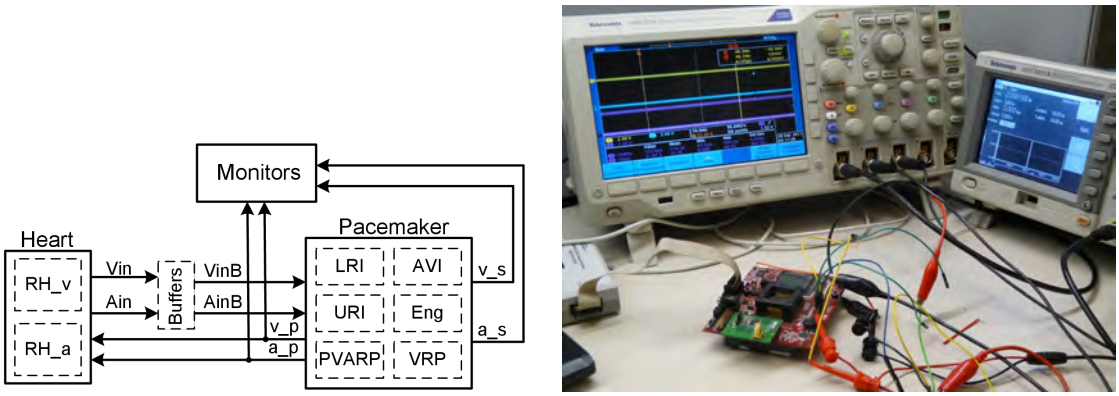


Figure 5.4: (a) Structure of the pacemaker model in UPPAAL and Stateflow, including the interaction between the pacemaker and heart, and the monitors used for verification. (b) Hardware setup with MSP430F5438 experimenters board.

5.1.1 Summary:

In this chapter, we first introduced a model translation method from UPPAAL timed automata models to Stateflow charts. Combined with Simulink coder, the tool chain provides rigorous evidence of traceability from physiological requirements to the C code implementation. Oversensing, crosstalk and lead displacement are three examples in which the cause of the problem is not modeled in the abstract interface in the timed-automata model. In such cases, closed-loop testing with the more refined heart models and EGM interface can be used to provide safety evidence.

Chapter 6

Theme 4: in-silico Pre-clinical Trials for Implantable Cardiac Devices

10,000 people in the U.S. receive an Implantable Cardioverter Defibrillator (a heart rhythm adjustment device) every month ICD [2015]. Clinical trials have presented evidence that patients implanted with ICDs have a mortality rate reduced by up to 31% Moss et al. [2012]. Unfortunately, ICDs suffer from a high rate of *inappropriate therapy*, which takes the form of unnecessary electric shocks or pulse sequences delivered to the heart. Inappropriate therapy increases patient stress, reduces their quality of life, and is linked to increased morbidity Rosenqvist et al. [1998]. Depending on the particular ICD and its settings, the rates of inappropriate therapy range from 46% to 62% of all delivered therapy episodes Gold et al. [2012].

After the verification and testing effort is completed, regulatory agencies like the F.D.A. require that the safety and efficacy of new devices be demonstrated in a *CT* (Fig. 6.1). In a trial, a group of patients that are treated with the new device (this is the 'intervention group') are compared to a group of patients who are treated with the current standard of care (e.g., a different device currently on the market; this is the 'control group'). The objective is to see whether the different devices result in significantly different effects on the patients. Clinical trials are major endeavors, involving physicians, patients, statisticians, clinical centers, companies and regulators, sometimes in several countries. Late-phase trials can run for several years, and cost millions of dollars. For example, a 2002 trial for stents lasted 2 years, enrolled 800 patients and cost $10 to $12 million and lasted 24 months Kaplan et al. [2004]. Trials also pose an inherent risk to the patients in the intervention group by exposing them to an unproven device. Thus it is crucial that they be well planned, and rigorously executed.

In reality, any trial runs the risk of errors during its planning and execution stages, which can invalidate the results of the trial. In this paper, we pose and propose an answer to the following question: *how can modeling of CPS assist in the planning*

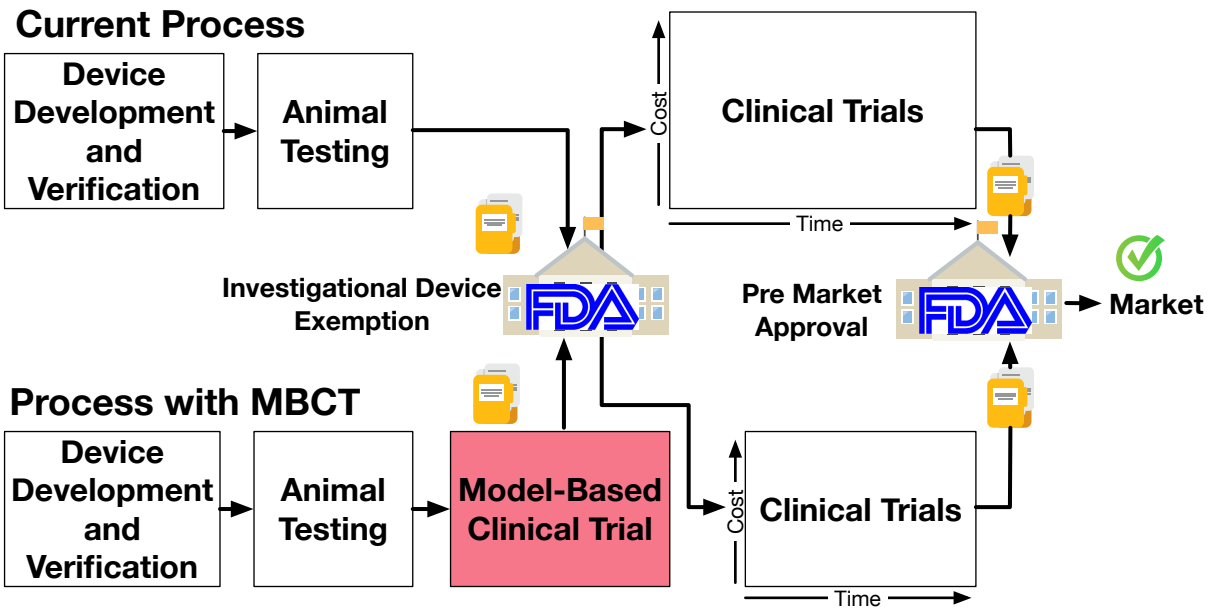


Figure 6.1: Bringing a device to market. Clinical trials are the last step before a new device's market approval. Model-based clinical trials will provide insight during planning and execution of clinical trials, leading to reduction in costs and increasing the chance of a successful trial.

and execution of a clinical trial, so as to increase the chances of a successful trial?

Most medical models today are aimed at either better understanding the phenomenon under study **?** or at device debugging and verification Jiang et al. [2012a]. There is only one case in which a computer model has been used to intervene in the regulatory process of medical devices, namely the T1 Diabetes Model (T1DM) of UVA/PADOVA **?**. T1DM models glucose kinetics in hypoglycemia, and has been accepted by the FDA as a substitute for animal trials. The T1DM has a fixed virtual cohort with 300 patients. Its objective is to test the efficacy of new glucose control algorithms by simulating them on the virtual cohort. While our models can be used in this way, our objective here is to target specific clinical trials steps and improve how they are conducted. This dictates the experimental setup and the cohort generation considerations.

The Avicenna consortium **?** lays out a vision for 'In-Silico Clinical Trials' similar to our approach. However, the emphasis in Avicenna is on individualized patient models, as they propose to customize the model to each patient enrolled in a trial. In the present work, we propose a usage of MBCT *prior* to recruitment. Thus our models need not be fitted to a given patient's data, which might be impossible, invasive, or burdensome for the conduct of the trial.

In this chapter, we demonstrate how computer models can be used for early, affordable and reproducible testing of a clinical trial's premises and assumptions. Model-based empirical validation of the premises reduces the risk of conducting a trial that

fails to demonstrate the desired effect (typically, an improvement of new intervention over the control). We used the Rhythm ID Going Head to Head Trial (RIGHT) Gold et al. [2012], which lasted five years and sought to compare the diagnostic algorithms used by two ICDs for correctly diagnosing potentially fatal tachycardias (abnormally fast heart rhythms).

6.1 Clinical trials and RIGHT

At the clinical trial stage (Fig. 6.1), the objective is no longer to find bugs in the device: it is, rather, to evaluate the safety and efficacy of the validated device on humans. RCT are the gold standard for evaluating the safety and efficacy of a new medical device L. M. Friedman and C. D. Furberg and D. L. DeMets [2010]. They constitute the only time prior to market use where the effects of the device on humans are actually observed, and are legally mandated for new high-risk medical devices like ICD. The planning and execution of an RCT requires carefully navigating a number of technical, logistical and ethical issues to obtain reliable and statistically significant results.

Because of the very high cost of RCT in terms of money, time, and the risk of harm they present to enrolled patients, our focus in this paper is on the use of CPS models, formalized in an MBCT, *to validate the assumptions made by the investigators and thus increase the chances of success of an RCT*. We illustrate our approach by applying it to the Rhythm ID Going Head-to-Head Trial (RIGHT) Gold et al. [2012], which we present next.

