Cyber-Physical Modeling of Implantable Cardiac Medical Devices

Zhihao Jiang, *Student Member, IEEE,* Miroslav Pajic, *Student Member, IEEE,*
and Rahul Mangharam, *Member, IEEE*

***Abstract*—The design of bug-free and safe medical device software is challenging, especially in complex implantable devices that control and actuate organs in unanticipated contexts. Safety recalls of pacemakers and implantable cardioverter defibrillators between 1990 and 2000 affected over 600,000 devices. Of these, 200,000 or 41%, were due to firmware issues and their effect continues to increase in frequency [1]. There is currently no formal methodology or open experimental platform to test and verify the correct operation of medical device software within the closed-loop context of the patient. To this effect, a real-time Virtual Heart Model (VHM) has been developed to model the electrophysiological operation of the functioning and malfunctioning (i.e., during arrhythmia) heart. By extracting the timing properties of the heart and pacemaker device, we present a methodology to construct a timed-automata model for functional and formal testing and verification of the closed-loop system. The VHM's capability of generating clinically-relevant response has been validated for a variety of common arrhythmias. Based on a set of requirements, we describe a closed-loop testing environment that allows for interactive and physiologically relevant model-based test generation for basic pacemaker device operations such as maintaining the heart rate, atrial-ventricle synchrony and complex conditions such as pacemaker-mediated tachycardia. This system is a step toward a testing and verification approach for medical cyber-physical systems with the patient-in-the-loop.**

***Index Terms*—Real-time systems, medical devices, validation, cyber-physical systems**

I. Introduction

Over the course of the past four decades, cardiac rhythm management devices such as pacemakers and implantable cardioverter defibrillators (ICD) have grown in complexity and now have more than 80,000 to 100,000 lines of code [2]. In 1996, 10% of all medical device recalls were caused by software-related issues. In June of 2006, software errors in medical devices made up 21% of recalls. During the first half of 2010, the US Food and Drug Administration (FDA) issued 23 recalls of defective devices, all of which are categorized as *Class I*, meaning there is a "reasonable probability that use of these products will cause serious adverse health consequences or death." At least six of the recalls were caused by software defects [3], [4]. There is currently no standard for testing, validating and verifying the software for implantable medical devices. Given the alarming rate at which medical device software has become a safety concern, there is an urgent need for new systematic approaches and tools to evaluate the safety of software in such devices.

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. Unlike other industries such as consumer electronics where product life cycles are measured in months, software engineering for medical devices often spans a decade and must prioritize safety and efficacy over time to market. The medical device industry is a regulated industry. The regulatory oversight reaches much further than a review of test results in a manufacturer's premarket submission and into every stage of the development process. Regulatory oversight also governs *how* the device was developed, not just *what* it turned out to be. The belief is that a well-planned, systematic engineering process produces more reliable devices, especially if software is a component of the device [5].

In safety-critical industries such as automotive electronics, avionics and nuclear systems, standards are enforced for safe software development, evaluation, manufacturing and post-market changes [6], [7]. This awareness is only beginning to enter the medical device industry [8]. However, the medical domain presents its own unique set of challenges:

1. Closed-loop context: Current evaluation of devices is open-loop and is unable to ensure the device never drives the patient into an unsafe state. Medical device testing and validation must thus be within the closed-loop context of the patient physiology. The context of the patient is a function of both the environment and the input from the device controller and must be captured by the device evaluation process.

2. Patient models: There is a scarcity of patient models and clinically-relevant simulators for device design [9]. High-fidelity models of interaction between the patient and device are needed to evaluate the safety and efficacy of device operation. Furthermore, these models must integrate the functional and formal aspects so that testing and verification are evaluated for the same patient states.

3. Adaptive patient-specific algorithms: The therapy offered by the device must adapt to the environment and specific patient's condition. There is a need for validation algorithms to ensure that device control and optimization can cover large classes of patient conditions.

A. The FDA and Medical Device Software

Before we delve into the current state of medical device software, it is useful to understand the evolution of the regulatory

The authors are with the Department of Electrical and Computer Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. E-mail: {zhihaoj, pajic, rahulm}@seas.upenn.edu.

This research has been partially supported by the National Science Foundation grants CNS-0834524, CNS-0930647 and CNS-1035715.

environment. The United States Food and Drug Administration (FDA) is the primary regulatory authority responsible for assuring the safety, efficacy and security of patients using medical devices. The history of the FDA is a reactionary one, where each stage of evolution was in response to a major health-care tragedy.

Through the course of the 1980s, software began to play an increasing role in medical devices. Software, as it turns out, is one of those technologies not anticipated by prior regulation, and was waiting for its disaster to prompt regulatory action. It was not until the 1980s when a number of cancer patients received massive X-ray overdoses during radiation therapy with the Therac-25 linear accelerator. This lead to a number of investigations, perhaps the most thorough of which was that of Leveson and Turner [10], which was rich with identified ways software could go wrong. Inadequate testing, dangerous code reuse, configuration management issues, inadequate manufacturer response, and failure to get to the root cause of the problem were among the leaders of the problems identified. The Therac-25 was an eye-opener for the FDA and legislators, and resulted in the Safe Medical Device Act of 1990. This finally required closer medical device tracking, post-market surveillance and recommendations on development, testing and validation of medical device software.

The FDA currently does not request or review the medical device software during pre-market submission. While no specific requirements or software verification standards are issued, a set of general guidelines for software evaluation are recommended [11], [12], [13]. The responsibility to test, validate and verify the device software to demonstrate its safety and efficacy is solely on the manufacturer. This is currently satisfied by the documentation of code inspections, static analysis, module-level testing and integration testing and their purpose is to establish "reasonable assurance of safety and effectiveness". These tests however fail to check for the correctness of the software and are largely open-loop tests that do not consider the context of the patient. Software is reviewed by the FDA only in the incident of a device recall. Software-related recalls are often issued in the form of *Safety Alerts* by the FDA such as "Safety alert - Pacemaker may revert to VVI mode at 70 beats/min if programmed to one of several specific ventricular pulse widths" [3].

B. Current Testing, Validation and Verification Approaches

In order to facilitate the early detection and correction of any software defects, the FDA has focused on infusion pumps due to the large number of recalls. In April 2010, the FDA began the "Infusion Pump Improvement Initiative" which offers manufacturers the option of submitting the software code used in their infusion pumps for analysis by agency experts prior to premarket review of new or modified devices." There is however a broader need for systematic and standardized testing, validation and verification of medical device software both as means to finding defects and for building confidence in the device's safety.

An effective software verification methodology is therefore needed for the risk analysis and certification of medical device software during the pre-market submission phase. While formal methods of verification are used for medical device software [14], [15], [16], testing continues to be required because it can expose different kinds of problems (e.g., compiler bugs), can examine the program in its system context, and increases the diversity of evidence available. Testing for medical device software currently is ad hoc, error prone, and very expensive. Traditional methods of testing do not suffice as the test generation cannot be done independently of the current state of the patient and organ. The primary approach for system-level testing of medical devices is unit testing using a playback of pre-recorded electrogram and electrocardiogram signals [17], [18]. This tests if the input signal triggers a particular response by the pacemaker but has no means to evaluate if the response was appropriate for the patient condition. Furthermore, this approach of "tape testing" is unable to check for safety violations due to inappropriate stimulus by the pacemaker. Pacemaker Mediated Tachycardia (PMT), a condition that is described later in this paper, is a strong example of why we need a model of the heart such as the one presented in this paper, which can be used for closed-loop system analysis. PMT is a condition where the pacemaker inappropriately drives the heart-rate toward the upper rate limit. With a tape test, PMT would not occur and the response of the pacemaker could be classified as appropriate therapy.

As the testing environment (i.e., patient condition) is not entirely under the control of the tester, the problem changes significantly as a degree of nondeterminism is introduced in the process. Implantable medical devices are a primary example of Medical Cyber-Physical Systems where the safety and efficacy of the device and device software must be evaluated within a closed-loop context of the patient. The key challenge is in the generation of physiologically relevant tests such that the device does not provide inappropriate therapy and does not adversely affect the safety of the patient. In addition, test generation must be interactive and adaptive such that the previous test stimulus affects the current state of the patient. The test generator must consider the current state when generating the next input in a way that advances the purpose of the test. The problem becomes one of the controller synthesis problems and cannot be addressed by an off-the-shelf model checker [19].

