

Real-time Heart Model for Implantable Cardiac Device Validation and Verification

Zhihao Jiang* Miroslav Pajic* Allison Connolly† Sanjay Dixit‡ Rahul Mangharam*
*Dept. Electrical & System Engineering †Dept. of Biomedical Engineering ‡Cardiology Division
University of Pennsylvania Johns Hopkins University Hospital of the Univ. of Penn.
{zhihaoj, pajic, rahulm}@seas.upenn.edu aconnol2@jhu.edu sanjay.dixit@uphs.upenn.edu

Abstract—Designing bug-free medical device software is difficult, especially in complex implantable devices that may be used in unanticipated contexts. Safety recalls of pacemakers and implantable cardioverter defibrillators due to firmware problems between 1990 and 2000 affected over 200,000 devices, comprising 41% of the devices recalled and are increasing in frequency [1]. There is currently no formal methodology or open experimental platform to validate and verify the correct operation of medical device software. To this effect, a real-time Virtual Heart Model (VHM) has been developed to model the electrophysiological operation of the functioning (i.e. during normal sinus rhythm) and malfunctioning (i.e. during arrhythmia) heart. We present a methodology to extract timing properties of the heart to construct a timed-automata model. The platform exposes functional and formal interfaces for validation and verification of implantable cardiac devices. We demonstrate the VHM is capable of generating clinically-relevant response to intrinsic (i.e. premature stimuli) and external (i.e. artificial pacemaker) signals for a variety of common arrhythmias. By connecting the VHM with a pacemaker model, we are able to pace and synchronize the heart during the onset of irregular heart rhythms. The VHM has also been implemented on a hardware platform for closed-loop experimentation with existing and virtual medical devices. The VHM allows for exploratory electrophysiology studies for physicians to evaluate their diagnosis and determine the appropriate device therapy. This integrated functional and formal device design approach will potentially help expedite medical device certification for safer operation.

I. INTRODUCTION

The heart is one of the most fundamental and important natural real-time systems. The heart spontaneously generates electrical impulses which organize the sequence of muscle contractions during each heart beat and are essential for optimizing the cardiac stroke volume. The pattern and the timing of these impulses determine the heart rhythm. Derangements in this rhythm impair the heart's ability to pump enough blood to meet the body's demand. Thus, the heart's electrical system and its timing, also known as its electrophysiological operation, are fundamental to the cardiac function.

The use of artificial implantable heart rhythm devices such as pacemakers and cardioverter-defibrillators has grown rapidly over the recent decades and have demonstrated over 99% efficacy for patients with cardiac arrhythmias or abnormal heart rhythm. In the 20-year period from 1985 to 2005, the US Food and Drug Administration's (FDA) Maude database records almost 30,000 deaths and almost 600,000 injuries from device failures [2]. There is currently no formal methodology or open experimental platform to validate and verify the correct operation of medical device software. The FDA has expressed the need for rigorous real-time methodologies to validate and verify medical device software as is currently done in the

domains of avionics and industrial control automation [3].

Medical devices are inherently Cyber-Physical Systems where the control and computation within the device is tightly coupled with the sensing and actuation of the biological physical substrate (i.e. the heart). It is therefore essential to model the functioning of the device within the physical environment. The relation between the physical state and the device state is largely non-deterministic, interactive and cannot be fully captured by computation models. The modeling of the physical substrate must therefore be restricted to specific cases and conditions of operation. Thus, the validation and verification observations are only valid for those specific cases.

To address this need, a Virtual Heart Model (VHM) has been developed to emulate the heart's electrophysiological operation for specific common arrhythmias. The VHM exposes functional and formal interfaces for validation and verification of implantable cardiac devices, as shown in Fig. 1. In this investigation, we present a methodology to extract timing properties of the heart to construct a timed-automata model. The functional model is then validated by comparing the behavior of the VHM to three common cases of normal and abnormal heart rhythm. These cases are observed in real patients due to failure of impulse generation and failure of impulse propagation. The clinical relevance of the electrogram outputs from the model have been validated for these specific cases by an electrophysiologist.

Now that the VHM has been validated, we are in the position to validate and verify medical devices in closed-loop operation with the VHM. We designed and validated the functional pacemaker model for the two most frequent arrhythmias. The formal model of the pacemaker was designed and verified within the context of the VHM using Simulink Design Verifier [4]. Following this, an initial version of the VHM was implemented on a FPGA-based hardware platform and the pacemaker was implemented on a microcontroller-based platform for closed-loop experimental evaluation.

The primary contribution of this effort is the development of an integrated functional and formal device design approach which has the potential to help expedite medical device certification for safer operation. In addition, the VHM allows for exploratory electrophysiology studies for physicians to evaluate their diagnosis and determine the appropriate device therapy.

A. Functional and Formal Modeling of the Heart

The functional and formal interfaces of the VHM are derived from the common kernel. The kernel models the cardiac action

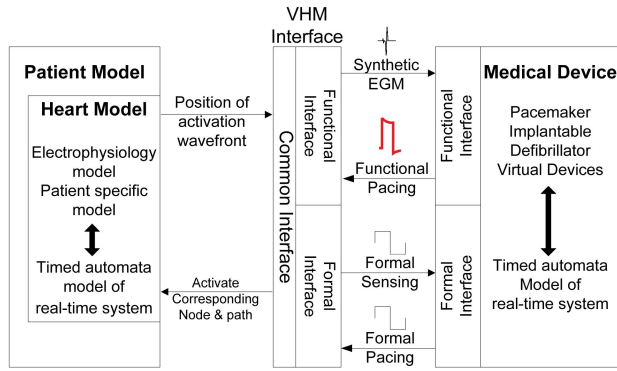


Figure 1. Structure of the VHM platform

potential which is a principal phenomena of the heart's conduction system. This allows us to model the heart as a network of nodes, which are abstractions of localized electrically active tissue. Conduction between nodes is modeled by paths with known propagation and timing behavior. The functional model emulates the behavior of the heart and allows validation through simulation and black-box testing of the implantable devices.

We designed the kernel using the timed automata approach[5] as the timing of the heart's electrical system is fundamental to the cardiac function[6]. While several cellular-based heart models exist [7], [8], [9], [10] they are useful for simulation only and not for software verification. By simplifying the continuous system to an event-based timed automata we are able to both simulate and verify the closed-loop system for specific cases of interest.

B. Organization of the paper

The rest of the paper is organized as follows: In Section II, we describe the VHM and the method in which the timing parameters are extracted. Section III provides medical validation of the VHM. In Section IV, we introduce the pacemaker model and its closed-loop evaluation. Finally, Section V describes an FPGA implementation of the VHM along with the closed-loop hardware setup used for pacemaker validation.

II. VHM PLATFORM

A. The Cardiac Electrical System

In order to better understand the operation of the heart, we provide some background. The human heart maintains blood circulation of the body by coordinated contraction of the atria and ventricles. Fig. 2(a) shows the essential elements of the electrical conduction system of the heart. The electrical signal originates from specialized tissue in the sinoatrial (SA) node, which serves as the primary pacemaker of the heart. The SA node spontaneously produces an electrical signal, which is conducted radially through both atria, causing them to contract. The signal then passes through the slow conducting AV node, allowing blood to empty out of the atria and fill the ventricles. The fast-conducting His-Purkinje system spreads the electricity through the ventricles, causing all of the tissue in both ventricles to contract simultaneously and force blood out of the heart [6]. This electrical system provides organized contraction of the heart muscle and optimizes hemodynamics. Abnormalities in the electrical signal generation and propagation can cause

different types of arrhythmias like *Tachyarrhythmias* (abnormally fast heart rate) and *Bradyarrhythmias* (abnormally slow heart rate), which require medical intervention in the form of medication, surgery, or an implantable device.