6.1.1 The RIGHT trial

We first provide a brief background to better understand RIGHT. Tachycardias (abnormally elevated heart rates) can be divided into VT, which originate in the heart's ventricles, and SVT, which originate above the ventricles. A sustained VT can be fatal, while an SVT is typically non-fatal. The therapy applied by the ICD often takes the form of a high-energy electric shock. The shock can be pro-arrhythmic, and was even linked to increased morbidity Rosenqvist et al. [1998]. Therefore, one of the biggest challenges for ICDs is to guarantee shock delivery for VT, and simultaneously reduce inappropriate shocks during SVT Ellenbogen et al. [2011].

RIGHT is a trial that sought to compare the VT/SVT discrimination abilities of two algorithms Gold et al. [2012]: the Rhythm ID detection algorithm found in Boston Scientific's Vitality II ICDs Boston Scientific Corporation [2007b], and the PR Logic + Wavelet (PRL+W) detection algorithm found in a number of Medtronic's ICDs (Medtronic Maximo, Marquis, Intrinsic, Virtuoso, or Entrust ICD). *Inappropriate therapy* was defined as therapy applied to an arrhythmia other than VT or VF (VF is a type of VT). RIGHT enrolled 1962 patients and ran for approximately five years. It was fully sponsored by Boston Scientific.

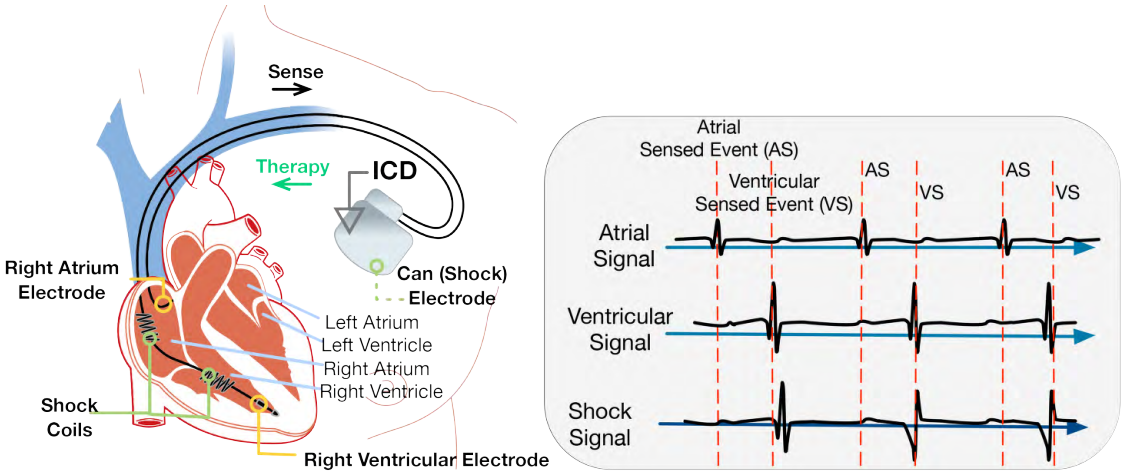


Figure 6.2: ICD connected to the heart. The atrial, ventricular, and shock electrogram signals are measured by the device, which uses them to diagnose the current state of the heart and determine whether therapy is required.

One of the trial's assumptions was that Rhythm ID would reduce the risk of inappropriate therapy by 25% over PRL+W Berger et al. [2006]. The outcome of the trial Gold et al. [2012], however, was that patients implanted with ICDs running Rhythm ID had a ***34% risk increase*** of inappropriate therapy as compared to patients implanted with ICD running PRL+W. This result is the opposite of the effect hypothesized by the trial investigators. In this paper, we design an MBCT to test early and quickly whether the hypothesized effect holds by comparing the two ICDs on a large *synthetic* cohort.

Organization: In the following sections we describe the building blocks of in-silico pre-clinical trials: modeling the heart, processing 100's of real patients' data, mapping the timing and morphology components of the signal to a heart model we developed, generating a population of 10,000+ synthetic heart models, implementing the device algorithms and conducting multiple trials for the comparative rate of inappropriate therapy, condition-level rates and evaluating the effect of device parameters on discrimination rates.

6.2 Virtual Cohort Generation

Implantable Cardioverter Defibrillators (ICDs) can diagnose VT and VF by observing the electrical activity through three channels, as shown in Fig. 6.2. The measured signals are known as *electrograms*, or EGMs. VT and SVT can share similar heart

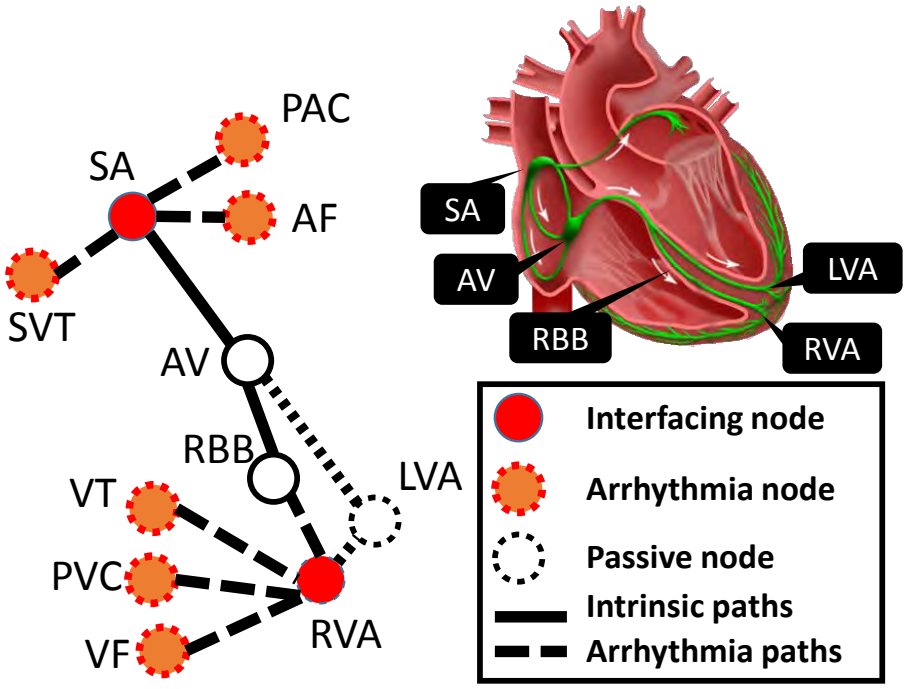


Figure 6.3: Timing model of the heart

rates and might even occur simultaneously, so an SVT can be mis-diagnosed as a VT. This is problematic because VT therapy consists of low and high energy electric shocks of 30-40 Joules (∼800V) delivered directly to the heart, which is very painful to the patient, and has been shown to increase morbidity Rosenqvist et al. [1998][1]. Therefore, one of the biggest challenges for ICDs is to discriminate between VT that typically requires a shock, and SVT that typically should not be shocked Ellenbogen et al. [2011].

An EGM signal can be characterized by the *timing of events* that produced it, and the *morphology of the signal itself*. An 'event' is roughly characterized as the source of the largest peak in the EGM (e.g. a ventricular depolarization), and event timing is a crucial element of an arrhythmia's definition in clinical Electrophysiology. The 'morphology' refers to the shape of the EGM (see Fig. 6.5 for examples). Both aspects are used by the ICD to make its decision. Correspondingly, our model has two components: a timing model, and a morphology model.

6.2.1 Timing Model

The node and path topology used in the MBCT is shown in Fig. 6.3. The hollow nodes are passive nodes representing key locations within the heart where electrical events may be blocked. These include the Atrioventricular node (AV), Right Bundle Branch (RBB) and Left Ventricle Apex (LVA). The filled nodes in red, Sinoatrial (SA)

[1]Physicians compare a shock to a "horse kicking you in the chest"

node and Right Ventricle Apex (RVA) node, represent the heart locations where ICD electrodes are placed to measure the EGMs. The timing of the activation events at these nodes determines the timing of corresponding EGMs. Different sources for tachyarrhythmias are represented by arrhythmia nodes (dashed filled nodes) which are capable of self-activating at prescribed rates. These include Premature Atrial Complexes (PACs) and Premature Ventricular Complexes (PVCs) which are sources of rhythm disturbances.

Every node and path automaton has timing parameters that determine, for example, the delay between events, and how long it takes to conduct an electrical event between two nodes. These timing parameters can be directly derived from clinical data Josephson [2008], and the model structure is compatible with clinical Electrophysiology concepts. Thus we know the ranges for these parameters. In Jiang et al. [2012a], the timing model's capability to simulate various normal and abnormal heart conditions was validated quantitatively and by cardiac electrophysiologists.

In this work we use the same heart model structure to ensure the correct timing of the EGM signals into the ICD. In order to account for inherent timing variability, during simulation the heart model randomly selects timing parameters within a prespecified range, instead of choosing specific values. By choosing the range, we control the variability of the signals produced by a given model instance.