Formal methods have traditionally been used for verification of time-critical and safety-critical embedded systems [20]. Until recently, these methods have not been used for medical device certification. The authors in [21] presented the use of Extended Finite State Machines for model checking of a resuscitation device. Formal techniques have also been applied to improve medical device protocols [22] and safety [23], but the authors either used a simplified patient model or did not model the patient at all.

C. Methodology for Closed-loop Medical Device Safety

The focus of this effort is three-fold: (a) We developed an integrated functional (i.e., clinically-relevant) and formal (i.e., timed automata based) Virtual Heart Model (VHM) and a pacemaker device model for interactive and clinically relevant test generation. (b) We provide a set of general and patient

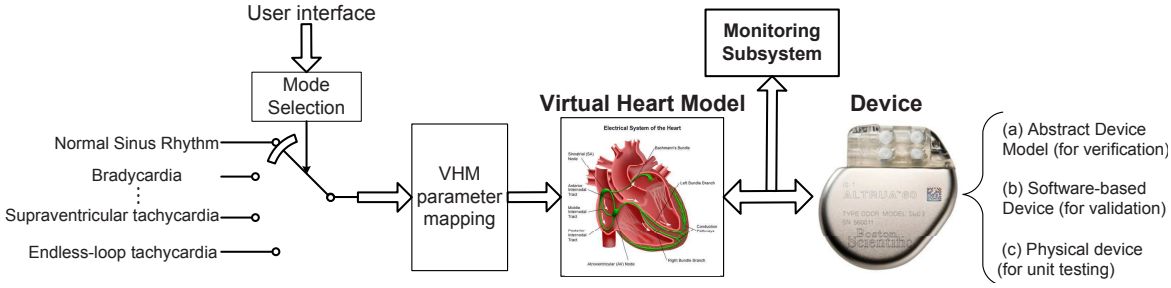


Fig. 1. Closed-loop V&V of a pacemaker. For validation, the VHM also serves as a test-generator for condition-specific testing. A similar approach is used for device verification, in which case a timed automata model of the device should be composed with the VHM in UPPAAL.

condition-specific pacemaker software requirements to ensure the safety of the patient is met under all cases, and (c) We provide a means to test and verify the closed-loop system over a variety of basic operation tests where the heart rate must be maintained, the atrial-ventricle synchrony must be enforced and complex closed-loop tests, where the pacemaker must not initiate tachycardia or perform improperly during lead displacement. With this approach of model-based testing, an executable functional model of the pacemaker is created at an early stage in the development process. This model-based methodology is an early step in addressing the urgent need for pre-market evaluation of medical device design and certification.

The rest of the paper is organized thus: We begin with an overview of model-based design for medical devices in Section II. This is followed by background knowledge of the human cardiac system in Section III, our integrated heart model and its validation in Sections IV and V. We then present our model for the pacemaker followed by a case study of the closed-loop system in Sections VI and VII. We conclude with a description of the physical implementation and a discussion.

II. Model-based Design for Medical Devices

Model-based design is a widely used and accepted approach in the development of complex and distributed embedded systems. With this approach, a model of the system plays an essential role throughout the development process. It can be used as an executable specification and virtual prototype for system development. It enables continuous validation & verification (V&V) from the early stage of development and thus reduces cost by error detection and prevention. Automated code generation can ensure faithful transformation from the model. For example, in the automotive electronics industry, AUTOSAR, the Automotive Open System Architecture [6], has united more than 100 automobile manufacturers, suppliers and tool vendors to develop a standard architecture for electronic control units (ECU). The aviation industry has a similar program called System Architecture Virtual Integration (SAVI). Developers benefit from the "Virtual Integration" at the model level to reduce cost in "Physical Integration" [7].

For medical devices, because of the strong coupling between a patient and the device, model-based frameworks that explicitly model a device's interaction with the environment and with the patient would lead to safer, higher-confidence devices [24]. Such frameworks will facilitate algorithms for medical devices that are certifiably safe for large classes of patients and can adapt to custom patients and environments [25].

The focus of this work is on the development of a system (both model-based and physical) and methods for integrated system-level testing and verification for implantable cardiac pacemakers (see Fig. 1). To address this, we model the heart and a pacemaker to expose the correct type and level of functionality and to be physiologically-relevant. In this section we provide an overview of our approach that is used for V&V of pacemakers. In this context, we define verification as the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (i.e., system specifications). Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified system requirements. One way to keep the distinction clear is to think of it this way: Verification is showing that you did what you intended to do. Validation is showing that what you intended to do was the right thing to do.

A. Previous Heart Modeling Efforts

The biggest challenge for modeling physiological systems is that the model can be built at different scales. A good model is built at the right level of abstraction for its application. To interact with implantable cardiac devices, the model of the heart should capture the electro-physiological (EP) properties of the heart (i.e., conduction and timing signals) and generate functional signals which are used as inputs to the device.

Computational and geometric heart models have been developed to study the heart functions from the electrical (e.g., signal propagation, distortion and attenuation) and mechanical (e.g., cardiac output and valve mechanisms) aspects. A high-order geometric model of human ventricles has been developed from ultrasound imagery using the Finite Element Method [26]. At the cellular level, models simulating or mimicking the ion channels activities of heart cells have been developed (e.g., [27]). These models can be used to study the pharmacological effect on heart when composed into a whole heart model. At the tissue level, the micro-structure of heart tissue has been studied in [28]. For example, the cardiac fiber direction from a canine heart has been researched in [29]. As the electrical activities of the heart influence muscle contraction, and thus control the flow dynamics of the blood,

several studies have focused on the electrical-biomechanical function of the whole heart ([30], [31]).

While these high-fidelity functional models capture the heart functions in great detail, the full electro-physiological model of the heart derived from them is computationally too expensive for implantable device V&V. Furthermore, during test case generation, fitting patient data with the large number of parameters (in the 100,000s) is nearly impossible and unnecessary as the pacemaker has only two or three electrodes to interface with the heart. Thus these models are therefore not at the right level of abstraction for V&V and do not interface with implantable cardiac devices.

Medtronic's Virtual Interactive Patient simulator can be used in closed-loop operation with real medical devices, but the lack of clinical relevance of this signal generator allows it to be only used as a training tool and not during the testing of device software itself [18]. In 1989, Malik et. al. [32] extracted the timing properties of the cardiac conduction system to model the heart. Their model was able to do close-loop simulation with pacemaker software for several clinically-relevant cases and produce template-based ECG signals. The VHM platform builds upon this modeling technique and allows for cardiac device testing and interaction with real devices.

B. Requirements for Model-Based Closed-loop V&V

For model-based V&V it is necessary to develop a framework wherein the device itself, or a model of the device, is verified or tested in closed-loop with a model of the patient or the organ of concern. Thus, the main part of the framework is the model of the patient or the organ of concern (i.e., in this case the heart) that satisfies the following requirements:

A. Model Fidelity: The design of the heart model must cover the functioning heart (i.e., normal sinus rhythm) and improper heart function including the most common and potent arrhythmias. Our Virtual Heart Model (VHM) covers the following conditions that capture a majority of closed-loop test cases: normal sinus rhythm, sinus bradycardia, Wenckebach type heat block, AV nodal reentry tachycardia (AVNRT) for supraventricular tachycardia, pacemaker mode-switch operation and pacemaker mediated tachycardia condition. In addition, pacemaker lead related issues such as crosstalk, lead dislodgement and lead displacement can be modeled in the spatial-temporal VHM model (for details see [33], [34], [35], [36], [37]. These conditions, while not exhaustive, suffice to demonstrate the closed-loop methodology for V&V.

B. Simplicity: A majority of the heart models currently used are extremely high order with hundreds of thousands of ordinary differential equations or millions of finite elements. While these models are several orders of magnitude richer than the one presented here, they are primarily concerned with the mechanics of fluid flow and muscle contraction. Furthermore, the simulation of the models are time-consuming and the models do not interact with medical devices [38], [39], [40], [41] and [42]. The VHM presents an abstraction of the timing and electrical conduction only as these are the primary inputs to the pacemaker.