The electrical signal that passes through the heart is known as an action potential. A typical ventricular action potential is shown in Fig. 3(a). The action potential is triggered by a voltage spike from the action potential of its neighboring tissue or from an artificial pacing signal. The upstroke indicates the depolarization of the cell and the time when the muscle contracts. This is followed by the plateau, which allows the muscle to hold its contraction and fully eject blood. The downstroke is repolarization, when the muscle relaxes and refills.

The total refractory period is the amount of time it takes for an excited cell to be ready for a second stimulus once it returns to its resting state. This can be divided into two time periods, the effective refractory period (ERP) and relative refractory period (RRP). The cell cannot be activated by an electrical stimulus in the ERP, which acts as a blocking interval. In the RRP, the cell can be activated again, but this causes changes in action potential morphology. Changes in the duration of the plateau will change the duration of the next ERP. A more gradual upstroke of depolarization will slow the conduction velocity of the tissue, as it will take more time to reach the voltage threshold necessary to trigger a neighbor to depolarize. Thus, the shape and timing of the action potential determines the conduction velocity, and refractoriness of the heart.

The electrical activity of the heart can be monitored externally by an electrocardiogram (ECG) or internally, using electrograms (EGM). The ECG measures the voltage difference between two leads placed on the skin of the torso and provides a general view of the electrical activities of the heart. The electrogram signal is recorded from electrodes embedded in a catheter placed on the inside surface of the heart. It is a representation of local electrical activities of the heart. The EGMs are useful in diagnosing cardiac arrhythmias for their capability of localizing the source of activation[11]. Similar electrodes are embedded in the leads of a pacemaker or implantable cardioverter-defibrillator (ICD).

B. Timing Model of the VHM

Modeling individual cells in order to obtain a global view of the heart is processor heavy and contains extraneous information for purposes of device testing. Instead, our model utilizes

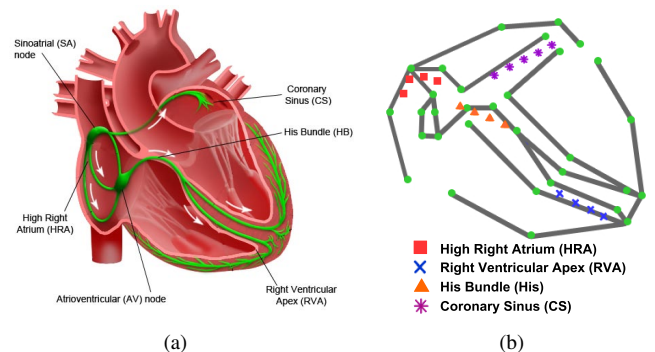


Figure 2. (a) The basic physiology and electrical conduction system of the heart. (b) Corresponding setup of nodes (dots), paths (lines) and probes (shapes) in our heart model.

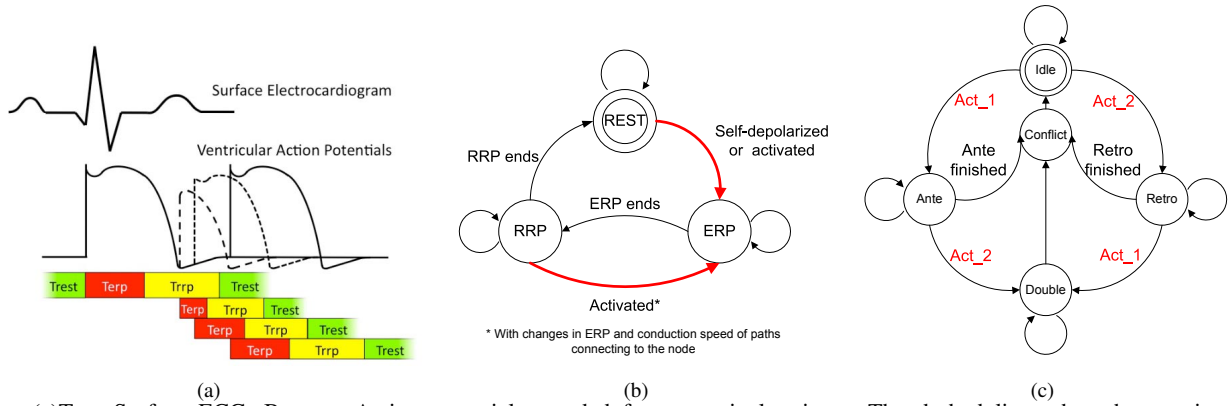


Figure 3. (a)Top: Surface ECG, Bottom: Action potential recorded from ventricular tissue. The dashed lines show how action potential morphology changes when an early stimulus is applied to the tissue and how their corresponding timer values change in our model.(b)Node automaton. (c)Path automaton

the timing properties of the heart to obtain a macro-level view by lumping cells into node automaton and path automaton.

The heart can be represented as a conduction network (Fig. 2(b)), because activation of heart tissue can only trigger its neighboring tissue. On a conduction path from tissue A to tissue B, a stimulus cannot reach B if A is in ERP, even if B is in the rest state. This is referred to as the functional refractory period (*FRP*) of B. We can use this idea to model a section of tissue as two node automata connected by one path automaton. The refractory properties of the component are represented by the nodes and the conduction properties are modeled by the path. We can represent different structures of the heart by varying the parameters of the nodes and paths.

The basic state transitions of the node automaton and the path automaton are shown in Fig. 3(b) and Fig. 3(c). In node automaton, the refractoriness is modeled as *ERP*, *RRP* and *Rest* states, and their durations are modeled by the timers T_{erp} , T_{rrp} , and T_{rest} . The default refractory parameters we are using are (in msec): ERP atrium:150-300; ERP AV node:230-400; ERP His-Purkinje:300-400; ERP Ventricle:180-290. These parameters are tuned in relation to the true refractory periods measured in clinical EP studies[12]. This allows us to produce clinically-relevant results. In path automaton, the conduction properties are modeled as no conduction (*Idle*), antegrade conduction or forward conduction (*Ante*), retrograde conduction or backward conduction (*Retro*), both direction conduction (*Double*) and conflict (*Conflict*) state, and the conduction delays are modeled by the timers T_{ante} and T_{retro} .

It has been studied in [13] that action potential duration is nearly logarithmically dependent on the timing of stimuli. We made a first order approximation of the trend to create similar behavior. In addition, the portion of total refractory period that is in ERP is dependent on the amplitude of the stimuli. In our model, we assumed a fixed amplitude for all the stimuli for simplification, which is true in most implantable devices.

For the node automata in our model, the default value of its T_{erp} timer is determined by the earliness of a stimulus when it is applied to the node. The earliness, C_{early} , is a factor between 0-1, which is a measurement of how early the stimulus is applied to the node before its Rest state (Eq. 1). For each node, the default value of T_{erp} has a range $[T_{erp}^{min}, T_{erp}^{max}]$. The equation for ERP default timer value T_{erp} of node automata is shown in Eq. 3. For nodes in atria, His-Purkinje system and the ventricle,

when the earliness increases, the default value of T_{erp} decreases slowly at first and abruptly when the earliness is around 1. The trend is similar to the result in[13]. The AV node has a opposite trend where T_{erp} increases when the earliness increases, which matches the result in[12]. The earliness of the stimulus also determines the conduction velocity of its neighboring paths.