6.2.2 Morphology Model

The ICD uses the EGM morphology in two ways: first, the atrial and ventricular EGMs are used to *sense* when events occur via peak detection (Section 6.3.1). Second, the Shock channel EGM is used in the morphology comparison discriminators (Section 6.3.2, Gold et al. [2002], C. D. Swerdlow et al [2002]). It is known that sensing (the detection of events) can be responsible for up to 20% of inappropriate therapies Daubert et al. [2008]. Therefore, it is important that our model generate realistic and varied EGM waveforms for a proper evaluation of the detection algorithms.

The timing model provides the time stamps for electrical events to happen at the interfacing nodes (SA,RVA). From path conduction we also know the source of the signals. In the heart model structure shown in Fig. 6.3 there are 5 different sources for SA node activation and 5 different sources for RVA node activation. Based on the clinical observations that electrical events from the same source produce very similar EGM morphologies, we can generate EGM signals by overlaying EGM templates corresponding to different sources onto the timing event diagram. The procedure is shown in Fig. 6.4. We also introduce small variations on EGM templates. The variations are obtained by a wavelet decomposition of the signatures followed by a random scaling of the 25% smallest coefficients. We guarantee that this does not change the signature of the EGM, by running one of the morphology comparison discriminators described in Section 6.3.2. This variation is parametrized, e.g. the percentage of modified coefficients, the range of the random scaling.

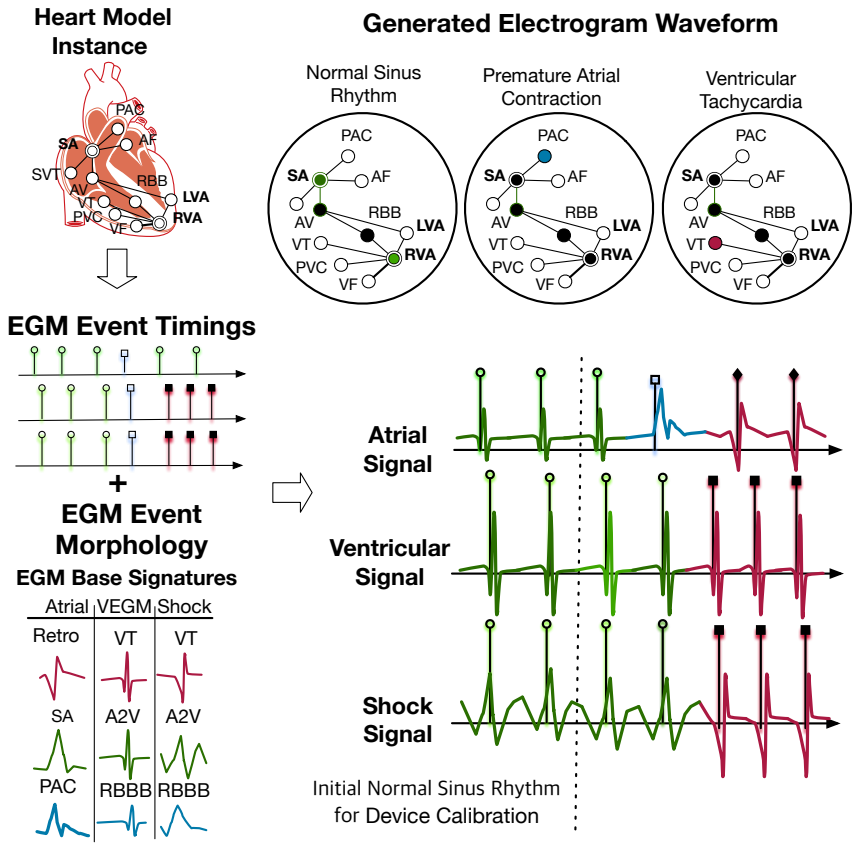


Figure 6.4: EGM waveform generation. From a given model instance and set of tachycardias, an EGM waveform is generated for the duration of an episode. The timing model determines event timings. When an event occurs, the EGM morphology for the event is output from the morphology model.

6.2.3 Patient Data Adjudication and EGM Template Extraction

In order to obtain realistic morphologies for our simulations we utilize the Ann Arbor Electrogram Libraries (AAEL), a database of over 500 EGM recordings made during clinical electrophysiology studies Jenkins and Jenkins [2003]. The AAEL is used by all major ICD manufacturers and is licensed by the US FDA. The AAEL provides descriptive annotations of records at a high level. We performed additional detailed examination to precisely segment each record according to rhythm type. 123 records from 47 patients were manually examined and adjudicated into segments called *episodes* containing one specific rhythm, e.g. NSR or VF. The adjudication was performed by a cardiologist. Fig. 6.5 (left) shows an example record (Record A185660) which has undergone this adjudication. From each episode, we developed an automated process which extracted EGMs from a given episode. The EGM are collected and organized by both patient record and by the type of rhythm which was

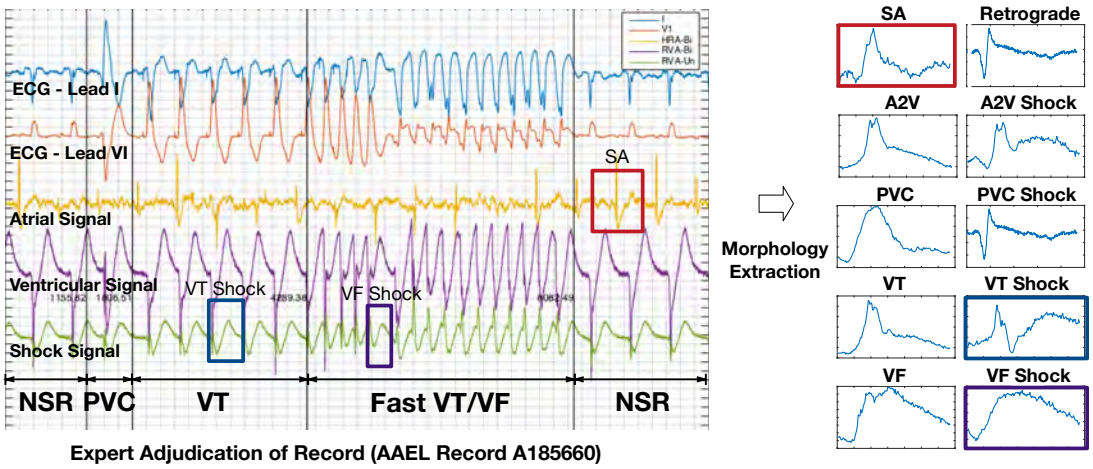


Figure 6.5: (Left) The EGM record is segmented into episodes with distinct rhythms in each. (Right) From each episode, individual EGM morphologies are extracted and stored.

annotated during the adjudication process. These extracted rhythm *signatures* provide the basis for the morphology information in the signal generated by our model. Fig. 6.5 (right) depicts an example of 10 signatures extracted from the record.

6.2.4 Cohort generation

Let $p = (p_1, \ldots, p_n) \in \Re^n$ be the vector of timing and morphological parameters of the heart model. Let $P_i \subset \Re$ be the range of parameter p_i. We generate a *synthetic cohort* of N probabilistic model instances. To produce one of these instances, for each scalar parameter p_i, we randomly select a sub-interval I_i of its range: $I_i \subset P_i$. The sub-interval I_i is chosen so that it fits with the tachycardia that this model instance is meant to simulate. E.g., for modeling VT, the rest period of the VT node might be assigned the sub-interval $I_i = [260, 280]ms$, reflecting the firing rate in the ventricles. When a model instance is simulated, each parameter p_i's value changes beat to beat by sampling it uniformly within its sub-interval I_i. Thus each generated model is probabilistic to reflect inherent rhythm variability.

6.3 Implementing Device Algorithms

Due to the limited sensing capability of ICDs, device manufacturers have developed different algorithms to identify the electrical events and correctly diagnose the cardiac arrhythmia as being VT or SVT. In this paper we implemented the detection algorithm Rhythm ID of Boston Scientific Boston Scientific Corporation [2007b], Ellenbogen et al. [2011],and PRLogic+Wavelet (PRL+W) of Medtronic Singer [2001], C. D. Swerdlow et al [2002]. We also set up a testing platform to validate our implementations against real ICDs using conformance testing.

6.3.1 Cardiac Signal Sensing

Sensing is the process by which cardiac signals measured through the leads of the ICD is converted to cardiac timing events. Appropriate sensing is essential for proper ICD detection algorithm operation which relies heavily on accurate event timing and morphology information provided by sensing. An *event* corresponds to a depolarization in the heart and manifests as a displacement from the baseline amplitude of the signal. In its simplest form, the sensing algorithm declares an event whenever the amplitude of the signal exceeds a given threshold. Once an event has been declared, the peak of the amplitude is measured and a *refractory period* begins, during which a consecutive event is ignored for a short period of time. This is to ensure that the same event is not counted repeatedly.