C. Physical Test-bed: One of the potential problems with medical device development is that the behavior of a manufactured device might differ from the model used during its development. Thus, it is necessary to provide a closed-loop test-bed that can be used for testing of the physical devices. In this case, the model of the organ of concern that was used for model-based simulation and verification has to be compatible with the device that mimics the organ during physical testing. The VHM is implemented in Mathwork's Stateflow/Simulink models which allows extraction of VHDL description of the heart. This enabled us to operate the heart on VHDL-based FPGA platform for black-box closed-loop testing, which complements the model-based V&V.

C. Overview of the VHM Approach

We developed Virtual Heart Model (VHM), an integrated framework for implantable device validation and verification (see Fig. 2). A formal model which captures the timing properties of the electrical conduction system of the heart is developed as a kernel. With a formal model of the device, closed-loop verification can be done to evaluate device software safety against safety requirements. Through a functional interface, the heart model is able to perform closed-loop device validation by generating synthetic electrogram signals (detailed in the next section) to the devices and respond to a functional pacing signal from the device. This combination of physiological model and environmental model within a formal model framework will assist with device certification, which is based on a validated and verified device model.

Fig. 3 presents a high-level description of the approach we used to V&V a pacemaker design. We started by formally specifying the behavior of the heart, using a network of extended timed-automata (see Section IV). Afterward, the formal specification was manually mapped into two types of Simulink designs, a counter-based design and a Simulink design that uses Stateflow temporal logic operators. Both models utilize discrete-time Stateflow charts and can be used for Simulink simulation of the closed-loop scenarios, where the VHM is composed with a Simulink model of the device. However, the main difference between the models is the approach that was used to control execution of a Stateflow chart in terms of time. The former model utilizes a set of counters that count the number of global, periodically generated clock events. On the other hand, the latter model utilizes absolute-time temporal

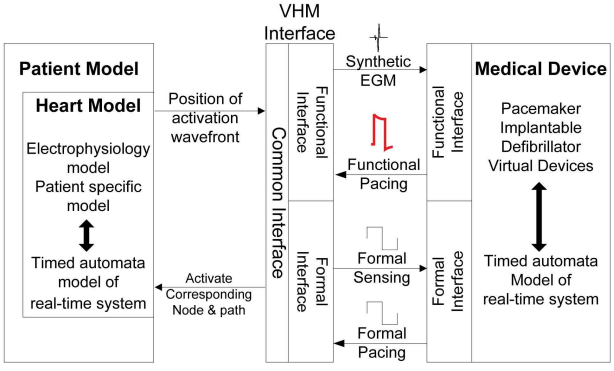


Fig. 2. Functional and Formal interfaces of the Virtual Heart Model [33].

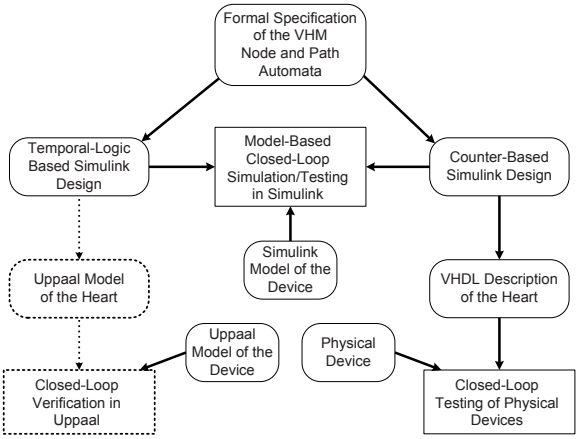


Fig. 3. High-level overview of the VHM application in closed-loop verification and validation of Implantable Cardiac Devices. The focus of our current research is denoted with dotted lines.

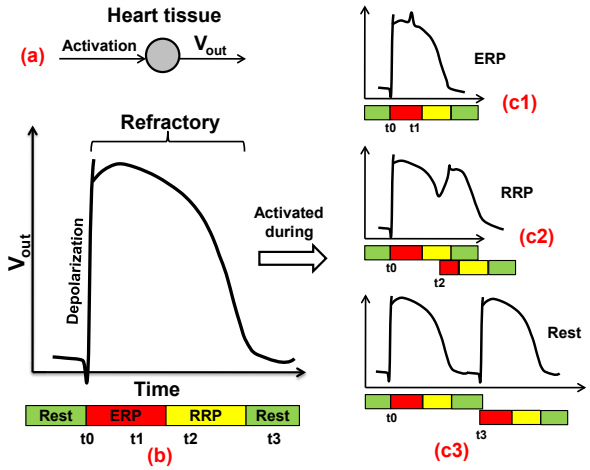


Fig. 4. (a) The generation of Action potential; (b) Action potential; (c1) The second activation arrived during ERP; (c2) Arrived during RRP; (c3) Arrived after refractory.

logic that defines time periods based on the chart simulation time [43]. Thus, this model can be translated into timed-automata description compatible with UPPAAL verification tool [44], [45].

For a particular VHM design, the behavior of the model is determined by a set of model parameters. For example, different clinical case-studies are obtained from the model by assigning appropriate values to these parameters (see Sections V and VII). Finally, given a VHM, we specify a set of general and condition-specific requirements for the closed-loop system [36]. These requirements are evaluated by constructing a set of monitors to check for violations of timing and safety conditions for each case [36]. The aforementioned setup allows utilization of the developed heart model for both verification of the closed-loop system, and simulation and testing of the system using a device model or an actual physical device.

III. Understanding the Heart Function

The human heart is perhaps the most important natural real-time system. The heart spontaneously generates electrical impulses which organize the sequence of muscle contractions during each heart beat. The underlying pattern and timing of these impulses determine the heart's rhythm and are the key to proper heart function. Derangements in this rhythm impair the heart's ability to pump blood. Thus, the heart's electrical timing, also known as its electrophysiological operation, is fundamental to proper cardiac function. Irregularities this timing result in inefficient and unsafe function of the blood-oxygen system and hence the heart rate must be maintained by artificial means. The implantable cardiac pacemaker is a rhythm management device that prevents the heart from operating below a minimum rate and maintains synchrony between the upper and lower chambers. Such devices have significantly improved the condition of patients with cardiac arrhythmias and in a majority of cases slowed down the degradation of the heart function.

A. Cellular-level Action Potential

The heart tissue can be activated by an external voltage applied to the cell. 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. 4(a)). The upstroke of the action potential is called depolarization, which corresponds primarily to the inward flow of Na^+ ions into the cell. During depolarization, the muscle will contract and the voltage change caused by the depolarization will activate the cells nearby, which causes an activation wave across the heart. After the depolarization there is a refractory period when ions flows out of the cell. The voltage is then dropped down to resting potential. The refractory period can be divided into *Effective Refractory Period (ERP)* and *Relative Refractory Period (RRP)* (Fig. 4(b)). During ERP, the cell cannot be activated due to the recovery process of the ion channels. So the activation wave will get blocked at the tissue during ERP (Fig. 4(c1)). During RRP, the ion channels are partially recovered and the cell can be activated. However, the new action potential generated by the depolarization will have different shape, thus affecting the refractory periods of the tissue and conduction delay of the activation wave (Fig. 4(c2)). Fig. 4(c1)-(c3) show the action potential shape change and corresponding timing change in refractory periods when the cell is activated at time stamp $t1$, $t2$, $t3$ after the initial activation $t0$. The slope change of the action potential will affect the time for the voltage to reach activation threshold of nearby cells, thus increases conduction delay.

B. Electrical conduction system of the heart

The electrical conduction system of the heart (Fig. 5(a)) controls the coordinated contraction of the heart. First, the specialized tissue at the Sinoatrial (SA) node periodically and spontaneously self-depolarizes. This is controlled by the nervous system and the SA node is the primary and natural pacemaker of the heart. The activation signal then travels through both atria, causing contraction and pushes blood into

the ventricles. Then the activation is delayed at the Atrioventricular (AV) node which allows the ventricles to fill fully. The fast-conducting His-Purkinje system then spreads the activation signal within both the ventricles. The simultaneous contraction of the ventricle muscles will push the blood out of the heart.