The conduction delays, which are the default values of T_{ante} and T_{retro} timers in our model, range from the ratio of path length/conduction velocity to 4 times of this value. The equations for forward conduction delay and backward conduction delay are shown in Eq. 5 and Eq. 6 respectively. For all heart tissues except the AV node, the conduction delay increases slowly at first as the earliness increases and changes abruptly when the earliness approaches 1. For the AV node the relationship is linear, which increase faster than other tissue. The equations that we used for adaptive ERP and conduction velocities are shown below:

$$C_{early} = \begin{cases} 0, s = ERP \\ C_{rrp}, s = RRP \\ 1, s = Rest \end{cases} \quad (1)$$

where C_{rrp} is the ratio between current RRP timer value of the i^{th} node and its default value when the node is activated

$$C_{erp}^i = \begin{cases} (1 - C_{early})^3, i = AVnode \\ C_{early}^3, i \neq AVnode \end{cases} \quad (2)$$

$$T_{erp}^i = T_{erp}^{min} + [(1 - C_{erp}^i) \cdot (T_{erp}^{max} - T_{erp}^{min})] \quad (3)$$

where T_{erp}^{min} and T_{erp}^{max} are minimal and maximal value of the default value of ERP timer.

$$C_{cond} = \begin{cases} 1 + 3C_{early}, i = AV \\ 1 + 3C_{early}^2, i \neq AV \end{cases} \quad (4)$$

$$T_{ante}^{i \rightarrow j} = \lfloor ante_del \cdot C_{cond}^i \rfloor \quad (5)$$

where $ante_del = path_length/V_{ante}$.

$$T_{retro}^{i \rightarrow j} = \lfloor retro_del \cdot C_{cond}^j \rfloor \quad (6)$$

where $retro_del = path_length/V_{retro}$.

The state transitions of node automaton and path automaton are shown below:

1. Node automata state transitions

- Received activation signal at node

- In Rest state: go to ERP state, calculate the value of T_{erp} (Eq. 3), reset the ERP timer, activate neighboring paths and change the T_{ante} or T_{retro} (Eq. 5- 6).

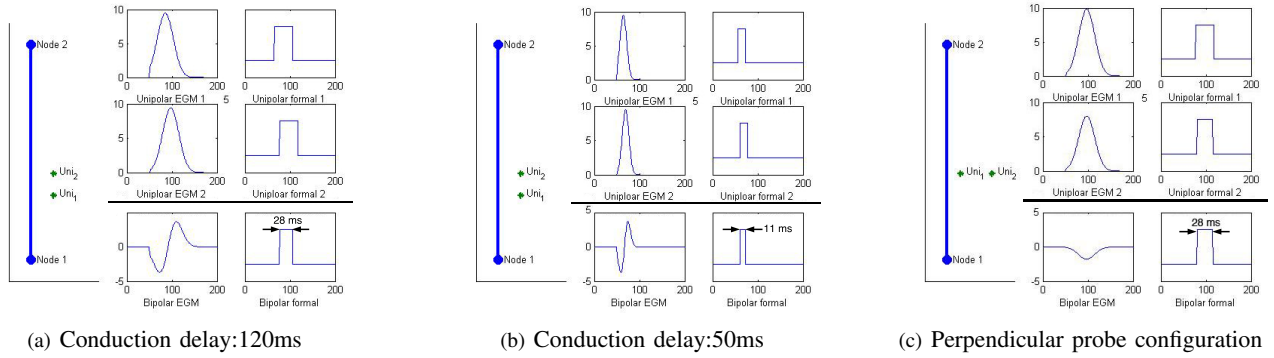


Figure 4. For (a),(b) and (c), the left columns show the placement of probes in relation to the path; the middle columns show the functional EGM, and right columns show the formal signal sensed by the probe.

- In ERP state: calculate T_{erp} (Eq. 3), neighboring paths are not activated.
 - In RRP state: go to ERP state, activate neighboring paths, calculate T_{erp} (Eq. 3), and change the conduction velocity of neighboring paths (Eq. 5- 6).
- No activation signal received at node
- In Rest state: SA and AV nodes count down the Rest timer, when the timer times out, activate itself, go to ERP state, activate neighboring paths and change the conduction speed of neighboring paths according to Eq. 5- 6. Other nodes stay in Rest state.
 - In ERP state: count down ERP timer and go to RRP state after the timer runs out.
 - In RRP state: count down RRP timer and go to Rest state after the timer runs out.

2. Path automata state transitions: The path automata is initially in an idle state until either of the nodes it connects to is activated(Act_1 or Act_2). The path starts the antegrade or retrograde conduction timer according to which node is activated. After the antegrade or retrograde timer times out, the path activates the node at the opposite end. Because this node activates all paths it is connected to, the path where the activation originated must go to Conflict state to prevent back flow. If a path is already in antegrade or retrograde conduction when a second activation signal enters from the opposite end, the path enters the Double state. Both Ante and Retro timer count down until the timers correspond to the same location and the path goes to the Conflict state.

C. Functional and Formal VHM interface

Our platform provides two interfaces, a formal signal for medical device software and a functional electrogram for real device implementation (see Fig. 1).

We introduce probes into the model which are equivalent to the electrodes on the catheters or pacemaker leads. These unipolar probes generate synthetic electrograms by multiplying all voltages by a Gaussian factor and summing them together. The bipolar electrograms are the differences between electrode pairs. The bipolar formal signal is generated by an AND operation between the two unipolar formal signals. Fig. 4(a), Fig. 4(b) and Fig. 4(c) demonstrate an activation signal traveling from node 1 to node 2. Changes in electrogram morphology and duration of corresponding formal signals can be seen with different configurations of the probe pair and the different con-

duction delay of the path. The formal signal is used to interact with medical device software running at a lower frequency.

D. Simulink Implementation of the VHM

The general automata and probe set used in our simulations is shown if Fig. 2(b). The probes are placed in specific areas to capture key activation timing intervals in the heart. The His-bundle electrogram (HBE) can record impulses from the atrium, His-bundle, and ventricles, which is useful for measuring the atrium-to-His conduction time (A-H interval). Fig. 5(a) shows the simulation GUI we developed in Matlab, where the heart anatomy is superimposed on the automata network. The length of paths are measured in pixels to provide a relative length relationship between different heart structures. Users can track the updated values of timers on the right side tables. The current state of all the nodes and paths are visualized using different colors, giving a more intuitive understanding of the simulation. Users can view electrograms and deliver programmed pacing in real-time. Fig. 5(b) shows the electrograms measured from all probes placed in the model. The underlying node automata and path automata are implemented in Simulink, as shown in Fig. 6 and Fig. 7. We chose Simulink as both the functional and formal models could be developed with a common kernel.

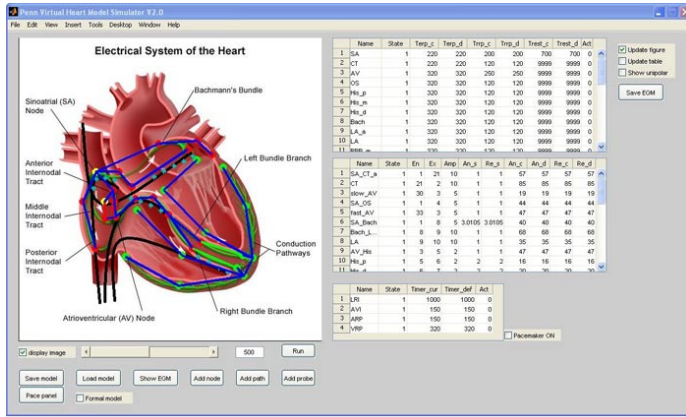
III. MEDICAL VALIDATION OF VHM

In order to validate the function of the heart model, we compare the behavior of the VHM to common arrhythmias seen in real patients due to failure of impulse generation and propagation. The clinical relevance of the electrogram outputs from the model have been validated by an electrophysiologist. In the following cases, we illustrate how our model can reproduce the underlying electrical activity of the cardiac phenomenon. While these cases may seem rather detailed to the real-time community they capture the fundamental behavior of the node and path automatas.

