Closed-Loop Modeling of Cardiac Pacemaker and Heart

Dominique Méry[1] and Neeraj Kumar Singh[2]

[1] Université de Lorraine, LORIA, BP 239, Nancy, France
Dominique.Mery@loria.fr
[2] Department of Computer Science, University of York, United Kingdom
neeraj.singh@cs.york.ac.uk

Abstract. The development of critical medical systems requires high levels of confidence in increasingly complex software systems. Formal methods have been identified as a means of contributing to assurance in this domain. We present a closed-loop modeling approach between an electrocardiography analysis based heart model and pacemaker. This stem is a step towards a modeling approach for medical systems at early stage of the system development. Implantable devices like cardiac pacemakers and implantable cardioverter-defibrillators require closed-loop modeling (integrated system and environment modeling) to qualify the certification standards. The industry has long sought such an approach to validating a system model in a virtual biological environment. This approach involves a pragmatic combination of formal specifications of the system and the biological environment to model a closed-loop system that enables verification of the correctness of the system and helps to improve the quality of the system.

Keywords: Heart Model, ECG, Cellular Automata, Event-B, Closed-loop model, Proof-based development, Refinement.

1 Introduction

In the area of medical engineering, cardiac pacemaker and implantable cardioverter-defibrillators are considered as remarkable innovations of the past century, used for saving millions of lives worldwide. The implantation rate of these devices has been increased [1–3]. Malfunctions related to the hardware and firmware are considered as a common type of defects for both pacemakers and implantable cardioverter-defibrillators [1, 4, 5]. During the 1990s, 17323 devices were explanted due to malfunction [3]. In 1996, 10% of medical device recalls were caused by software-related issues. In 2010, the Food and Drug Administration (FDA) reported 23 cases of defective devices, where some of the cases were due to software defects [1, 5–7].

Nowadays, manufacturers use standard guidelines for system development. These standards include software evaluation, which covers mainly code inspection, static analysis, module-level testing and integration testing. The purpose is to use these standards to establish *reasonable assurance of safety and effectiveness.* However, these approaches are not sufficient to check the software correctness. Testing — combined with finding bugs at the final stage of system development — is very expensive. As software plays an increasingly more important role in medical devices and in healthcare-related activities more generally, regulatory agencies such as the FDA, and certification

J. Weber and I. Perseil (Eds.): FHIES 2012, LNCS 7789, pp. 151–166, 2013.

bodies such as the FDA's Quality System Regulation and the International Standards Organization's 13485 [8, 7, 9], need effective methods for ensuring that newly developed software-based healthcare systems are *safe* and *reliable*. Regulatory agencies, in addition to the medical device manufacturers themselves, have been striving for a more rigorous engineering-based review strategy to provide this assurance [10]. Traditional methods of system development are not using formal techniques for verifying the correctness of the system requirements. An effective way of finding bugs at an early stage of the system development is practical application of formal methods. Formal methods have been successful in targeted applications of medical devices [11–14, 10, 9]. Over the past decade, there has been considerable progress in the development of formal methods for improving confidence in complex software-based systems [15, 16].

Software bugs and unexpected behaviors of the system are not easy to find from system specifications alone. To apply formal methods for verifying the specification of such complex systems is not enough. Such systems require a *closed-loop modeling approach*, where formal models of the system and an environment form a closed-loop model. The closed-loop model captures the possible behaviors of the system under environmental conditions. Such closed-loop modeling is the primary technique in system engineering and *cyber-physical* systems.

Verifying the correct behavior of a system model using an environment, is a challenging problem, where the system model and environment are both developed using identical formal notations. For example, a formal model of a cardiac pacemaker or implantable cardioverter-defibrillators requires a heart model to verify the correctness of the developed system (see Fig. 1). No tools and techniques exist for environment modeling that would enable verification of the developed system model. Most medical devices are tightly coupled with their biological environment (i.e., the heart), where these devices use sensors and actuators as interaction points. The integration of the heart and pacemaker is formally modelled and provides a good example of medical device integration [17]. In our previous work [18], we have developed a mathematical heart model. This heart model is an electro-physiological model, which models the timing and electrical conduction of the heart with both intrinsic and artificial pacing signals. In this paper, we recall the heart model for closed-loop modeling of pacemaker functionality for identifying complex behavior of the system. In the closed-loop model, the heart and pacemaker interact with each other [17]. The pacemaker responds according to the heart requirements. The heart generates all possible behaviors of the normal and abnormal conditions. The focus of this effort is three-fold: (a) we develop a mathematical heart model based on logico-mathematical theory, which provides a set of general and patient condition-specific pacemaker software requirements to ensure the safety of the patient, (b) we develop both cardiac pacemaker and heart models for closed-loop modeling, (c) we verify the closed-loop system over a variety of basic operations where the heart rate must be maintained and the atrial-ventricular synchrony must be maintained through formal proofs of the system.