ICDs require a balance in sensitivity in order to operate in noisy, complex, environments where cardiac events can vary greatly in signal amplitude and frequency, such as during VF. Setting the threshold low achieves higher sensitivity to events of small amplitude, but increases the chances for incorrectly sensing other cardiac electrical artifacts such as T-waves and noise artifacts (oversensing). Conversely, setting the threshold too high allows sensing to be more robust to noise and other cardiac electrical events, but creates the potential for undersensing events of interest, such as during VF when the peak amplitude can be low.

In order to achieve adequate balance, ICD sensing algorithms are enhanced by applying dynamic adjustment of the sensitivity threshold. Initially, the threshold is raised during the refractory period after an event and once the refractory period concludes, the threshold is decayed to a minimum pre-set threshold. In our device models, we implemented the AGC algorithm of Boston Scientific and the AAS of Medtronic ICDs to incorporate dynamic threshold adjustment.

The complexity of these enhancements adds to the difficulty of properly programming device settings of ICDs and requires calibration process at the time of ICD implantation and during patient follow-up visits.

6.3.2 VT Detection Algorithm

Device companies have developed different algorithmic components to distinguish SVT from VT, referred to as *discriminators*. Each discriminator utilizes the history of timing and/or morphology of the EGM signals to determine whether the current rhythm is a VT or SVT (or neither). No single discriminator is sufficient on its own to discriminate between SVT and VT, because these classes of arrhythmias can appear similar in a number of criteria. Therefore discriminators are organized in a decision tree.We have implemented the detection algorithms Rhythm ID from Boston Scientific and PRL+W from Medtronic. This section gives an overview of both algorithms.

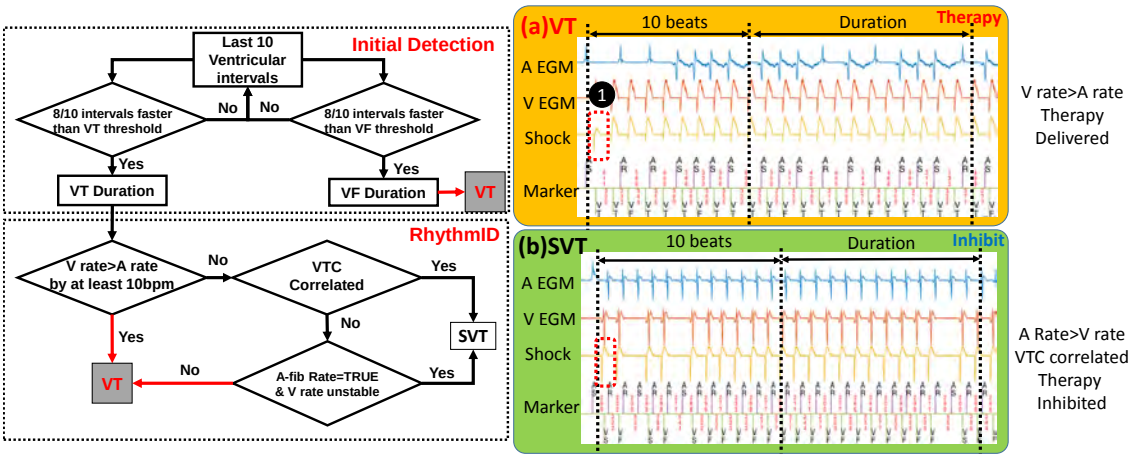


Figure 6.6: SVT/VT detection algorithm by Boston Scientific Boston Scientific Corporation [2007b]. The two cases on the right illustrate two different decisions by the algorithm. (a) illustrates a sustained VT case where at the end of the Duration, the ventricular rate is faster than the atrial rate. The algorithm correctly identified the rhythm as VT and delivered therapy. (b) illustrates a SVT case where at the end of the Duration, the ventricular rate is slower than the atrial rate. Then by comparing the EGM morphology in the Shock channel (Marker 1) with the stored NSR template (Marker 2) for the last 10 EGM events, the algorithm decided that the morphology is correlated, therefore therapy is inhibited.

Rhythm ID

Rhythm ID's decision tree is shown in Fig. 6.6. Rhythm ID detects an episode by continuously examining the last 10 ventricular intervals and comparing them with VT and VF thresholds. If 8/10 intervals are shorter than the VF threshold for a certain pre-set *VF Duration* (e.g., 2.5 seconds) then the algorithm declares VF. Otherwise, if 8/10 intervals are shorter than the VT threshold for a certain VT Duration, then further discriminators are used. First, if the ventricular rate is greater than the atrial rate for the last 10 ventricular beats, Rhythm ID will determine the condition is VT. Otherwise, the Vector Timing and Correlation (VTC) discriminator Gold et al. [2002] compares EGM morphology of the last 10 ventricular events with an EGM *template* saved during NSR. VTC is based on the assumption that the EGM morphology of the shock channel during VT is different from its morphology during SVT and NSR. If the correlation between the current EGM's morphology and the stored NSR morphology is above a pre-set threshold, the current rhythm is more likely to be SVT than VT, and therapy is withheld. Otherwise, if the atrial rate is equal to a pre-set fibrillation rate and the variance of the ventricular interval length exceeds a certain limit, the algorithm decides it's an SVT. Otherwise, it decides this is a VT.

PR Logic+Wavelet

PRL+W also utilizes rate-based and morphology-based discriminators similar (but not identical) to Rhythm ID. The morphology discriminator used in PRL+W is sim-

ilar to VTC in Rhythm ID, but operates in the wavelet domain C. D. Swerdlow et al [2002]. PRL+W also continuously compares the pattern of atrial and ventricular activation to 19 pre-defined patterns Singer [2001]. Each pattern is associated to a heart condition, like Sinus Tachycardia or VT. A match between the current activity and one of the pre-defined patterns is used as an indication that the current rhythm is explained by the associated condition.

6.3.3 Validation

Conformance testing was used to validate the software implementations of the Vitality II device by Boston Scientific. The validation hardware setup is illustrated in Fig. 6.7. 14 different scenarios were specified and programmed into an EGM Waveform generator (CRM3 Simulator, Guidant, USA) such that it would output a signal to the connected Vitality II device. The various scenarios traverse 7 out of 9 branches of the detection algorithm for Boston Scientific described in Sec. 6.3.2 and shown in Fig. 6.6. The response of the ICD interrogated using an ICD programmer (ZOOM Latitiude, Boston Scientific). As the waveform was applied to the ICD, the waveform was simultaneously acquired using a National Instruments DAQ board. The recorded waveform was then applied to the device model and response was compared. Fig. 6.7 shows an example of one such scenario, specificallly VF. In this case, the software model matched to the decision of the actual ICD which also determined that therapy should be applied.

In all scenarios, the decision of model conformed to that of the ICD. The remaining two branches were not reachable due to the limited output capability of the programmer. The Medtronic software implementation can be validated using a similar process.

6.4 Results

6.4.1 The rate of inappropriate therapy

The first objective of the MBCT is to estimate the rate of inappropriate detection $\bar{t}$ for each of the two algorithms for all arrhythmias combined, i.e., for the entire synthetic cohort. The rate of inappropriate therapy is defined as

$$\bar{t} = \frac{\text{Number of inappropriately applied therapies}}{\text{Number of applied therapies}}$$

From this we can confirm or invalidate the assumption that Rhythm ID outperforms PRL+W. We generated a synthetic cohort of 11,400 heart instances, equally distributed among the 19 arrhythmias. The number of instances was obtained from a Monte Carlo calculation.

Conclusion 1: PRL+W delivers less inappropriate therapy. The obtained rates of inappropriate detection were 6.65% for Boston Scientific and 2.91%

for Medtronic (P ¡ 0.0001), assuming an equal number of patients from each arrhythmia in the synthetic cohort. The corresponding relative improvement *of Medtronic over Boston Scientific* is 56%. In other words, the MBCT reveals that the PRL+W algorithm from Medtronic actually differentiates between VT and SVT better than Rhythm ID from Boston Scientific. Our findings are consistent with the observations of the RIGHT trial itself Gold et al. [2012].

Conclusion 2: result holds across population characteristics. The above rates were obtained under the assumption that each arrhythmia is equally represented in the cohort. A significant feature of MBCT is that it allows us to study the endpoint of interest (here, rate of inappropriate detection) on a variety of populations, which have the various arrhythmias in different proportions. This may not be feasible in a real clinical trial, which has to contend with the population present at the clinical centers where the trial is conducted. We may then ask: does PRL+W maintain a lower rate of inappropriate detection across different populations? To answer this question, we varied the distribution of the arrhythmias in the synthetic cohort, and recomputed the cohort-wide rates of inappropriate therapy. Fig. 6.8 shows the results for 10 random variations of the arrhythmia distribution. It can be seen that indeed, PRL+W maintains a better rate of arrhythmia discrimination (and by inference, less inappropriate therapy) across the board.

This illustrates very well the benefit that an MBCT can bring to the planning of an RCT: the fact that Rhythm ID could not be shown to be better than PRL+W can cause the investigators to re-consider their assumptions and the feasibility of the trial. In this case, the MBCT casts doubt on the assumed *direction* of the effect, i.e. whether intervention is better than control, or the other way around. This early check can mean the difference between an expensive trial that fails at showing the desired effect, and a trial that is appropriately sized to demonstrate the desired effect size.