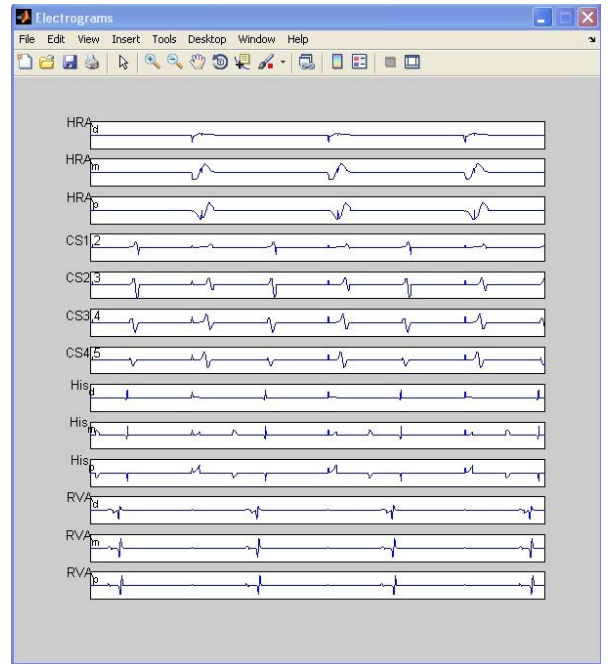
A. Case 1: Wenckebach-type A-V nodal response

This case illustrates the Wenckebach block. The conduction through the AV node slows, causing the ERP to lengthen, resulting in a dropped beat due to failure of propagation. This case highlights the capability of VHM to mimic the behavior of the AV node.

Electrophysiologists use atrial pacing to induce a Wenckebach block [14] where the signal fails to conduct from atria to ventricles. During this study, the heart is paced at constant cycle



(a) Simulation environment along with the Pace panel



(b) Synthetic electrogram

Figure 5. GUI environment

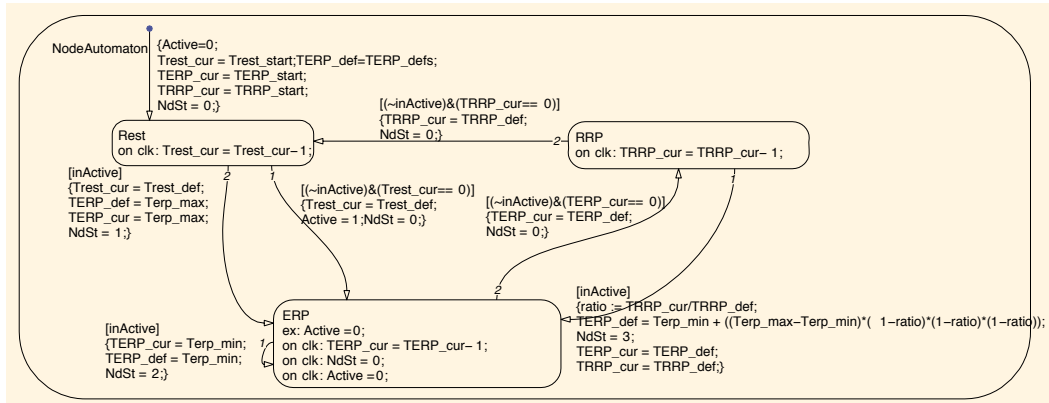


Figure 6. Simulink design of node automata

length. As pacing cycle length shortens, the AV conduction delay increases after each pacing signal until there is a dropped beat in the ventricles (as shown in Fig. 8). This phenomena occurs because the ERP lengthens if a stimulus arrives during the RRP interval of the AV node. When the ERP becomes longer than the cycle length, conduction is blocked (as shown in Fig. 9). This is known as a dropped beat, after which the A-H conduction time recovers and the cycle repeats. The behavior of the ERP is captured in the VHM's AV node automata.

B. Case 2: Conduction Response to Atrial Extrastimuli

This case demonstrates conduction responses to atrial extrastimuli (premature stimuli), which is the most basic electrophysiological test to locate abnormalities in the heart's conduction system. This case highlights the capability of our model to simulate different responses to extrastimuli in the atrium.

Atrial extrastimuli testing is used to characterize different sites of conduction delay and block in the heart. After a drive train is used at constant basic cycle length (BCL), an extra stimulus is introduced with shorter cycle length to test conduction

velocity. The time between the BCL and the extrastimulus is the coupling interval $A_1 - A_2$. For long coupling intervals, there is no apparent delay. When the coupling interval is short, it falls on the RRP of the delayed node and causes delays downstream. The most common delays are seen in the AV node. The total refractory period of the AV node is longer than that of any other node group, which causes an increase in conduction time of the extrastimulus from the atrium to the His-bundle ($A_2 - H_2$)

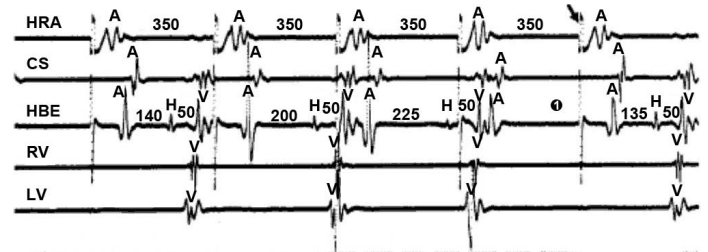


Figure 8. Electrograms of induced Wenckebach block in a patient [12]. The atria are stimulated with short cycle length of 350 msec. With each stimuli, the A-H interval in the HBE increases until there are no H and V at (1) due to the block in A-V node. The A-H interval then recovers and the cycle repeats.