The rest of this paper is organized as follows. Section 2 summarizes the construction of the heart model, which is extensively described in our previous publication [18]. Section 3 presents a closed-loop formal model of a pacemaker which interacts with the heart model. The closed-loop requirements are described in Section 4. Section 5

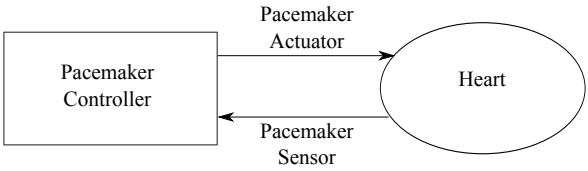


Fig. 1. Cardiac pacemaker and Heart interaction

discusses lessons learned from this experience, and Section 6 concludes the paper with some perspectives together with proposals for future work.

2 Heart Model

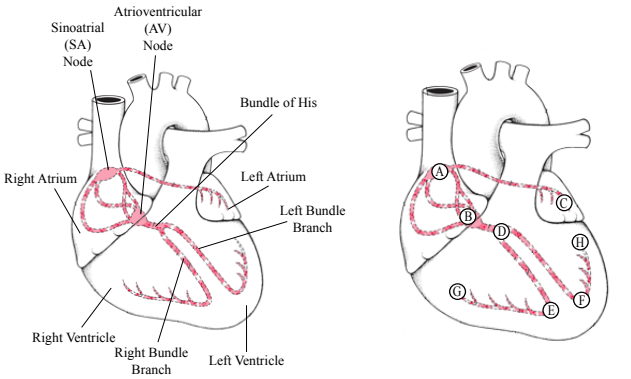
The heart consists of four chambers (see Fig. 2(a)): right atrium, right ventricle, left atrium and left ventricle, which contract and relax periodically. The natural heart's system requires an electrical stimulus, which is generated by the small mass of specialized tissue located in the right atrium called the sinus node. This electrical stimulus travels down through the conduction pathways and causes the heart's chambers to contract and pump out blood. Each contraction of the ventricles represents one heartbeat. The atria contract for a fraction of a second before the ventricles, so their blood empties into the ventricles, before the ventricles contract.

Fig. 2(a) presents a set of basic components and an impulse conduction path of the heart. The electrical current flows progressively in the heart muscle using special conduction cells. To model the heart system abstractly, we consider a set of landmark nodes (A, B, C, D, E, F, G, H) in the entire conduction network (see Fig. 2(b)), which provides a control behavior of the heart. These landmarks were identified in literature surveys [19–22] and extensive discussions with two experts, a cardiologist and a physiologist.

This section presents an elementary information about the heart modeling, which helps the reader to understand the modeling of the closed-loop system. A detailed description about the heart system and formalization steps are available in [18, 23]. We introduce the necessary elements using formal notations to define the heart system as follows:

Definition 1 (The Heart System). *Given a set of nodes N, a transition (conduction) t is a pair (i, j), with i, j* $\in N$ *. A transition is denoted by i* $\rightsquigarrow$ *j. The heart system is a tuple HSys = (N, T,* N_0*,* TW_{time}*,* CW_{speed} *) where:*

• $N = \{ A, B, C, D, E, F, G, H \}$ *is a finite set of landmark nodes in the conduction pathways of the heart system;*
• $T \subseteq N \times N = \{A \mapsto B, A \mapsto C, B \mapsto D, D \mapsto E, D \mapsto F, E \mapsto G, F \mapsto H\}$ *is a set of transitions to represent electrical impulse propagation between two landmark nodes;*
• $N_0 = A$ *is the initial landmark node (SA node);*