Thus, while an MBCT does not replace or mimic the RCT since we can not capture patient-level outcomes of the therapy, it can provide but *early insight* at a small fraction of the RCT cost and duration and without the ethical issues.

6.4.2 Condition-level rates

Having a heart model allows us to better estimate the *sensitivity* and *specificity* of the diagnostic algorithms' performance, something which is not possible in a clinical trial because the device only records a limited number of episodes. These are defined as

$$\text{Sensitivity} = \frac{\text{Number of correctly classified VTs}}{\text{Number of true VTs}}$$

$$\text{Specificity} = \frac{\text{Number of correctly classified SVTs}}{\text{Number of true SVTs}}$$

In words, the sensitivity measures how well the device recognizes VTs. Specificity measures how well the device discriminates between VT and SVT. An ideal device

Specificity and sensitivity.

Arrhythmia	Rhythm ID	PR Logic + Wavelet	P value
	Specificity (%)		
Atrial Fibrillation	99.8	99.6	0.3167
Atrial flutter	58.3	79.33	¡0.0001
Premature ventricular complexes	100	100	1
Nonsustained ventricular tachycardia	100	99.8	0.3171
Other Supraventricular tachycardia	96.3	99.7	¡0.0001
Brady-Tachy	100	98.83	0.0079
	Sensitivity (%)		**P value**
Ventricular fibrillation	100	100	1
Ventricular tachycardia	100	100	1

would have 100% sensitivity and specificity. Unfortunately, these are typically competing goals: the more sensitive the device, the more likely it will mis-diagnose some SVTs as VTs, so its specificity will drop.

We calculated sensitivity and specificity in our MBCT, and report them in Table **??** on a per-arrhythmia basis. The conditions are drawn from RIGHT's baseline characterization Gold et al. [2012]. Specificity is reported for SVTs and sensitivity is reported for VTs. It can be seen from these results that in our synthetic cohort, Atrial flutter and other Supraventricular tachycardias are the main source of inappropriate detection for Rhythm ID compared to PRL+W. In the case of Atrial flutter, Rhythm ID categorizes it inappropriately as VT for 41.7% of the cases.

Condition-level analysis pinpoints the specific pathways of the discrimination algorithm which must be addressed to reduce the device's rate of inappropriate therapy. It is difficult to get such insight through an RCT as the patient population is fixed and the conditions are determined retroactively. Such analysis can be further used to investigate condition distributions across different patient population types (e.g. abnormal heart rhythms in children vs geographic region-specific or race-specific condition distributions).

6.4.3 Effect of Device Parameters on Discriminating Capability

ICDs have a number of parameters which can be tuned to accommodate specific patient conditions by the physicians. Currently there are very few clinical results on the effect of tuning parameters and their effect on sensitivity and specificity Moss et al. [2012]. One of the main causes of VT/SVT mis-classifications is inappropriate parameter settings Daubert et al. [2008]. In order for the physicians to set appropriate parameters, it is very important to understand how the change of one parameter can affect the discriminating capability of the device. It is costly to experiment this on real patients. With MBCT, one can use the same population across multiple devices with different parameter settings at virtually no cost.

In this section, we use MBCT to demonstrate the effects of changing two common parameters on SVT/VT discrimination specificity. The first parameter is the ***duration*** of arrhythmia before the ICD makes a therapy decision. For Boston Scientific ICD the value can be set to 1 to 30 seconds. In this experiment we explore the values {1,2,3,4,5,8,10}. The equivalent parameter for Medtronic ICD is the number of consecutive fast ventricular intervals which can be set from 8 to 20 beats. In this experiment we explore the values {8,10,12,16,18,24,30} which roughly correspond to the parameters of Boston Scientific ICD. Intuitively, with a longer duration the device can examine a longer history of the arrhythmia episode, and also allows a greater chance for the arrhythmia to self-terminate. This can prevent inappropriate therapy. Setting the duration too long can also delay and in some cases withhold appropriate therapy. These results are in agreement with the recently conducted ADVANCE-III RCT which showed that longer arrhythmia detection windows reduce shocks for Medtronic ICDs Gasparini et al. [2013].

The second parameter we varied is the ***VF threshold***. For both devices, if the ventricular rate is faster than the VF threshold for a period of time the devices will deliver therapy without going into the SVT/VT discrimination algorithm. So a higher VF threshold means that more signals are passing through the discrimination algorithm. In this experiment we explored the values {170,184,200} for both algorithms. Intuitively the higher the threshold, the more episodes will be examined by the SVT/VT discrimination algorithm, which may increase specificity.

Conducting the Model-based Clinical Trial. For each of the 21 parameter combinations described above, we ran a MBCT with 11,400 EGM episodes on both device models. From the results we observe that for both algorithms the specificity increases monotonically with the length of the duration. When the duration is longer than 5, sensitivities dropped below 100%, which is in line with the intuition. However, Rhythm ID and PRL+W displayed opposite trends for the VF threshold. For PRL+W, the specificity increased when the VF threshold was increased from 170BPM to 184BPM - i.e. a higher threshold admits more signals through the discrimination algorithm which performs better across all rates. For Rhythm ID the specificity dropped when the VF threshold was increased from 170BPM to 184BPM

- i.e. the discrimination algorithm is less effective at higher rates. One possible interpretation of the result is that the Boston Scientific algorithm is more prone to inappropriate therapies for SVTs with ventricular rate between 170BPM to 184BPM, which may be a useful for the physicians to consider during parameter settings.

6.5 Discussion

The above experiment has illustrated a practical application of the MBCT approach in the use of computer modeling for the support of clinical trial planning and execution. We now present the medically relevant limitations of this particular experiment, then discuss the MBCT approach in general. First, we did not account for post-shock detection (a phase of detection that follows the delivery of therapy), which was part of the RIGHT results. The Onset discriminator was not implemented so the results exclude its effects. RIGHT included both dual-chamber and single-chamber devices, whereas we only implemented the algorithms for dual-chamber devices. For the VTC and Wavelet algorithms, the literature did not specify which samples are taken from the electrogram. We chose to sample the electrogram uniformly in time, and validated that this gives correct results.

Clinical trials study the effect of an intervention *in the patient*, and report patient-level results (e.g., "The event of interest was observed in X% of patients in Group 1"). Our results are at the condition level: they take the form "the event of interest was observed in X% of generated conditions". To produce patient-level estimates requires an estimate of how conditions are distributed among patients. This low-level data is not readily available.

It is important to stress that in general, one should not expect *absolute numbers* from an MBCT to match those from a clinical trial, nor should this be the goal of the MBCT. For example, in this work, it is unlikely that our MBCT will yield rates of inappropriate therapy that are equal to the rates obtained by RIGHT itself. The reasons for this are many:

- Many factors that affect the outcomes of the trial (such as changes in patient lifestyle) are not modeled.
- The adjudication of episodes in RIGHT (and other trials) is limited by the fact that only therapy episodes were recorded by the devices. The adjudication process is further limited by the lack of surface EKGs, which makes it hard to reliably distinguish certain atrial arrhythmias. Neither of these is a limitation in MBCT since we have the ground truth: we know exactly what arrhythmia is being simulated by the model. Furthermore, the AAEL signals have both device electrograms (EGMs) and the corresponding surface EKGs which allow for precise adjudication.
- Experts may disagree on how to adjudicate the more complex episodes, so our classification of episodes from the AAEL database and the classification of the

RIGHT investigators have an irreducible discrepancy. Again, this will affect the statistics that they and we compute.

That said, we can expect that a good heart model will reveal *the trend* of the results, such as improvement of intervention over control or not, as shown in this paper.