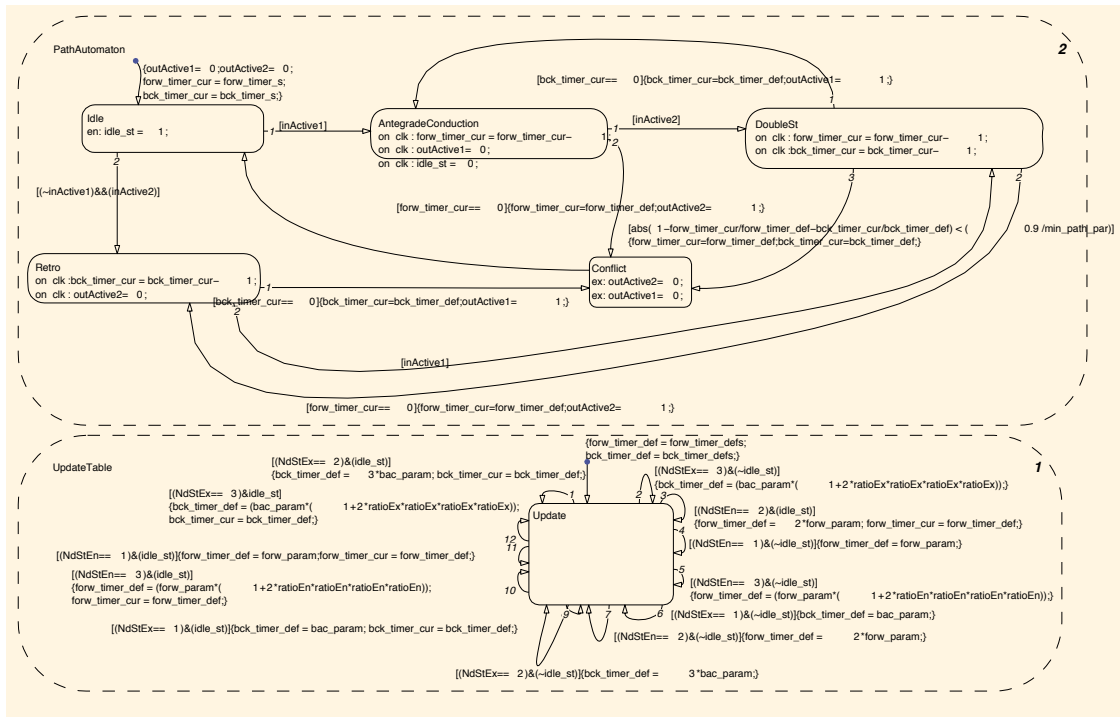


Figure 7. Simulink design of path automata

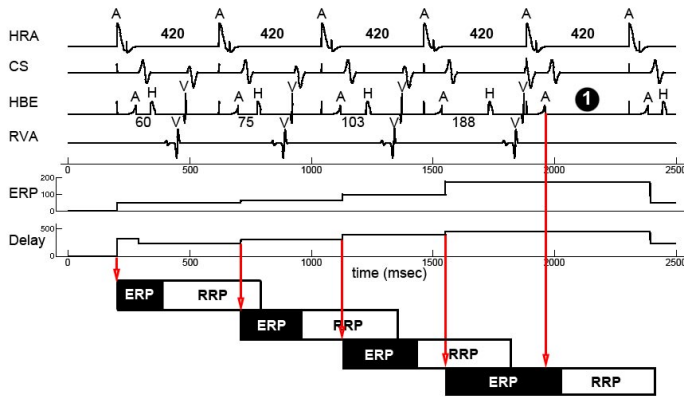


Figure 9. Electrograms of induced Wenckebach block in the heart model with stimulus cycle length of 420 msec. The heart model also displays lengthening in the A-H interval and block in A-V node. Rows 5 and 6 show the increase in the ERP and conduction delay of A-V node.

Fig. 10. The His-bundle and ventricles are downstream from the delay in the AV node, so the time between two His activations ($H_1 - H_2$) and two ventricular activations ($V_1 - V_2$) become longer. There are no delays between the His-bundle and the ventricles, so the conduction time between these two nodes ($H_2 - V_2$) does not change. By setting the refractory period parameters so that the AV node has the longest total refractory period, as shown in Table I, the heart model can generate similar trends, as depicted in Fig. 11. When the extrastimulus occurs during T_{rest} of the AV node, there is no delay. However, when the extrastimulus occurs early during the T_{rrp} of the AV node, the $A_2 - H_2$ interval lengthens dramatically.

C. Case 3: AV Nodal Reentry Tachycardia

Tachyarrhythmias are the cause of significant mortality. The most common mechanism for these arrhythmias is a reentry circuit. Case 3 uses a supraventricular tachycardia (SVT) known as AV nodal reentry tachycardia (AVNRT) to demonstrate the

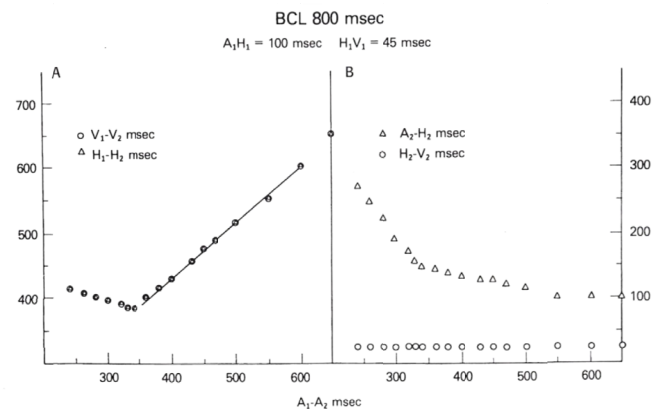


Figure 10. Response to atrial extrastimuli in a patient [12]. Drive train with period 800 msec is used before the extrastimuli are introduced. Conduction occurs without delay for longer coupling intervals. Significant AV nodal block causes the increase in A_2-H_2 interval due to long refractory periods in the AV node. The in V_1-V_2 and H_1-H_2 intervals increase because these nodes are downstream from the AV node.

capability of our model. It captures the conduction properties of paths and refractoriness of nodes. AVNRT can occur in patients of all ages due to premature depolarization of the atria, and it can be induced by rapid atrial pacing. In these patients there is an extra path with slow conduction and short refractory period in the AV node in addition to the normal fast path with long refractory period Fig. 12. Normally, conduction cannot occur through this slow path because signals from the fast and slow

Table I
DEFAULT TIMER VALUES FOR ATRIUM STIMULI RESPONSE SIMULATION

Node Group	ERP range(ms)	RRP(ms)
Atrium	150-200	100
AV node	230-300	200
His-Purkinje	300-350	50
Ventricle	200-290	100

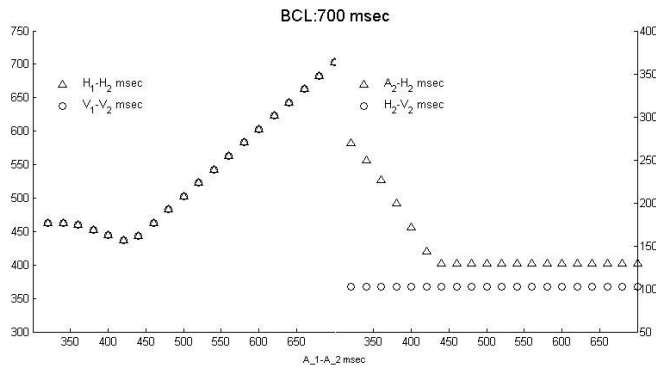


Figure 11. Response in the heart model simulation, with drive train period 700 msec.

paths conflict. However, when the atria are activated early by an extrastimuli, as in Fig. 12, AVNRT can be induced. The simulation results from our heart model are shown in Fig. 13.

IV. CLOSED LOOP MEDICAL DEVICE VALIDATION AND VERIFICATION

Now that the VHM has been validated, we are in the position to validate and verify medical devices in closed-loop operation with the VHM. The general framework for evaluating a medical device within its operating environment is as follows: (a) different modes of device operation are specified; (b) the device functional model is developed and validated in the context of the patient's physiology; (c) the formal model is developed and interfaced with the formal model of the patient; and finally (d) an experimental evaluation is conducted between the patient model and the implementation of the device.

In this study we specifically focus on the artificial pacemaker device (hence forth referred as pacemaker). The pacemaker was selected as it is a relatively simple device, which has been widely used for a number of decades, and yet has been subject to a large number of recalls. We designed and validated the functional pacemaker model using the two most frequent arrhythmias. The formal model of the pacemaker was designed and verified within the context of the VHM using Simulink Design Verifier[4] (Fig. 14). The results are applicable only to the cases investigated and further results may be found in the technical report[15].

A. Artificial Pacemaker

In order to better understand the need and the operation of the pacemaker, we provide some background. Rhythm man-

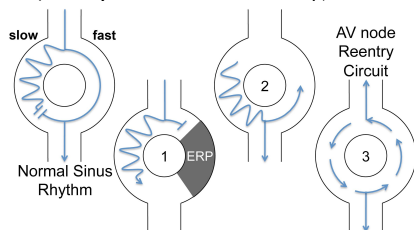


Figure 12. AVNRT in different timing scenarios with the accessory pathway. Each circuit is isolated in the AV node. Conduction at normal rhythm is shown on the left, where the two signals *conflict* in the slow pathway. 1, 2, and 3 show the progression towards AVNRT. Early activation reaches a block in the fast path and conducts down the slow path (1). The activation then travels retrograde up the fast path (2) and continues to cycle around (3), activating up to the atria and down to the ventricles.