C. Cardiac Arrhythmias

The coordination of the heart's electrical activity can be impaired by the anomalies of the conduction and refractory properties in heart tissue. The disease is referred to as *arrhythmia*, which means rhythm disorder of the heart. It can be categorized into *Bradycardia* and *Tachycardia*. Bradycardia features slow heart rate which will result in insufficient blood supply. Bradycardia maybe 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 even blocked. Tachycardia features fast heart rate which would impair hemodynamics. It can be caused by anomalies in SA node or *Reentry circuit*. Reentry circuit is the most common cause for Tachycardia and is responsible for the majority of arrhythmia-related fatalities. The basic idea of reentry circuit is that additional conduction pathways form a conduction loop with the primary conduction pathways. Since the frequency for the activation signal going around the loop is higher than the heart rate generated by the SA node, the circuit will override the natural pacemaker function and results in a fast and irregular heart rate.

D. Arrhythmia Diagnosis & Treatment

Arrhythmia can be diagnosed either invasively using Electrophysiology (EP) testing or non-invasively using Electrocardiography (ECG). The EP study is used to precisely locate timing anomalies within the heart. Catheters with multiple electrodes on the tip are inserted from the groin into the heart via the blood vessels. The local potential change can be sensed by the electrodes, which generate Electrogram signals (EGMs). The common catheter placement and example EGMs are shown in Fig. 5(a) and Fig. 5(b). Using the spatial information from catheter placement, as well as the temporal information inferred from the timing difference between the pulses within one channel or among channels, the physician can locate timing anomalies within the heart. Ablation surgery can be performed to treat tachycardia if the tachycardia is caused by reentry circuit. After the additional pathway of the reentry circuit is located during EP testing, the physician can deliver RF signal from the tip of the catheter. The tissue will be killed and the pathway will be disabled.

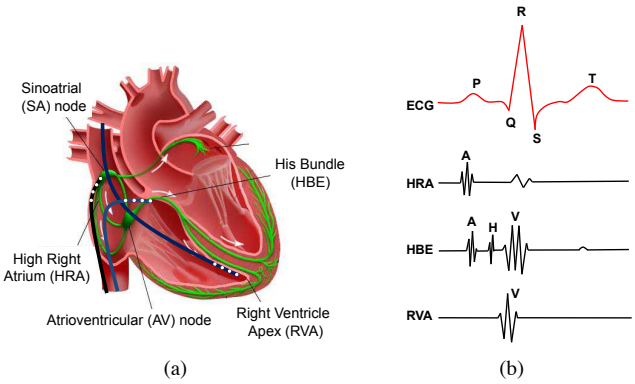


Fig. 5. (a) The electrical conduction system of the heart and basic catheter placement for EP study; (b) Example ECG and EGM signals.

The electrical activities of the heart can also be sensed on body surface as ECG. The ECG signals provide a global view of the electrical activities of the heart. Since it is non-invasive and easy to operate, it is the most commonly used technique for initial arrhythmia diagnosis. The contraction of the atria will generate the P wave. The depolarization of the two ventricles will generate high voltage QRS wave. The refractory of the ventricles will generate the T wave. (Fig. 5(b))

E. Rhythm management devices

Since the heart tissue can be activated by an external voltage, devices like implantable pacemakers have been developed to deliver timely electrical pulses to the heart to treat Bradycardia. The pacemaker has two leads inserted into the heart, one in the right atrium and one in the right ventricle. By doing timing analysis of the EGM signals sensed from the two leads, the artificial pacemaker generates electrical pulses when necessary that can maintain ventricular rate and enforce atrial-ventricular synchronization.

IV. Heart Model

In this section, the formal specification of the VHM is presented. The model has been manually translated into two Simulink designs: a counter-based model (more details regarding the model can be found in [33]), and a temporal logic based model [46].

The electrical conduction system of the heart consists of conduction pathways with different conduction delays and refractory periods. Since refractory properties of a conduction path are determined by the refractory properties of the tissue at its two terminals [47], a conduction path can be modeled with two "node" components that model refractory properties and a "path" component modeling conduction properties between the two nodes. Since the refractory and conduction properties are all timing based, it is natural to model the electrical conduction system as a network of timed-automata [48], which are widely used for model verification. Several verification tools based on timed automata have been developed, including UPPAAL [44] and Kronos [49].

A. A Brief Overview of Extended Timed Automata

Timed automaton [48] is an extension of a finite automaton with a finite set of real-valued clocks. The formal description of the VHM uses an extended version of timed automata semantics, which is similar to the semantic extension used in UPPAAL [44], [45]. The value of all clocks increases over time at the same rate. Each location (i.e., state) can be assigned with a set of *clock invariants* which are conditions

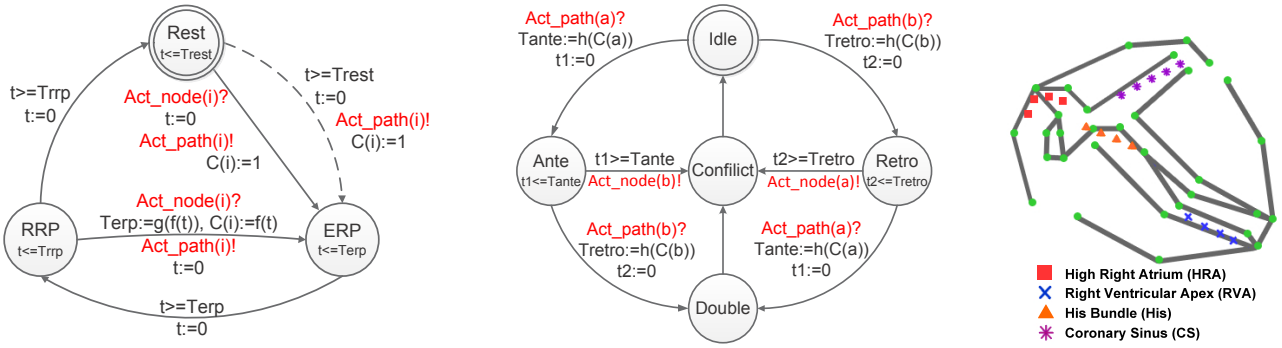


Fig. 6. (a) Node automaton. Dotted transition is only valid for pacemaker tissue like SA node; (b) Path automaton; (c) Model of the electrical conduction system of the heart using a network of node & path automata [33].

expressing constraints on the clock values for the location. In most models, a state invariant defines an upper bound on the values that a clock can have while the state is active.

A transition *guard* is a condition on the values of clocks. A typical guard is of the form $t \geq T$, which provides a lower bound for the clock value. A transition between locations is enabled when the guard of the transition is true. However, a transition between locations can occur at each moment when it is enabled. Thus, to model deterministic transitions at a particular time, state invariants are usually defined as the closure of the complement of the guard (e.g., if the guard is defined as $t \geq 1$, then an invariant $t <= 1$ is added to the state). Finally, when a transition occurs, associated *actions* are taken, which involve updating local variables and/or reseting clocks.

The VHM is modeled as a network of extended timed-automata that includes synchronization primitives and shared variables. In addition, in the VHM each automaton has its own clocks and all clocks progress synchronously. Automata synchronize with each other using broadcast channels. A channel c synchronizes between a sender $c!$ and an arbitrary number of receivers $c?$. A transition with receiver $c?$ is taken if $c!$ is available. Since shared variables are used, as with UPPAAL, linear operations and conditions that include variables can be used as a part of guards and reset actions. However, unlike timed-automata semantics in UPPAAL, in our framework a variable can be assigned with a value that is obtained using a non-linear function of variables and clocks. This is done to be able to model the refractory period and conduction delay change. However, this aspect complicates the use of standard verification tools based on the timed-automata framework.

B. Modeling the Electrical Conduction System of the Heart

Using the aforementioned semantics, we define *node automaton* that models the refractory properties of heart tissue, and *path automaton* that models the propagation properties of heart tissue. Heart tissue along a conduction path can then be modeled using two node automata and one path automaton connecting them. When one of the node automata is activated, it will send an *Act_path!* event to the path automaton. The path automaton, which models a conduction delay, generates an *Act_node!* event after the conduction is completed. The event will activate the node automaton at the other end of the conduction path. The activation signal will keep propagating if the node automaton is connected to other path automata. In this manner, the electrical conduction system of the heart can be modeled as a network of node and path automata (see Fig. 6(c)).