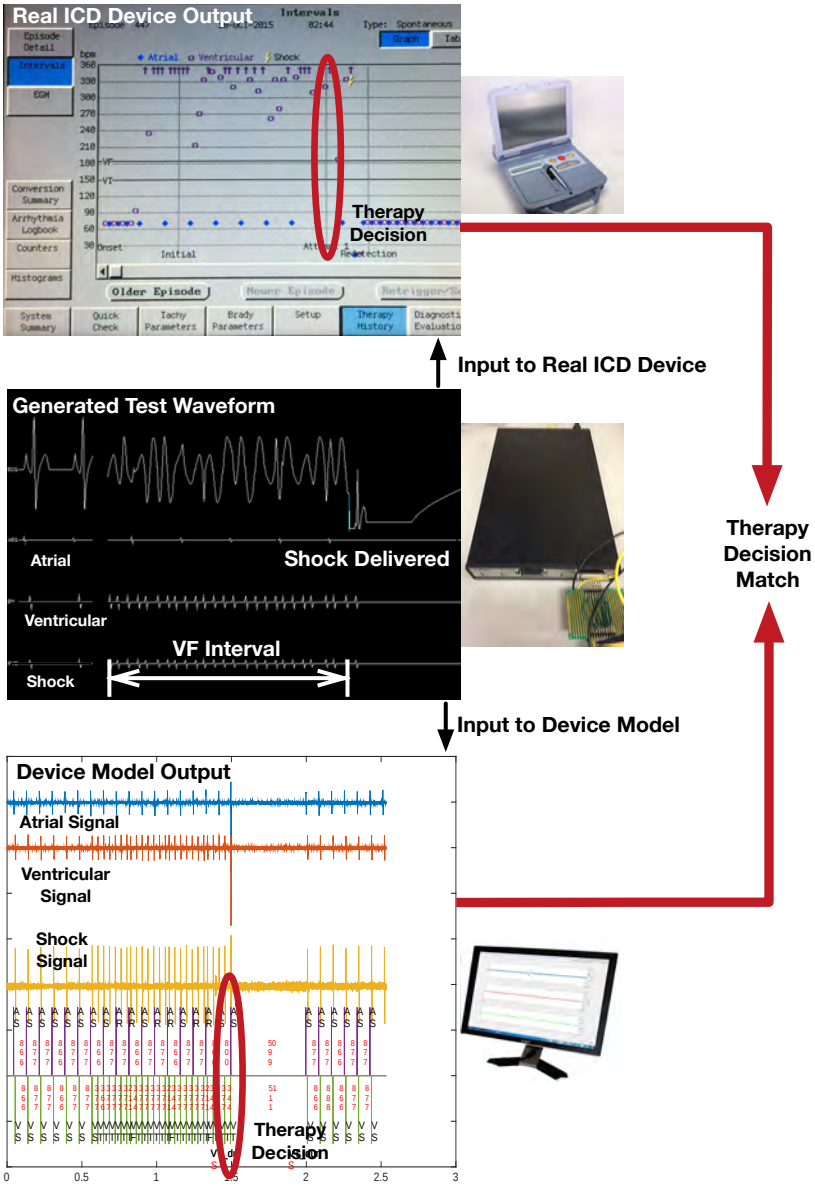


Figure 6.7: Example of validation output screenshots (Ventricular fibrillation) showing matching therapy decision for the ICD and our implementation.

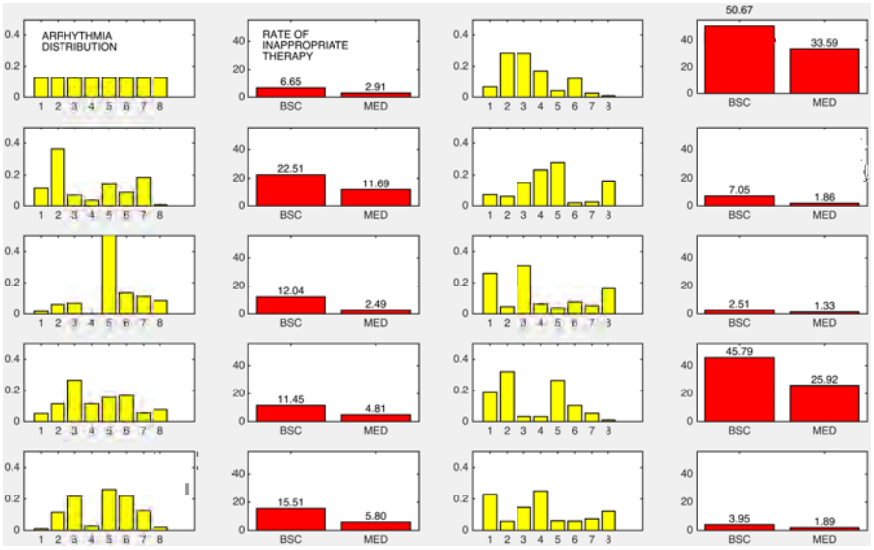


Figure 6.8: Rate of inappropriate detection (2^{nd} and 4^{th} columns) for different arrhythmia distributions (1^{st} and 3^{d} columns). The arrhythmias are (left to right on the x axis): Atrial fibrillation, Atrial flutter, Premature Ventricular Complexes, Nonsustained Ventricular Fibrillation, Supraventricular Tachycardia, Sinus Brady-Tachy, Ventricular Fibrillation, Ventricular Tachycardia Josephson [2008]. The top left distribution is uniform, and the bottom right distribution is that of the baseline characterization in RIGHT Gold et al. [2012].

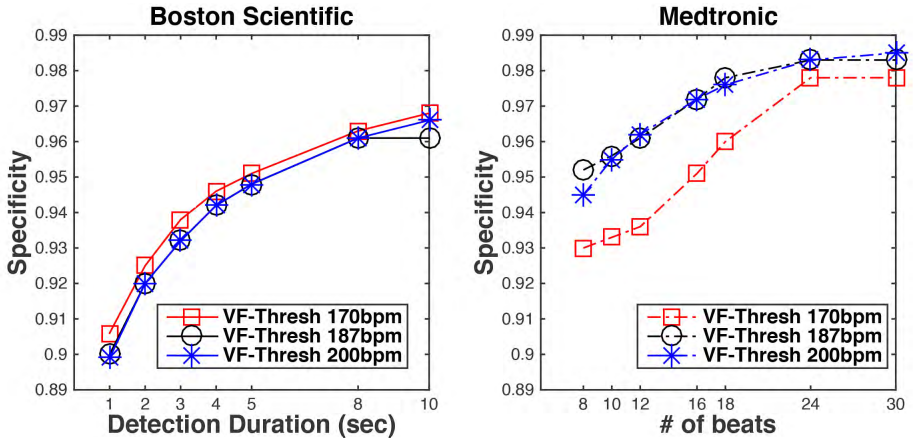


Figure 6.9: Effects of Duration and VF threshold parameters on Specificity

Chapter 7

Discussion and Open Challenges

Closed-loop medical devices like implantable cardiac devices have both diagnostic and therapeutic capabilities and interact with the patient autonomously in closed-loop. Their autonomy makes them among the highest risk devices which require the most stringent regulation. Currently, clinical trials are the primary means to identify risks associated with the closed-loop interaction between the devices and the patient. While such clinical trials are a necessity they are expensive and ineffective for verification of the safety and efficacy of medical device software.

Model-based design can potentially enable closed-loop evaluation earlier in the design process. This approach requires validated physiological models that represents the closed-loop interaction of different physiological conditions from the device's perspective. In this effort, we use an implantable pacemaker as an example to demonstrate the application of model-based design in providing safety and effectiveness towards "regulatory grade evidence" of the device and describe how these activities align into the regulatory process.

We developed heart models that capture the electrical behaviors across a range of heart conditions, and tailored the heart models for closed-loop model checking and closed-loop testing, which have different requirements. In closed-loop model checking, an abstract model of the pacemaker was validated against physiological requirements. We identified the need for heart models at different abstraction levels and demonstrated an automated approach to select the most appropriate heart model for specific safety requirements. The abstract model of the pacemaker was then automatically translated to Stateflow chart using a model translation tool and then generated into C code implementation. With this model-driven design, we are able to retain the safety properties of the modeled device from verified models to verified code. In closed-loop testing, we use more refined heart models to capture mechanisms that were not captured in the abstract heart models and evaluated pacemaker algorithms.

Eventually we aim to conduct *Model-based Clinical Trials* with automated approaches to capture and tune patient-specific electrophysiological heart models using data acquired from ablation procedures. By applying parametric and sensitivity analysis to a small sampling of real patient heart models, across a select set of cardiac

conditions, to derive a statistically significant, and physiologically relevant, population of patient models. This allows us to explore a wider range of heart behaviors and expose more corner cases to isolate software safety issues prior to an actual clinical trial. Using certified heart models, a model-based clinical trial provides additional confidence in the closed-loop safety and efficacy of medical devices prior to randomized controlled clinical trials. Model-based clinical trials for medical device software have the potential to complement the current regulatory approach by reducing the cost, scope and probability of failure of a traditional clinical trials. The area of Medical Cyber-Physical Systems is in its early days with several exciting and urgent fundamental challenges in modeling, control, verification and testing for higher confidence life-critical systems.

.1 Physiological Requirements

Physiological requirements for medical devices specify the closed-loop conditions that the device is designed to achieve with its outputs to the patient. Unlike *specifications* which specify desired device actions in response to the inputs from the patients, physiological requirements focus on the conditions of the patient with and without the device, which would indicate whether the device has fulfilled its intended goals. In this chapter, we address the following questions:

- How requirements are different from specifications?
- How can the requirements be represented?
- Are all requirements equally important?

The most basic requirement for a pacemaker is to maintain the ventricular rate above a minimum level. The corresponding physiological requirement is: *The interval between two ventricular contractions should always be less or equal to 1000ms.*

Note that this requirement focuses only on the condition of the patient (ventricular contractions), and there is no mention of the operation of the pacemaker or how the pacemaker should achieve the requirement. Here $1000ms$ is a patient-specific parameter. An example specification of a single chamber pacemaker corresponding to the requirement is: *If there is no sensed ventricular event 1000ms since the last ventricular event (sensed or paced), the pacemaker should deliver ventricular pacing.*

The specification is described using the terminology internal to the pacemaker software (paced and sensed events), and specifies the action of the pacemaker corresponding to certain inputs. For more complex requirements, multiple specifications have to work together to achieve the requirement. Therefore there may exist executions that satisfy all the specifications but not the corresponding requirement. Verifying physiological requirements requires knowledge of the physiological condition and how device interacts with the physiological environment, thus can only be performed in closed-loop.