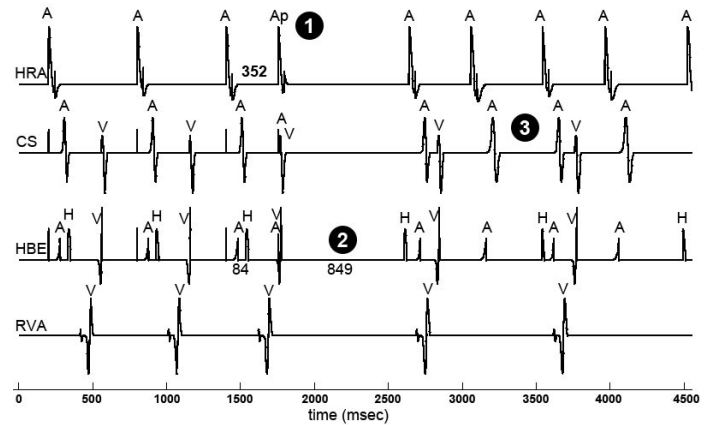


Figure 13. Induced AVNRT by extrastimuli pacing in the heart model. The normal cycle length is 700 ms, and the extrastimuli occurs after only 380ms at (1), causing a prolonged Ap-H interval (2) and inducing AVNRT, the fast atrial rate causes a 2:1 conduction from A to V (see the CS electrogram (3)).

Position	I	II	III	IV	V
Category	Chambers paced	Chambers Sensed	Response to sensing	Programmability rate, modulation	Antiarrhythmia functions
Letters	O=none A=atrium V=ventricle D=dual (A&V)	O=none A=atrium V=ventricle D=dual (A&V)	O=none T=triggered I=inhibited D=dual (T&I)	O=none P=simple programmable M=multiprogrammable C=communicating R=rate modulation	O=none P=pacing (antiarrhythmia) S=shock D=dual(P&S)

Table II
HEART RHYTHM SOCIETY - GENERIC PACEMAKER CODE

agement from a cardiac pacemaker is a necessary therapy for patients with brady and tachyarrhythmias. The pacemaker leads are implanted inside the patient's heart in the wall of the atrium, ventricle, or both chambers of the heart. Electrodes on the lead can have both sensing and pacing abilities, depending on the mode in which the pacemaker is operating. The Heart Rhythm Society has developed a five-position code to describe the modality of the pacemaker (Table II), though the first three positions contain the most pertinent information. Position I describes the pacing locations, position II describes the sensing locations, and position III describes how the device responds to sensing. The physician chooses the pacemaker mode based on the location and nature of the patient's specific arrhythmias.

B. Pacemaker model

The pacemaker model has the capability to operate in any mode, but specifically the AAI and DDD modes have been implemented and tested with our heart model. AAI is single chamber mode that provides demand pacing from an atrium lead. If an intrinsic event is sensed by the lead, the pacing function is inhibited; otherwise, it will pace the chamber. This mode cannot ensure synchrony between the atrial and ventricular rhythms, which can be problematic and lead to pacemaker syndrome and inefficient pumping. DDD is a more

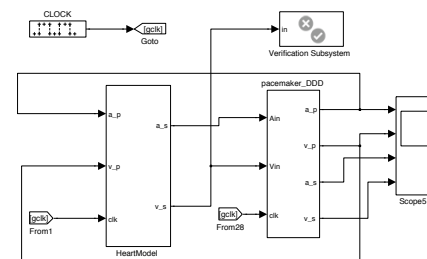


Figure 14. Closed Loop Model in Simulink

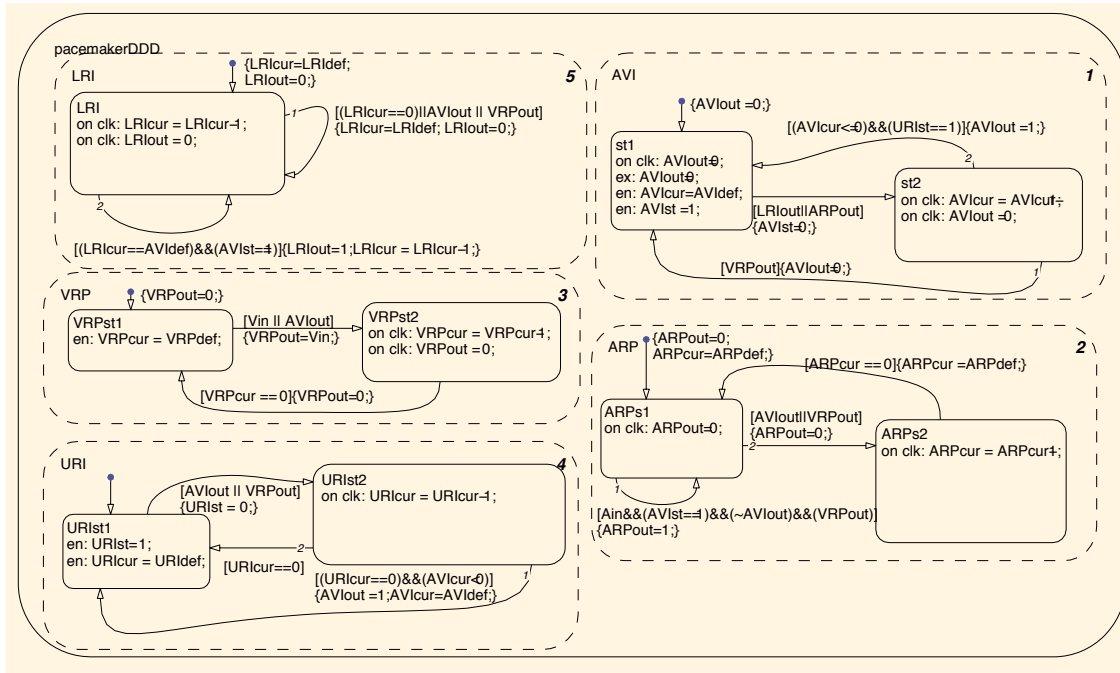


Figure 15. Simulink Stateflow model of Pacemaker

sophisticated mode of operation and is implemented in dual chamber pacemakers. The device can both sense and pace from the atrium and the ventricle, which allows for synchronous pacing and proper hemodynamics. The 5 basic timing cycles of the DDD mode pacemaker are shown in Fig. 16[16]. In our pacemaker model, we designed 5 corresponding software components, shown in Fig. 15. These components run in parallel and are triggered by events. We group the components as follows:

1. Lowest Rate Interval (LRI) is the most basic timing cycle and it is the only timing cycle in our AAI mode. This component keeps the heart rate above a minimum value. The LRI timer is reset after there is a sensed cardiac event (atrial event in AAI mode and ventricular event in DDD mode). For AAI mode, if no atrial event has been sensed (AS) before the timer runs out, the pacemaker will deliver pacing signal from the atrial lead (AP). For DDD mode, if no ventricular event has been sensed (VS) before the timer runs out, the pacing signal will be delivered from the ventricular lead (VP).

2. Atrio-Ventricular Interval (AVI) component only appears in dual chamber modes of pacemaker. The function of this components mimic the intrinsic AV delay to synchronize the atria and ventricle events. The timer is started by a sensed or paced atrial event and can be terminated by a sensed ventricular event. If no ventricular event is sensed before the timer times out, the pacemaker will pace the ventricle.