1) Node automaton: The node automaton with index i (presented in Fig. 6(a)) is used to mimic the timing behavior of cellular action potential from Fig. 4(a). The refractory periods are modeled as three states. The automaton starts from *Rest* state, which corresponds to the resting potential of the action potential. For the specialized tissue like SA node, the corresponding node automaton will be self-activated into *ERP* state after *Trest*. A broadcast event *Act_path(i)!* is sent out to all path automata that are connected to node automaton i. In this case the shared variable $C(i) \in (0, 1]$, which is shared among all paths connecting to node i, is updated to 1, indicating a normal conduction delay. All node automata can be activated by receiving event *Act_node(i)?* after some path connecting to it finishes conduction.

ERP state serves as a blocking period since the node does not react to activation signals while the state is active. After *Terp* time in *ERP* state, the transition to *RRP* state occurs. If no external stimuli occurs, the node will return to *Rest* state after *Trrp* time. If a node is activated during *RRP* state, the transition to *ERP* state will occur, activating all paths connected to the node. Before entering the *ERP* state, the variable used for the the clock invariant of the state is modified. This behavior corresponds to the ERP change due to early activation. The conduction delay of the paths connecting to the node will also be updated by the value of shared variable *C(i)*. The physiological basis and clinical data of this behavior has been studied in [50] and [51] and an exponential approximation of the changing trend has been made to approximate similar behavior.

The functions f and g that are used to mimic the change in *Terp* are defined as:

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

The AV node has a different profile than the other tissue. The ERP period increases rather than decreases when activated

during its RRP [47].

$$g(x) = \begin{cases} T_{min} + (1-(1-x)^3) \cdot (T_{max} - T_{min}), i = AV \\ T_{min} + (1-x^3) \cdot (T_{max} - T_{min}), i \neq AV \end{cases} \quad (2)$$

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

2) Path automaton: The path automaton models the electrical conduction between two nodes. Path automaton connecting nodes a and b is designed as in Fig. 6(b). Its initial state is *Idle* state, which corresponds to no conduction. It is worth noting that the path automaton is able to conduct both ways. 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 *Act_path* event is received from one of the nodes connected to it, the a transition to *Ante* or *Retro* state correspondingly will occur 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:

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

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)* repectively. A transition to *Conflict* state occurs followed by the transition to *Idle* state. The intermediate state *Conflict* is designed to prevent backflow, 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 and the transition to *Idle* state is taken.

3) Geometric model of the heart: The node and path automata are overlaid onto a 2D heart anatomy to provide rough information about the model topology and relative path length. A more detailed 3D anatomical model of the heart is currently being developed. The new model will have more geometric and anatomical details to allow simulation of more complex clinical cases with high fidelity. It is also essential for developing a patient-specific heart model. The 2D geometric model also limits the flexibility to measure electrical activities at precise locations of the heart and the morphology of EGM signals have low fidelity. However, since the morphology of EGM signals have little influence on pacemaker function and the pacemaker leads are fixed, the interface is good enough for our current application.

C. Functional interface

To test implantable cardiac devices like a pacemaker, the VHM has to be able to generate EGM signals which are the inputs to the devices. According to [52], during EP study, a potential difference can be sensed when the activation wavefront passes by the electrode on catheters. The same mechanism applies to the pacemaker leads. Thus, a functional interface has been developed and we use *probes* to represent electrodes on the a catheter. The probe is able to generate synthetic EGM signals using temporal information of the formal kernel and spatial information from the 2D geometric model. The same idea can also be extended to the 3D model. Fig. 7 shows the morphology of EGM signal changes with different conduction velocity and probe configurations. Due to space limitation, detailed description of the probe model can be found in [35].

D. Parameter estimation

In order to model the heart in clinical cases, parameter estimation from patient data is an essential step after the topology of the model is defined. During EP study, refractory periods and conduction delays of the heart tissue are estimated from EGM signals by the physicians to identify potential anomalies. These two features are similar to the model parameters of interest. Ideally all model parameters can be extracted from EGM data from EP studies. However, EGM signals contain only partial information and some of the parameters cannot be exactly extracted. The techniques used to extract timing parameters in EP studies are introduced in the following section.

V. Heart Model Validation

By modeling actual clinical cases, the functional outputs of the VHM were validated by the director of cardiac electrophysiology in the Philadelphia VA Hospital and electrophysiologists in the Hospital of the University of Pennsylvania [33]. In this section, we first explain how parameter estimation is done during electrophysiological (EP) study. Then a VHM model, using the parameters extracted from the clinical data, is constructed and is able to generate similar data (i.e. similar to an actual patient condition).

A. Electrophysiology Study

1) Catheter placement: During EP study, catheters, with multiple electrodes on their tip, are inserted into the heart

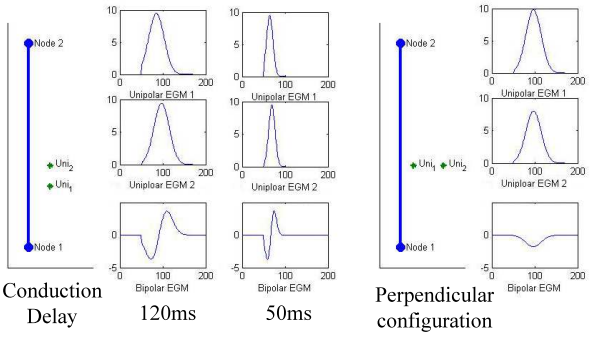
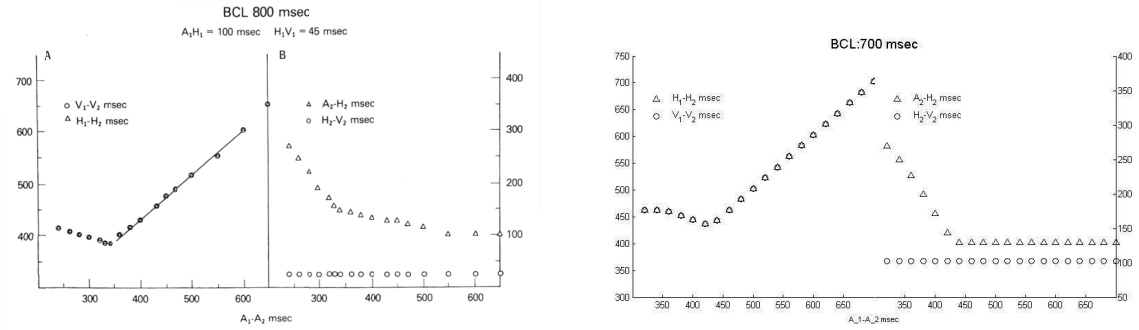


Fig. 7. The influence of conduction velocity and probe configuration on the EGM morphology. The left columns show the placement of probes in relation to the path; the right columns show the functional EGM.



(a) Testing result for a real patient [47]

(b) Testing result for VHM simulation

Fig. 8. Key interval values when the coupling interval shortens for (a) a real patient and (b) in VHM simulation [33].

and placed against the inner heart wall. The typical catheter positions used are *Hight Right Atrium (HRA)*, which is placed near the SA node and monitors its activity; *His Bundle Electrogram (HBE)*, which is placed across the valve between atrium and ventricle and is able to sense local electrical activation from atrium, His bundle and ventricle; *Right Ventricle Apex*, which is placed at the right ventricle apex to monitor electrical activity of the ventricle. The catheter placement is shown in Fig. 5(a) and the corresponding probe placement is shown in Fig. 6(c). Since HBE catheter monitors the electrical activities from atrium (A), His bundle (H) and ventricle (V) (Fig. 5(b)), it is often used to evaluate the conduction properties along the conduction path from atrium to the ventricle.