Devices are designed to improve certain physiological conditions, the performance of the devices is evaluated on the difference between the patient conditions without the device and with the device. The device should also avoid deteriorating certain patient conditions, thus physiological requirements are specified in the form of:

$$C_{pre} \rightarrow C_{post}$$

in which C_{pre} is the physiological conditions without the device, and C_{post} is the physiological condition with the device. For model-based closed-loop verification, C_{pre} is often in form of a set of constraints on patient parameters. As a special case, C_{pre} can equal to $true$, means that C_{post} should be satisfied under all possible conditions.

One of the challenges for developing medical device software is to convert physiological requirements, which are generally informal descriptions of physiological conditions, into mathematical descriptions that can be used by verification tools. In

State	Conditions	Equivalence in VHM
NSR	Intrinsic heart rate between 60-120bpm	Trest_SA within [500 1000];
Brady	Intrinsic heart rate<60bpm	Trest_SA>1000
Sinus Tachy	Intrinsic heart rate>120bpm	Trest_SA<500
A-V Block	AV node ERP is long so that A-V conduction is slow or blocked	TERP_AV>400
SVT	Atrial rate above 200 due to reentry circuit	Interval between two Activation_SA is shorter than 250ms && Trest_SA>600

Figure 1: Patient state and equivalence in the heart model

model-based verification, these physiological conditions are mapped onto constraints on model parameters.

While the device is to aid the physiological function under certain conditions, it must deal with all possible environment conditions. However, in certain extreme conditions, not all physiological requirements can be satisfied due to the limitations of device function. It is thus intuitive to assign priorities to the requirements and assess the capability of the device to prioritize more important requirements in these scenarios.

We use pacemaker as example to demonstrate how to convert physiological descriptions into mathematical requirements and how priorities of the requirements affect the verification process.

1. Encoding Physiological Conditions: Fig. 1 lists example heart conditions and their corresponding constraints on heart model parameters.

2. Conditional Requirements: Physiological requirements are in general conditional requirements. Specifying conditional requirements enables tighter constraints on the closed-loop system. Fig. 2 shows a list of conditional requirements for normal sinus rhythm (NSR), supraventricular tachycardia (SVT), and bradycardia used during the pacemaker case study.

3. Requirement Hierarchy For a closed-loop system including a heart H and a pacemaker P: $S = H||P$ and two requirements φ_1 and φ_2 such that $\mathbb{P}_{\varphi_1} < \mathbb{P}_{\varphi_2}$, the following statement should always hold:

$$H \models \varphi_2 \rightarrow H||P \models \varphi_2$$

meaning if a higher priority requirement is satisfied by the open-loop environment, it should be satisfied by the closed-loop system as well. An example violation of the

Condition	Requirement given condition	Correspondence in VHM
NSR=true && A-V block=false	No a_p and v_p should happen	a_p=false; v_p=false
Brady=true && A-V block=false	no a_s and v_p should happen	a_s=false; v_p=false
Brady=true && A-V block=true	no a_s and v_s should happen	a_s=false; v_s=false
SVT=true && State_PM=DDD	Mode switch to VDI should happen within 5s	PM_ModeSwitch=true within 5s && PM_NextState=VDI
SVT=false && State_PM=VDI	Mode switch to DDD should happen if the atrial rate is lower than 60bpm	if Interval between two Activation_SA is larger than 1000, PM_ModeSwitch=true && PM_NextState=DDD

Figure 2: Conditional requirements for the close-loop system

Index	General Requirements	Correspondence in the close-loop system	Priority
1	No ventricular pace should happen during ventricle refractory	If State_RVA=ERP, no v_p should happen	1
2	Each atrial and ventricular event should be sensed by the corresponding lead	Interval between two Activation_RVA should between 500ms to 1000ms	2
3	Ventricular rate should be maintained between 60bpm and 120bpm	If Condition_NSR=true, Interval between two Activation_RVA shouldn't be larger than Trest_SA; If Condition_Brady=true, Interval between two a_p should be equal to LRI timer of pacemaker	3
4	Without activity sensor, the pacemaker should not increase ventricular rate above it's programmed LRI during Brady and above intrinsic heart rate during NSR	If Activation_SA=true, a_s=true; If Activation_RVA=true, v_s=true;	4
5	If the intrinsic heart rate is below some threshold, After each atrial event there should be a ventricular event within some interval(1:1 conduction)	If Interval between two Activation_SA is larger than 600, after each Activation_SA, there should be a Activation_RVA within [100, 150]	5
6	No activation conflict should happen within muscle tissue	musclepaths=path automata 1-10 in Fig 3(b); State_musclepaths should never be *Double*	6

Figure 3: General requirements for the close-loop system

statement is:

$$H \not\models \varphi_1 \&\& H \models \varphi_2 \&\& H || P \models \varphi_1 \&\& H || P \not\models \varphi_2$$

which should not be allowed. In our closed-loop verification of pacemaker, the list of physiological requirements with assigned priorities are showing in Fig. 3.

Bibliography

R. Alur and D. L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

B. P. Kovatchev and M. Breton and C. Dalla Man and C. Cobelli. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes. *Journal of Diabetes Science and Technology*, 3, 2009.

J. Beaumont, D. C. Michaels, M. Delmar, J. Davidenko, and J. Jalife. A Model Study of Changes in Excitability of Ventricular Muscle Cells. *American Journal of Physiology. 268*, 1995.

R. D Berger et al. The Rhythm ID Going Head to Head Trial (RIGHT): Design of a Randomized Trial Comparing Competitive Rhythm Discrimination Algorithms in Implantable Cardioverter Defibrillators. *Journal of Cardiovascular Electrophysiology*, 17(7):749–753, 2006.

P. Bogdan, S. Jain, and R. Marculescu. Pacemaker control of heart rate variability: A cyber physical system perspective. *ACM Transactions on Embedded Computing Systems*, 12(1s):50:1–50:22, 2013.

Boston Scientific Corporation. PACEMAKER System Specification. Boston Scientific. *Device Documentation*, 2007a.

Boston Scientific Corporation. The Compass - Technical Guide to Boston Scientific Cardiac Rhythm Management Products. *Device Documentation*, 2007b.

C. D. Swerdlow et al . Discrimination of Ventricular Tachycardia from Supraventricular Tachycardia by a Downloaded Wavelet Transform Morphology Algorithm: A Paradigm for Development of Implantable Cardioverter Defibrillator Detection Algorithms. *J. Cardiovascular Electrophysiology*, 13, 2002.

Taolue Chen, Marco Diciolla, Marta Kwiatkowska, and Alexandru Mereacre. Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation*, 236(0):87 – 101, 2014.

E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counter Example-Guided Abstraction Refinement for Symbolic Model Checking. *Journal of the ACM*, 50(5): 752–794, 2003.

E. M. Clarke and E. A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logic of Programs, Workshop*, pages 52–71, 1982.

R J. Coffey. Deep brain stimulation devices: A brief technical history and review. *Artificial Organs*, 33(3):208–220, 2009.

Macedo H. D., Larsen P. G., and Fitzgerald J. Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System using VDM. *Formal Methods*, pages 28–30, 2008.

D A. Vogel. *Medical Devices Software: Verification, Validation and Compliance*. Artech House, 2011.

J. P. Daubert et al. Inappropriate Implantable Cardioverter-Defibrillator Shocks in MADIT II: Frequency, Mechanisms, Predictors, and Survival Impact . *Journal of the American College of Cardiology*, 51(14):1357 – 1365, 2008.

K. Ellenbogen, G. N. Kay, C-P Lau, and B. L. Wilkoff. *Clinical Cardiac Pacing, Defibrillation, and Resynchronization Therapy*. Elsevier, 2011.

E.W. Hsu and C.S. Henriquez. Myocardial fiber orientation mapping using reduced-encoding diffusion tensor imaging. *Journal of Cardiovascular Magnetic Resonance*, 3(4):339–347, 2011.

P. Feiler, L. Wrage, and J. Hansson. System architecture virtual integration: A case study. *Embedded Real-time Software and Systems Conference*, 2010.

U. S. Food and Drug Administration. Design Control Guidance For Medical Device Manufacturers. *Center for Devices and Radiological Health*, 1997.

U. S. Food and Drug Administration. General principles of software validation; final guidance for industry and fda staff. *Center for Devices and Radiological Health*, 2002.

U. S. Food and Drug Administration. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. *Center for Devices and Radiological Health*, 2005.

U. S. Food and Drug Administration. Ensuring the Safety of Marketed Medical Devices: CDRH's Medical Device Postmarket Safety Program. *Center for Devices and Radiological Health*, Jan 2006.

U. S. Food and Drug Administration. Medical device recall report - fy2003 to fy2012. *Center for Devices and Radiological Health*, 2012.

U. S. Food and Drug Administration. Classification of medical devices. *US FDA documents*, 2014.

B. Fuertes and J. Toquero. Pacemaker Lead Displacement: Mechanisms And Management. *Indian Pacing Electrophysiology Journal*, 2003.