3. Three auxiliary components which filter out noise and

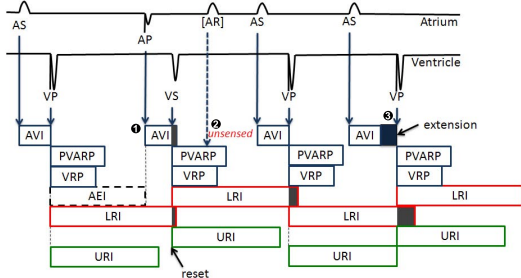


Figure 16. Pacemaker timing cycles.

early events which would otherwise cause undesired pacemaker behavior. The post ventricular atrial refractory period (PVARP) component is a blocking interval where atrial sensing cannot occur. It mimics the atrial refractory period, during which no atrial events should occur for proper function. The ventricular refractory period (VRP) component is the blocking interval for ventricular events. The upper rate interval (URI) component provides an upper bound for ventricular pacing.

C. Pacemaker Functional Model Validation

By simulating clinical cases where pacemaker therapy is needed using our heart model and comparing the effects using different modes of pacemakers, we demonstrate how our model can be used to validate pacemaker design and help physicians to determine an appropriate pacemaker mode. The following cases are common arrhythmias and are considered clinically-relevant by an electrophysiologist.

1) Case 1: Sinus Bradycardia: This is the most basic arrhythmia that can be controlled by a pacemaker, where the patient's SA node fires at a rate less than 60 beats per minute. Fig. 17 shows the heart model simulating a bradycardia event. As the heart model is running, event (1) triggers the T_{rest} interval of the SA node to lengthen, causing bradycardia. When the AAI pacemaker model interacts with the bradycardia heart model, the device intervenes at event (1) and delivers pacing to

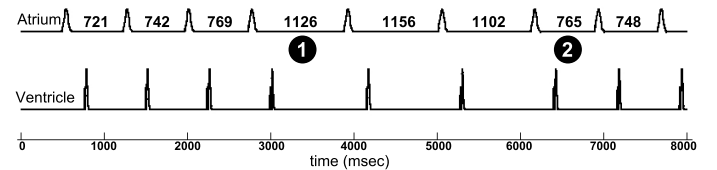


Figure 17. Case 1: Sinus bradycardia without pacemaker. The figure shows the atrial and ventricle electrograms recorded by the pacemaker model. The heart model starts in Normal Sinus Rhythm (NSR), with beat-to-beat interval of approximately 750 ms. Event 1 changes the T_{rest} timer of SA node, causing bradycardia with beat-to-beat interval stretched to 1100 ms. Event 2 changes the same T_{rest} to resume NSR.

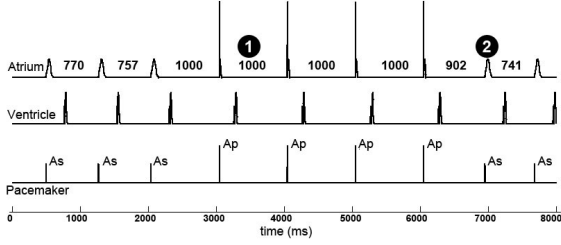


Figure 18. Case 1: Sinus bradycardia with AAI pacemaker. The bottom panel shows the sensing and pacing state of the pacemaker model. At (1) the pacemaker delivers a pacing stimulus (AP) when interval between two atrium events is longer than LRI(1). At (2), pacing is inhibited by a sensed atrial event.

the atrium (Fig. 18). This pacing signal is conducted throughout the heart and triggers a ventricular event, maintaining 1:1 conduction in the heart.

2) *Case 2: Sinus Bradycardia with Second degree heart block (Wenckebach type)*: A patient with bradycardia and abnormal A-V conduction system was simulated in Fig. 19. The mechanism of this kind of heart block was shown in Section III. The EGM in Fig. 20 shows gradually increased AV delay due to fast atrial pacing until there is a dropped beat. A DDD mode pacemaker is shown to maintain a steady ventricular rate (Fig. 21) and prevent the dropped beat.

In this subsection we validated our pacemaker model with two most common arrhythmias requiring the assistance of the pacemaker. We also showed that the heart model can be used as a tool to assist cardiologists in selecting appropriate mode.

D. Formal Verification

The major benefit of the VHM is that the exposure of the formal interface allows use of the formal methods for verification of close-loop designs. Several verification tools are available that can be used for checking of structural and safety properties along with the test cases generation.

Simulink Design Verifier (SDV)[4] exploits a Prover technology based formal method proving engine. To prove a property using SDV we created a Simulink sub-system that describes a temporal condition that needs to be obtained by the signal. For example, Fig. 22 shows a structure of the *Verification Subsystem* from Fig. 14, which is used to verify property that the pacemaker in DDD mode of operation will keep the right ventricular inter-pacing interval between two consecutive beats, in $[0.9s, 1.1s]$ range (since in the pacemaker is modeled to work with 1ms clock). Due to space limitations, more details about other designed sub-systems can be found in [15].

UPPAAL[17] is a widely used tool for system verification, where properties must be expressed using temporal logic formulas before they are checked. In order to use UPPAAL the kernel of the VHM along with its formal interface have to be ported from Stateflow to UPPAAL.

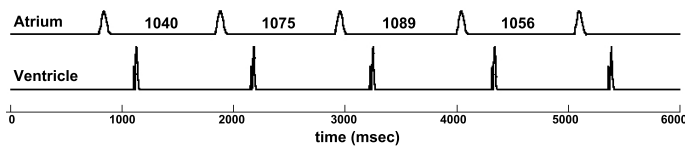


Figure 19. Case 2: Sinus bradycardia with concealed second degree heart block (Wenckebach type). Bradycardia with beat-to-beat intervals around 1200 ms produces 1:1 atrium to ventricle conduction.

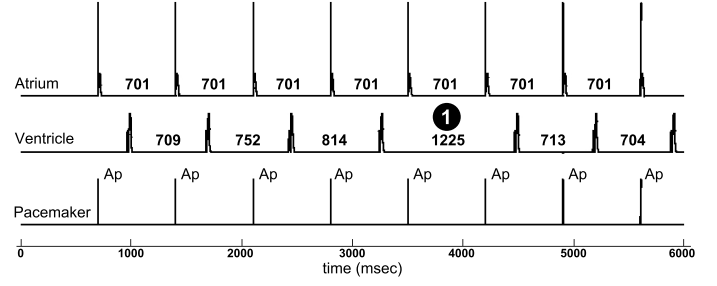


Figure 20. Case 2: Sinus bradycardia with second degree heart block (Wenckebach type) with AAI pacemaker. The coupling interval of the pacing stimuli is shorter than the RRP of AV node, which lengthens the ERP and the conduction delay of AV node until a dropped beat occurs in the ventricle. AAI pacemaker cannot maintain ventricle rate

V. PHYSICAL IMPLEMENTATION

The Virtual Heart Model (VHM) was implemented on Xilinx Spartan-3, XC3S1000 FPGA [18]. The automatas described in Section II were designed to work with a 10MHz clock. The transition from Simulink/Stateflow design to VHDL description was done manually although appropriate tools exist, which can automatically extract VHDL code from Stateflow design ([19], [20]). Since manual migration from Simulink to VHDL can be a cause of discrepancy between two designs, the use of automatic translators should be a part of future VHM models.