2) Extrastimuli Technique: The nervous system and the heart form a closed-loop system to maintain the appropriate intrinsic heart rate. In order to evaluate heart function we need to "open the loop" and isolate the heart. During EP study, a sequence of external pacing signals is delivered from the HRA catheter at a rate slightly faster than the intrinsic heart rate. The interval between two consecutive pacing signals is referred to as Basic Cycle Length (BCL). This sequence of pacing will override the SA node function, thus, functionally isolating the heart from the nervous system. The pacing sequence can also drive the heart into a known state since the time of the last pacing signal will be the start of the ERP of the SA node. After the initial pacing sequence, another pacing signal is delivered which is referred as *extrastimulus*. The interval between the extrastimulus and the last pacing signal of the pacing sequence is referred to as *Couping interval*. By decreasing the coupling interval gradually, the extrastimulus will reach the RRP of the tissue, causing changes in conduction delays. The ERP of the tissue is eventually reached and no conduction can happen. The extrastimuli technique is a common clinical practice to estimate the refractory periods for heart tissue.

B. Clinical case study

Fig. 8(a) shows the testing result of a real patient using the extrastimuli technique. The intervals are measured from the HBE catheter. A_1, H_1, V_1 are the atrial, His bundle and ventricular pulses on the HBE electrogram respectively, which are caused by the last pulse of the pacing sequence. A_2, H_2, V_2 are the pulses caused by the extrastimulus. The interval $A_1 - A_2$ on the x-axis is equal to the coupling interval. On the left side the $H_1 - H_2$ interval and the $V_1 - V_2$ interval remain equal during the test, indicating that the conduction delay between the His bundle and the ventricle is not affected by shortening the coupling interval. This can be confirmed by the unchanged $H_2 - V_2$ interval on the right side of the figure. When the coupling interval is larger than 350ms, the intervals $A_1 - A_2$, $H_1 - H_2$ and $V_1 - V_2$ remain equal. As the coupling interval decreases, the $A_2 - H_2$ interval start increasing after $A_1 - A_2$ interval reduced to 350ms, indicating the RRP of AV node reached. The conduction delay is also revealed by the $H_1 - H_2$ and $V_1 - V_2$ intervals on the left side.

The result of this test indicates that for this particular patient, AV node has the longest refractory period on the conduction path from atrium to the ventricle. The total refractory period (ERP+RRP) of AV node is around 350ms and the RRP is as long as 70ms. Since no conduction block happened during the test, the ERP period of AV node cannot be determined. The conduction delays of heart tissue can be determined by the interval between pulses in EGMs. Extrastimuli technique will not provide us with all the parameters of VHM since it will only be performed at select locations of the heart tissue during EP study.

C. VHM Simulation

By setting the total refractory period of the AV node automaton, the longest among other automata (Fig. 9), the VHM is able to generate similar result with extrastimuli technique (Fig. 8(b)).

D. Other Validation Case Studies

Other case studies have been introduced in detail in our previous work. Wenckebach type AV nodal response has

Node	ERP Range (ms)	RRP (ms)
SA	150-200	100
AV	230-300	200
His	300-350	50
Ventricle	200-290	100

Fig. 9. Node automata parameters for the case study.

been modeled in [33]. Tachycardia due to reentry circuit like Atrioventricular Nodal Reentry Tachycardia (AVNRT) and Atrial Flutter has been modeled in [33] and [34].

VI. Pacemaker Model

As part of the model-based design, it is very important to have a formal model of the device software. In our study, we focus on the implantable pacemaker since it is relatively simple among implantable cardiac devices as its functionality is based only on timing and does not consider signal morphology. This serves as a good base case to demonstrate the proposed methodology.

The artificial pacemaker is designed for patients with Bradycardia. 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 timing diagram

The timing diagram of a DDD pacemaker is shown in Fig. 10. There are 5 basic timing cycles which diagnoses heart condition from sensed events (AS, VS) and deliver appropriate pacing events (AP, VP). Five corresponding timing constants are programed by the physicians according to patient condition. They are denoted as TLRI, TAVI, TURI, TPVARP and TVRP in our description. The basic functions of these timing cycles are: maintaining heart rate, maintaining A-V synchrony, preventing inappropriate fast pacing and filter noises.

The Lowest Rate Interval (LRI) component is initialized by ventricular events (VS, VP). It counts the time elapsed after the ventricular event and deliver Atrial Pacing (AP) if no atrial sense (AS) happened within TAEI, which equals to TLRI-TAVI. (Marker 1 in Fig. 10) An atrial event (AS, AP) initializes the Atrio-Ventricular Interval (AVI) component. It counts the time elapsed after the atrial event and deliver ventricular pacing (VP) if no ventricular sense (VS) happens within TAVI time. This will maintain synchronization between atria and the ventricles. If after TAVI the time between current time and the last ventricular event is less than TURI, the ventricular pacing is suspended until the end of TURI (Marker 3). This will prevent the pacemaker from pacing the ventricle at interval shorter than the Upper Rate Interval (URI). After each ventricular event, a Post Ventricular Atrial Refractory Period (PVARP) is initialized. Atrial signals will be ignored during this period (Marker 2). A Ventricular Refractory Period (VRP) is also initialized to filter ventricular signals.

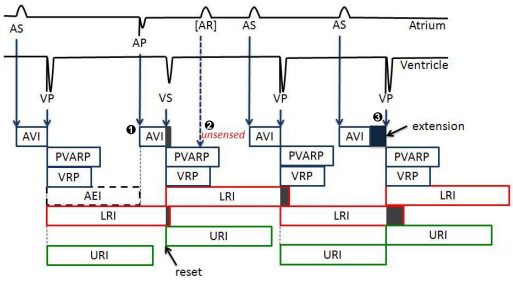


Fig. 10. Timing cycles of a dual-chamber DDD mode pacemaker [33].

B. Pacemaker model

The formal temporal logic based pacemaker model has been developed in both Simulink and UPPAAL [46]. A timer-based Simulink model was developed in [33].

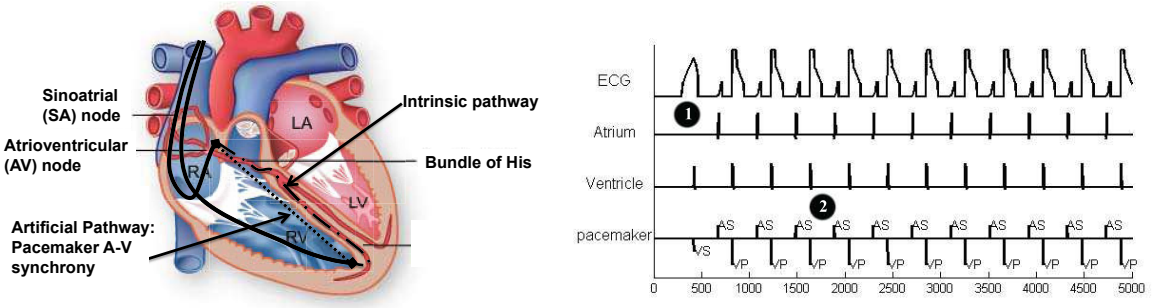
VII. Closed-loop Case Study

The function of the SA and AV nodes is controlled by the nervous system, which controls the heart rate and A-V conduction. A DDD pacemaker which is also able to change heart rate and A-V conduction may cause a "controller synthesis problem" where the pacemaker's pacing incorrectly drives the heart rate faster. In this section, we introduce a clinical case where a pacemaker drives the heart into a harmful condition. This behavior cannot be detected with open-loop testing of the device as the state of the heart changes in response to the pacemaker's premature stimulus.

A. Endless Loop Tachycardia (ELT)

In a healthy heart there is only one intrinsic conduction path from atria to the ventricles, which is from the SA node to the AV node and to the ventricles through the His bundle. The AVI timer of a DDD pacemaker introduces another virtual pathway between the atrial lead and the ventricular lead Fig. 11(a). The two pathways have the potential to form a conduction loop. Fig. 11(b) is a closed-loop simulation of the VHM and our pacemaker model which shows a clinical case where the heart rate is abnormally increased due to the conduction loop. This case is referred to as Endless Loop Tachycardia (ELT), which is one case in Pacemaker Mediated Tachycardia (PMT).