S. Fürst, J. Mössinger, S. Bunzel, T. Weber, F. Kirschke-Biller, P. Heitkämper, G. Kinkelin, K. Nishikawa, and K. Lange. Autosar–a worldwide standard is on the road. In *14th International VDI Congress Electronic Systems for Vehicles*, volume 62, 2009.

M Gasparini et al. Effect of Long-detection Interval vs Standard-detection Interval for Implantable Cardioverter-Defibrillators on Antitachycardia Pacing and Shock Delivery: The ADVANCE III Randomized Clinical Trial. *JAMA*, 309(18):1903–1911, 2013. doi: 10.1001/jama.2013.4598. URL `+http://dx.doi.org/10.1001/jama.2013.4598`.

M. Ghorbani and P. Bogdan. A cyber-physical system approach to artificial pancreas design. In *2013 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 1–10, 2013.

M. Gold et al. Advanced Rhythm Discrimination for Implantable Cardioverter Defibrillators Using Electrogram Vector Timing and Correlation. *J Cardiovasc Electrophysiol*, 2002.

M. R. Gold et al. Prospective comparison of discrimination algorithms to prevent inappropriate ICD therapy: Primary results of the Rhythm ID Going Head to Head Trial . *Heart Rhythm*, 9(3):370 – 377, 2012.

A. O. Gomes and M. V. Oliveira. Formal Specification of a Cardiac Pacing System. *In Proceedings of the 2nd World Congress on Formal Methods (FM '09)*, pages 692–707, 2009.

R. Grosu, G. Batt, F. H. Fenton, J. Glimm, C. Le Guernic, S.A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *Computer Aided Verification*, volume 6806 of *Lecture Notes in Computer Science*, pages 396–411. Springer Berlin Heidelberg, 2011.

R. G. Hauser and B. J. Maron. Lessons from the Failure and Recall of an Implantable Cardioverter-Defibrillator. *"American Heart Association, Circulation"*, pages 2040–2042, 2005.

Ask The ICD. `http://asktheicd.com`, 2015. Accessed on 10/11/2015.

Md. A. Islam, A. Murthy, A. Girard, S. A. Smolka, and R. Grosu. Compositionality results for cardiac cell dynamics. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, (HSCC '14), pages 243–252, 2014.

E. Jee, S. Wang, J. K. Kim, J. Lee, O. Sokolsky, and I. Lee. A Safety-Assured Development Approach for Real-Time Software. *The Proceedings of 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 133–142, 2010.

J. M Jenkins and R. E Jenkins. Arrhythmia database for algorithm testing: surface leads plus intracardiac leads for validation. *Journal of Electrocardiology*, 36, Supplement 1:157 – 161, 2003.

R. Jetley, S. P. Iyer, and P. L. Jones. A Formal Methods Approach to Medical Device Review. *IEEE Computer*, 39:61–67, 2006.

Z. Jiang and R. Mangharam. Modeling Cardiac Pacemaker Malfunctions with the Virtual Heart Model. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 263 –266, Sept 2011.

Z. Jiang, A. Connolly, and R. Mangharam. Using the Virtual Heart Model to Validate the Mode-Switch Pacemaker Operation. *IEEE Engineering in Medicine and Biology Society*, pages 6690 –6693, 2010a.

Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam. Real-time heart model for implantable cardiac device validation and verification. In *2010 22nd Euromicro Conference on Real-Time Systems (ECRTS)*, pages 239 –248, July 2010b.

Z. Jiang, M. Pajic, and R. Mangharam. Model-based Closed-loop Testing of Implantable Pacemakers. In *ACM/IEEE Second International Conference on Cyber-Physical Systems (ICCP'11)*, 2011.

Z. Jiang, M. Pajic, and R. Mangharam. Cyber-Physical Modeling of Implantable Cardiac Medical Devices. *Proceedings of the IEEE*, 100(1):122 –137, Jan. 2012a.

Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam. Modeling and Verification of a Dual Chamber Implantable Pacemaker. *Tools and Algorithms for the Construction and Analysis of Systems*, 7214:188–203, 2012b.

Z. Jiang, M. Pajic, R. Alur, and R. Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2):191–213, 2014.

Z. Jiang, H. Abbas, PJ. Mosterman, and R. Mangharam. Tech Report: Abstraction-Tree For Closed-loop Model Checking of Medical Devices. *http://repository.upenn.edu/mlab_papers/73*, 2015.

M. E. Josephson. *Clinical Cardiac Electrophysiology.* Lippincot Williams and Wilkins, 2008.

A. V. Kaplan et al. Medical device development: From prototype to regulatory approval. *Circulation*, 109(25):3068–3072, 2004. doi: 10.1161/01.CIR.0000134695.65733.64. URL `http://circ.ahajournals.org/content/109/25/3068.short`.

L. M. Friedman and C. D. Furberg and D. L. DeMets. *Fundamentals of clinical trials.* Springer, 2010.

K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a Nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, pages 134–152, 1997.

W. H. Maisel, M. O. Sweeney, W. G. Stevenson, K. E. Ellison, and L. M. Epstein. Recalls and Safety Alerts involving Pacemakers and Implantable Cardioverter-Defibrillator Generators. *JAMA*, 286(7), 2001.

D. Mery and N. K. Singh. Pacemaker's Functional Behaviors in Event-B. *Research report, INRIA*, 2009.

A. J. Moss et al. Reduction in inappropriate therapy and mortality through icd programming. *New England Journal of Medicine*, 367(24):2275–2283, 2012. doi: 10.1056/NEJMoa1211107.

A. Murthy, E. Bartocci, F. H. Fenton, J. Glimm, R. A. Gray, E. M. Cherry, S. A. Smolka, and R. Grosu. Curvature analysis of cardiac excitation wavefronts. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2): 323–336, 2013.

Nano-RK. Nano-RK Sensor RTOS, Carnegie Mellon University. *http://nanork.org*, 2007.

M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam. From Verification to Implementation: A Model Translation Tool and a Pacemaker Case Study. In *Proceedings of the 2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium*, RTAS '12, pages 173–184, 2012.

M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam. Safety-critical medical device development using the upp2sf model translation tool. *ACM Transactions on Embedded Computing Systems*, 13(4s):127:1–127:26, 2014.

C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart. 1. immersed elastic fibers in a viscous incompressible fluid. *Journal of Computer Physics*, 81(2):372–405, 1989.

M. Rosenqvist, T. Beyer, M. Block, K. Dulk, J. Minten, and F. Lindemans. Adverse Events with Transvenous Implantable Cardioverter-Defibrillators: A Prospective Multi-center Study. *Circulation*, 1998.

S. Rossi, R. Ruiz-Baier, L. F. Pavarino, and A. Quarteroni. Active strain and activation models in cardiac electromechanics. *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 11(1):119–120, 2011.

F. B. Sachse, A. P. Moreno, and J. A. Abildskov. Electrophysiological modeling of fibroblasts and their interaction with myocytes. *Annals of Biomedical Engineering*, 36(1):41–56, 2008.

K. Sandler, L. Ohrstrom, L. Moy, and R. McVay. Killed by Code: Software Transparency in Implantable Medical Devices. *Software Freedom Law Center*, 2010.

S. J. Saxonhouse, J. B. Conti, and A. B. Curis. Current of injury predicts adequate active lead fixation in permanent pacemaker/defibrillation leads. *Journal of the American college of Cardiology*, 2005.

R. Schulte, G. Sands, F. Sachse, O. Dossel, and A. Pullan. Creation of a Human Heart, Model and its Customisation using Ultrasound Images. *Biomedizinische Technik/Biomedical Engineering*, 46:26–28, 2001.

Igor Singer. *Interventional Electrophysiology.* Lippincott William & Wilkins, 2001.

W. Stevenson and K. Soejima. Recording Techniques for Clinical Electrophysiology. *Journal of Cardiovascular Electrophysiology*, 16:1017–1022, 2005.

N. A. Trayanova and P. M. Boyle. Advances in modeling ventricular arrhythmias: from mechanisms to the clinic. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(2):209–224, 2014.

L. A. Tuan, M. C. Zheng, and Q. T. Tho. Modeling and Verification of Safety Critical Systems: A Case Study on Pacemaker. *Fourth International Conference on Secure Software Integration and Reliability Improvement*, pages 23–32, 2010.

U. S. Food and Drug Administration. Human Subject Protection; Acceptance of Data from Clinical Studies for Medical Devices; Proposed Rule. *Docket No. FDA-2013-N-0080*, 2013.

J. E. Wiggelinkhuizen. Feasibility of Formal Model Checking in the Vitatron Environment. *Master thesis, Eindhoven University of Technology*, 2007.

S. Yamane. Timed Weak Simulation Verification and its Application to Stepwise Refinement of Real Time Software. *International Journal of Computer Science and Network Security*, 6, 2006.

S. Zhang, C. Kriza, S. Schaller, and P. L. Kolominsky-Rabas. Recalls of cardiac implants in the last decade: what lessons can we learn? *PLoS ONE 10(5): e0125987.*, 2015. doi: doi:10.1371/journal.pone.0125987.