In order to demonstrate how the VHM model can be used for closed-loop black-box testing of implantable medical devices, we used a setup shown in Fig. 23. The pacemaker design described in previous section was implemented on FireFly sensor nodes [21]. The nodes run the nano-RK [22], real-time operating system developed with timeliness as a first-class concern. On FireFlies, nano-RK operates with a 1ms OS tick.

The pacemaker was implemented on a FireFly node using five tasks, corresponding to the automatas from Fig. 15, 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 ventricular 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 from Fig. 15 (lower value has greater priority). 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 behavior of the system.

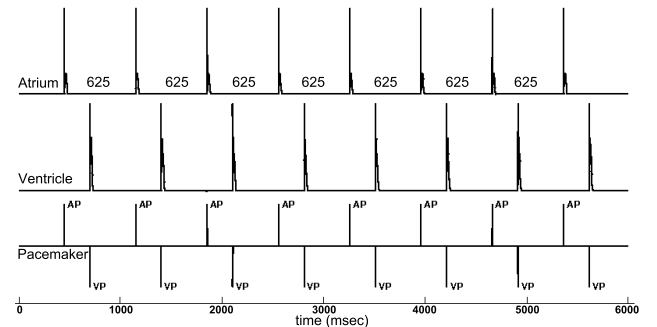


Figure 21. Case 2: Sinus bradycardia with second degree heart block (Wenckebach type) with DDD pacemaker. The pacemaker maintains the synchronization between the atria and ventricles by pacing the ventricle (VP)

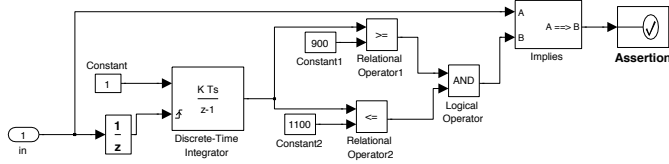


Figure 22. Design Verifier Block in Simulink

VI. RELATED WORK

Researchers have tackled the problem of whole-heart modeling using a variety of different methods. Modern techniques are capable of producing high-resolution techniques almost down to the cellular level. Jalife et al [8] modeled the activity of single ion channels in order to get a detailed description of a single cell. This model can produce action potentials nearly identical to physiological ones, but creating a full model of the heart based on such details within single cells is not computationally possible in real time. Harrild and Henriquez [9] designed a finite element three dimensional mesh model with resolution close to the cellular level. They were able to model the potentials across both atria, but could not extend the model to the entire heart. In order to accommodate the memory and time constraints presented by current computing tools, empirical models are the best solution to allow for full heart modeling. These models, such as ones created by Adam [10] and Berenfeld et al. [7], simplify the heart tissue and do not attempt computing on the cellular level. Adam analyzed the wavefront of depolarization within his cardiac model, but he was unable to recreate the repolarization front, and thus he could not produce reentry circuits and arrhythmias.

Formal methods have traditionally been used for verification of time-critical and safety-critical embedded systems [23]. Until recently these methods have not been used for medical device certification. The authors in [24] presented the use of Extended Finite State Machines for model checking of the Computer Automated Resuscitation A medical device. Formal techniques have also been applied to improve medical device protocols ([25]) and safety ([26]). However, the authors either used a simplified patient model or did not model the patient at all.

VII. CONCLUSION

A primary challenge in life-critical real-time systems is with the design of bug-free medical device software. While

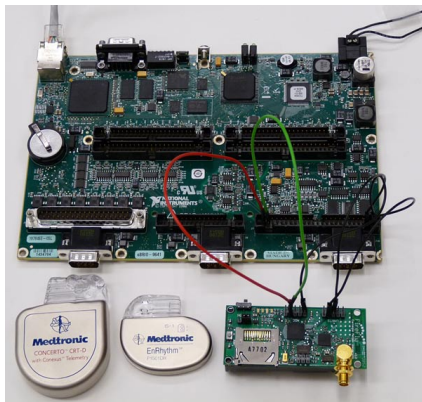


Figure 23. Closed-loop experimental setup. Medtronic cardiac defibrillator and pacemaker models are shown for reference.

implantable cardiac devices are being inserted into millions of patients worldwide, the number of recalls due to firmware problems are significant and growing in frequency. There is, therefore, an urgent need for methodologies for medical device certification within the context of the biological environment. We present an integrated approach for medical device software validation and verification with a focus on artificial cardiac pacemakers. To this effect, a real-time Virtual Heart Model (VHM) has been developed to model the electrophysiological operation of the human heart and interfaces both functionally and formally with a pacemaker model. This approach will potentially help expedite medical device certification for safer operation. The platform is available as a free and open source platform to the research community.

REFERENCES

- [1] W. H. Maisel et. al. Recalls and safety alerts involving pacemakers and implantable cardioverter-defibrillator generators. *J. American Med. Ass.*, 286, 2001.
- [2] Center for Devices and Radiological Health. *Ensuring the Safety of Marketed Medical Devices*. US FDA, 2006.
- [3] I. Lee et. al. High-Confidence Medical Device Software and Systems. *IEEE Computer*, Vol. 39, Is. 4, 2006.
- [4] Simulink design verifier 1.5. mathworks.
- [5] R. Alur. *Timed Automata*. Lecture Notes in Computer Science, Springer, 1999.
- [6] R.N. Fogoros. *EP Testing*. Blackwell Science, 1999.
- [7] 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, No. 8, pp. 615-625, 1996.
- [8] J. Beaumont et. al. A model study of changes in excitability of ventricular muscle cells. *Am. J. Physiol.* 268 (*Heart Circ Physiol.* 37), 1995.
- [9] D. Harrild and C. Henriquez. A computer model of normal conduction in the human atria. *Circ. Res.*, 2000.
- [10] D. R. Adam. Propagation of depolarization and repolarization processes in the myocardium - an anisotropic model. *IEEE Trans. Biomed. Eng.* Vol 38, No. 2, pp. 133-141, 1991.
- [11] W. Stevenson and K. Soejima. Recording Techniques for Clinical Electrophysiology. *J Cardio EP*, 2005.
- [12] M.E. Josephson. *Clinical Cardiac Electrophysiology*. Lippincott Williams and Wilkins, 2008.
- [13] M. Franz et al. Cycle length dependence of human action potential duration in vivo. *J. of Clinical Investigation*, 1988.
- [14] Zipes and Jalife. *Cardiac Electrophysiology: From cell to bedside*. Saunders, 2004.
- [15] Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam. Virtual heart model, technical report ese-2010-01, 2010.
- [16] S. Barold, R. Stroobandt, and A. Sinnaeve. *Cardiac Pacemakers Step by Step*. Blackwell Futura, 2004.
- [17] K.G. Larsen et al. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1997.
- [18] Spartan-3 FPGA Datasheet.
- [19] Simulink HDL Coder 1.6. Mathworks, 2009.
- [20] K. Camera. *SF2VHD: A StateFlow to VHDL Translator*. Masters Thesis, UC Berkeley, 2001.
- [21] R. Mangharam, A. Rowe, and R. Rajkumar. FireFly: A Cross-layer Platform for Real-time Embedded Wireless Networks. *Real-Time System Journal*, 2007.
- [22] nano-RK Sensor RTOS. <http://nanork.org>.
- [23] E.M. Clarke and J.M. Wing. Formal methods: state of the art and future directions. *ACM Comput. Surv.*, 28(4), 1996.
- [24] D. Arney et. al. Formal Methods Based Development of a PCA Infusion Pump Model. *HCMDSS Workshop*, 2007.
- [25] R. Alur et. al. Formal Spec. and Analysis of the CARA Infusion Pump Control System. *J. on Software Tools for Tech.*, 2004.
- [26] A. Teije et. al. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36, 2006.