The ELT is induced by Premature Ventricular Contraction (PVC), which is due to abnormal self-depolarization of ventricular tissue. The PVC is sensed by the pacemaker, and triggers V-A conduction along the intrinsic pathway. Since the conduction pattern is different than intrinsic heart contraction, the PVC will initialize an abnormal QRS wave in the ECG channel (Marker 1). The V-A conduction will then trigger Atrial Sense (AS). The pacemaker will then pace the ventricle (VP) after TAVI according to its A-V synchrony function. The conduction loop is then formed and the VP-AS pattern will



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

(b) EGM for ELT initialized by a early ventricular signal

Fig. 11. Endless Loop Tachycardia case study demonstrating the situation when the pacemaker drives the heart into an unsafe state [36].

persist if no actions are taken. The heart rate is kept as high as the upper rate limit of the pacemaker since the cycle length of the conduction loop is very short. ELT is a harmful condition since the heart rate is a fast fixed heart rate that will cause inefficient pumping of the heart.

From pacemaker point of view, for every input AS, the pacemaker is working as specified. Thus, open-loop testing is unable to detect this closed-loop behavior. Modern pacemaker algorithms are able to identify potential ELT by measuring the time that the pacemaker is pacing at the upper rate limit. The ELT can be then terminated by skipping one ventricular pacing or by increasing the PVARP time to block the AS caused by the V-A conduction.

B. Other closed-loop case studies

There are other closed-loop case studies we have discussed in our previous publications. Normal pacemaker function for heart with bradycardia has been studied in [33]. Pacemaker with mode-switch function has been studied in [34]. Pacemaker malfunction due to functional aspects like Crosstalk and lead displacement has been studied in [35].

VIII. Discussion

In this section we describe the implementation of the VHM and pacemaker and discuss avenues of future work.

A. Physical Implementation

While a system model is useful for simulation-based testing and extraction to appropriate formal models for verification, the realization of the physical system is essential. Implementation issues such as the noise, jitter, timing overhead and signal distortion at the VHM-pacemaker interface can only be captured faithfully in a physical realization of the system. The VHM was implemented on a Xilinx Spartan-3, XC3S1000 FPGA [53], shown in Fig. 12. The pacemaker was implemented on a FireFly microcontroller-based platform [54]. FireFly runs nano-RK [55], a real-time operating system developed with timeliness as a first-class concern.

The pacemaker was implemented on a FireFly node using five tasks, corresponding to the automata from our Simulink model of the pacemaker, for the ventricle-pace (LRI_task), ventricle-sense (AVI_task), atrial-pace (ARP_task), atrial-sense (VRP_task) and a coordinator between the atrium and ventricle leads (URI_task). Each task was assigned a period of 10ms. The priorities for the tasks (along with equal offsets) are assigned to match the execution order of the parallel states in the timer-based pacemaker Simulink model. This implementation, while not fully reflective of the complexity of a modern pacemaker, is simple and allows the evaluator to easily disable some of the tasks to test pacemakers in any of the modes. In our initial setup of the implementation we have been able to experimentally validate the closed loop electrical interaction between the heart (FPGA) and pacemaker (FireFly node). This platform demonstrates real-time behavior of the pacemaker for normal sinus rhythm, sinus bradycardia (atrial pace only) and sinus bradycardia with heart block (synchronized dual-chamber pacing).

B. Conclusion and Future Work

As implantable medical devices grow in sophistication with significant capabilities implemented in software, firmware bugs account for an increasing fraction of device recalls. Since the FDA currently does not test, verify or certify the software in such devices, there is an urgent need for a systematic methodology to guarantee the safety of the device software. Furthermore, this safety must account for the closed-loop interaction with the organs of interest. In this effort, we have presented a first step in the direction of a holistic approach for model-based testing and physical patient–device interactive validation. We achieve this through the design of an integrated functional and formal model of the heart and pacemaker device using timed automata. Using this closed-loop system, we employ a monitor-based testing approach [36] that is both

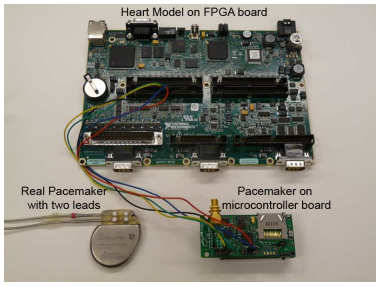


Fig. 12. Closed-loop experimental setup. A Boston Scientific pacemaker is shown for reference.

clinically-relevant and captures the complexities of interaction with physiological components.

This effort describes early steps toward cyber-physical model based testing, validation and verification of medical cyber-physical systems. This is a challenging domain as patient modeling is both complex, highly variable and non-deterministic and the safety properties must include over-approximated models for verification, abstract models for simulation and be realizable in physical form for testing. We envision several avenues for future work and discuss a few:

1. Test Generation: For the setup presented in Fig. 1 it is necessary to create a relevant sequences of mode selection signals and model parameters which would guarantee appropriate testing of the device. The procedure must take into account the device model along with desired test-coverage approach.

2. Verification: For a specific VHM configuration it is possible to automatically extract a formal, timed-automata description of the VHM compatible with the UPPAAL verification tool [44]. This would allow a closed-loop system verification using UPPAAL's built-in verification procedures.

3. Combining the VHM with more complex devices: such as ICDs which are rate-adaptive and operate with varying levels of hysteresis. We plan to investigate a translation of the VHM to Linear Hybrid Systems in order to use a *symbolic analysis framework* proposed in [56].

ACKNOWLEDGMENT

We wish to thank Insup Lee and Rajeev Alur for their valuable insights. Preliminary versions of this paper have appeared in ECRTS 2010 [33], EMBC 2010 [34], ICCPS 2011 [36] and EMBC 2011 [35]; and demonstrated at CPSWeek 2010 and 2011.

REFERENCES

[1] W. H. Maisel, M. O. Sweeney, W. G. Stevenson, K. Ellison, and L. M. Epstein, "Recalls and safety alerts involving pacemakers and implantable cardioverter-defibrillator generators," *J. American Med. Ass.*, vol. 286, pp. 793–799, 2001.

[2] "Personal communication with Paul L. Jones, Senior Systems/Software Engineer, Office of Science and Engineering Laboratories, Center for Devices and Radiological Health, US FDA. August, 2010."

[3] "List of Device Recalls, U.S. Food and Drug Admin., (last visited Jul. 19, 2010)."

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

[5] D. A. Vogel, *Medical Device Software Verification, Validation, and Compliance*. Artech House, 2010.

[6] "AUTOSAR website: www.autosar.org/."

[7] "AVSI website: http://www.avsi.aero."

[8] R. Jetley, S. P. Iyer, and P. L. Jones, "A Formal Methods Approach to Medical Device Review," *IEEE Computer*, vol. 39, pp. 61–67, 2006.

[9] I. Lee, G. J. Pappas, R. Cleaveland, J. Hatcliff, B. H. Krogh, P. Lee, H. Rubin, and L. Sha, "High-Confidence Medical Device Software and Systems," *IEEE Computer*, vol. 39, no. 4, pp. 139–148, 2006.

[10] N. G. Leveson and C. S. Turner, "Investigation of the Therac-25 Accidents," *IEEE Computer*, pp. 18–41, 1993.

[11] "US FDA, General Principles of Software Validation; Final Guidance for Industry and FDA Staff," 2002.

[12] "US FDA, Design Control Guidance For Medical Device Manufacturers," 1997.

[13] "US FDA, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices," 2005.

[14] "PACEMAKER System Specification. Boston Scientific. 2007."

[15] A. O. Gomes and M. V. Oliveira, "Formal Specification of a Cardiac Pacing System," in *Proceedings of the 2nd World Congress on Formal Methods*, ser. FM '09. Springer-Verlag, 2009, pp. 692–707.

[16] E. Jee, I. Lee, and O. Sokolsky, "Assurance Cases in Model-Driven Development of the Pacemaker Software," in *Leveraging Applications of Formal Methods, Verification, and Validation*, ser. LNCS, 2010, vol. 6416, pp. 343–356.

[17] J. M. Cortner, "Testing Implantable Medical Devices," *Global Healthcare Medical Device Manufacturing Technology*, pp. 2–4, 2003.

[18] *Medtronic ViP-II Virtual Interactive Patient: User's Manual Software v1.5*. Rivertek Medical Systems, 2006.

[19] J. Rushby, "Verified software: Theories, tools, experiments." Springer-Verlag, 2008, ch. Automated Test Generation and Verified Software, pp. 161–172.

[20] E. M. Clarke and J. M. Wing, "Formal Methods: State of the Art and Future Directions," *ACM Computing Surveys*, vol. 28, pp. 626–643, 1996.

[21] D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky, "Formal Methods Based Development of a PCA Infusion Pump Reference Model: Generic Infusion Pump (GIP) Project," in *High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP. Joint Workshop on*, 2007, pp. 23 –33.

[22] R. Alur, D. Arney, E. L. Gunter, I. Lee, J. Lee, W. Nam, F. Pearce, S. Van Albert, and J. Zhou, "Formal Specifications and Analysis of the Computer-Assisted Resuscitation Algorithm (CARA) Infusion Pump Control System," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 5, pp. 308–319, 2004.

[23] A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang, "Improving medical protocols by formal methods," *Artificial Intelligence in Medicine*, vol. 36, no. 3, pp. 193 – 209, 2006.

[24] D. Arney, M. Pajic, J. M. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, "Toward patient safety in closed-loop medical device systems," in *ACM/IEEE International Conference on Cyber-Physical Systems*, 2010, pp. 33–38.

[25] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. M. Goldman, and I. Lee, "Model-driven safety analysis of closed-loop medical systems," UPenn Technical Report, 2011.

[26] 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*, vol. 46, pp. 26–28, 2001.

[27] C. C. Mitchell and D. G. Schaeffer, "A Two-current Model for the Dynamics of Cardiac Membrane," *Bulletin of Mathematical Biology*, vol. 65, no. 5, pp. 767 – 793, 2003.

[28] D. Hooks, K. Tomlinson, S. Marsden, I. LeGrice, B. Smaill, A. Pullan, and P. Hunter, "Cardiac Microstructure: Implications for Electrical Propagation and Defibrillation in the heart," *Circulation Research*, vol. 91, pp. 331–338, 2002.

[29] E. Hsu and C. Henriquez, "Myocardial Fiber Orientation Mapping Using Reduced Encoding Diffusion Tensor Imaging," *Journal of Cardiovascular Magnetic Resonance*, vol. 3, no. 4, pp. 339–347, 2001.

[30] P. Hunter, A. Pullan, and B. Smaill, "Modeling Total Heart Function," *Annual Review of Biomedical Engineering*, vol. 5, no. 1, pp. 147–177, 2003.

[31] R. R. Aliev and A. V. Panfilov, "A Simple Two-variable Model of Cardiac Excitation," *Chaos, Solitons & Fractals*, vol. 7, no. 3, pp. 293 – 301, 1996.

[32] M. Malik, T. Cochrane, D. W. Davies, and A. J. Camm, "Clinically Relevant Computer Model of Cardiac Rhythm and Pacemaker/Heart Interaction," *Medical & Biological Engineering & Computing*, vol. 25, 1987.
[33] Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam, "Real-Time Heart Model for Implantable Cardiac Device Validation and Verification," in *22nd Euromicro Conference on Real-Time Systems (ECRTS)*, July 2010, pp. 239 –248.
[34] 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*, pp. 6690 –6693, 2010.
[35] Z. Jiang and R. Mangharam, "Modeling Cardiac Pacemaker Malfunctions with the Virtual Heart Model," *33nd Intl. Conf. IEEE Engineering in Medicine and Biology Society*, 2011.
[36] Z. Jiang, M. Pajic, and R. Mangharam, "Model-based Closed-loop Testing of Implantable Pacemakers," in *ICCPS'11: ACM/IEEE 2nd Intl. Conf. on Cyber-Physical Systems*, 2011.
[37] Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam, "Virtual Heart Model, Technical Report ESE-2010-01," 2010.
[38] O. Berenfeld and S. Abboud, "Simulation of Cardiac Activity and the ECG using a Heart Model with a Reaction-Diffusion Action Potential," *Med. Eng. Phys. Vol. 18, pp. 615-625*, 1996.
[39] J. Beaumont, D. C. Michaels, M. Delmar, J. Davidenko, and J. Jalife, "A Model Study of Changes in Excitability of Ventricular Muscle Cells," *Am. J. Physiol. 268 (Heart Circ Physiol. 37)*, 1995.
[40] D. Harrild and C. Henriquez, "A Computer Model of Normal Conduction in the Human Atria," *Circ. Res.*, vol. 87, pp. e25–36, 2000.
[41] D. R. Adam, "Propagation of Depolarization and Repolarization Processes in the Myocardium - An Anisotropic Model," *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 2, pp. 133–141, 1991.
[42] A. J. Pullan, L. K. Cheng, and M. L. Buist, *Mathematically Modeling the Electrical Activity of the Heart*. World Scientific, 2005.
[43] "Matlab R2011a Documentation → Stateflow," http://www.mathworks.com/help/toolbox/stateflow.
[44] K. Larsen, P. Pettersson, and W. Yi, "Uppaal in a Nutshell," *International Journal on Software Tools for Technology Transfer (STTT)*, pp. 134–152, 1997.
[45] G. Behrmann, A. David, and K. G. Larsen, "A Tutorial on Uppaal," *Formal Methods for the Design of Real-Time Systems, Lecture Notes in Computer Science*, pp. 200–236, 2004.
[46] Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam, "DDD Pacemaker Verification Report, University of Pennsylvania," 2011.
[47] M. Josephson, *Clinical Cardiac Electrophysiology*. Lippincot Williams and Wilkins, 2008.
[48] R. Alur and D. L. Dill, "A Theory of Timed Automata," *Theoretical Computer Science*, vol. 126, pp. 183–235, 1994.
[49] S. Yovine, "Kronos: A Verification Tool for Real-time Systems." *Journal on Software Tools for Technology Tansfer*, 1997.
[50] P. Denes, D. Wu, R. Dhingra, R. J. Pietras, and K. M. Rosen, "The Effects of Cycle Length on Cardiac Refractory Periods in Man," *Circulation*, vol. 49, pp. 32–41, 1974.
[51] M. R. Franz, C. D. Swerdlow, L. B. Liem, and J. Schaefer, "Cycle Length Dependence of Human Action Potential Duration In vivo," *The Journal of Clinical Investigation*, vol. 82, no. 3, pp. 972–979, 1988.
[52] W. Stevenson and K. Soejima, "Recording Techniques for Clinical Electrophysiology," *Journal of Cardiovascular Electrophysiology*, vol. 16, pp. 1017–1022, 2005.
[53] Xilinx, "Spartan-3 FPGA Datasheet."
[54] R. Mangharam, A. Rowe, and R. Rajkumar, "FireFly: A Cross-layer Platform for Real-time Embedded Wireless Networks," *Real-Time System Journal*, vol. 37, no. 3, pp. 183–231, 2007.
[55] "nano-RK Sensor RTOS. http://nanork.org."
[56] R. Alur, A. Kanade, S. Ramesh, and K. C. Shashidhar, "Symbolic Analysis for Improving Simulation Coverage of Simulink/Stateflow Models," in *Proceedings of the 8th ACM international conference on Embedded software*, ser. EMSOFT '08, 2008, pp. 89–98.

Zhihao Jiang (S'10) received his Bachelor's degree in Engineering from University of Electronic Science and Technology of China in 2004 and M.S. in Robotics in University of Pennsylvania in 2010. Since 2010 he has been a Ph.D. Candidate in the Computer & Information Science Department in University of Pennsylvania. His research interests are in Cyber-Physical System with a focus on medical device modeling, validation and verification.

Miroslav Pajic (S'06) received the Engineer Diploma (graduated with *summa cum laude*) from the School of Electrical Engineering, University of Belgrade in 2003 and M.S. degrees in Electrical Engineering from University of Belgrade in 2007 and the University of Pennsylvania in 2010. Since 2008 he has been a Ph.D. candidate in the Department of Electrical & Systems Engineering at the University of Pennsylvania. His research interests include real-time embedded systems and cyber-physical systems, with a focus on networked control systems and medical devices.

Rahul Mangharam (M'02) received his BS, MS and Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University in 2000, 2002 and 2008 respectively. He is the Stephen J Angello Chair and Assistant Professor in the Department of Electrical & Systems Engineering and Department of Computer & Information Science at the University of Pennsylvania. His interests are in real-time scheduling for networked embedded systems with applications in automotive systems, medical devices, energy systems and control networks.