(a) Electrical Conduction system (b) Landmarks in network

Fig. 2. The Electrical Conduction and Landmarks of the Heart System [18]

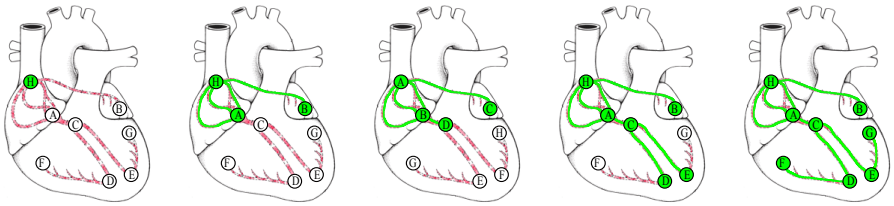
• $TW_{time} \in N \rightarrow TIME$ *is a weight function as time delay of each node, where* $TIME$ *is a range of time delays;*
• $CW_{speed} \in T \rightarrow SPEED$ *is a weight function for the impulse propagation speed of each transition, where* $SPEED$ *is a range of propagation speed.*

Property 1 (Impulse Propagation Time). *In the heart system, the electrical impulse originates from the SA node (node A), travels through the entire conduction network and terminates at the atrial muscle fibres (node C) and at the ends of the Purkinje fibres in both sides of the ventricular chambers (node G and node H). The impulse propagation time delay differs for each landmark node (N). The impulse propagation time is represented as the total function* $TW_{time} \in N \rightarrow \mathbb{P}(0..230)$. *The impulse propagation time delay for each node (N) is represented as:* $TW_{time}(A) = 0..10$, $TW_{time}(B) = 50..70$, $TW_{time}(C) = 70..90$, $TW_{time}(D) = 125..160$, $TW_{time}(E) = 145..180$, $TW_{time}(F) = 145..180$, $TW_{time}(G) = 150..210$ *and* $TW_{time}(h) = 150..230$.

Property 2 (Impulse Propagation Speed). *The impulse propagation speed also differs for each transition (i* $\rightsquigarrow$ *j, where i, j* $\in$ *N). The impulse propagation speed is represented as the total function* $CW_{speed} \in T \rightarrow \mathbb{P}(5..400)$. *The Impulse propagation speed for each transition is represented as:* $CW_{speed}(A \mapsto B) = 30..50$, $CW_{speed}(A \mapsto C) = 30..50$, $CW_{speed}(B \mapsto D) = 100..200$, $CW_{speed}(D \mapsto E) = 100..200$, $CW_{speed}(E \mapsto G) = 300..400$ *and* $CW_{speed}(F \mapsto H) = 300..400$.

Electrical activity is spontaneously generated by the SA node, which propagates through the conduction network in the entire heart system using several intermediate landmark nodes (see Fig. 2). The electrical system synchronizes the contraction between atria and ventricles. To change time intervals or conduction speeds between landmarks (see Fig. 2(b) and Fig. 2(a)) are a major cause of abnormalities in the heart system. Abnormalities in electrical signals in the heart can generate various kinds of arrhythmias. A slow conduction speed generates bradycardia and a fast conduction speed generates tachycardia. In this model, we consider the range of all possible values for

conduction speeds and conduction times for each landmark node and conduction path (see Table 1). This model represents the morphological structure of the ECG signal through the conduction network (see Fig. 2(a)).



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4 (e) Step 5

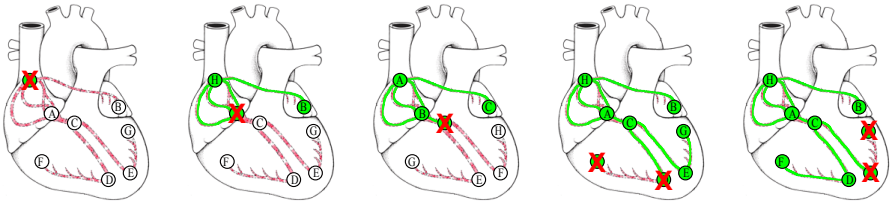
Fig. 3. Impulse Propagation through Landmark nodes [18]

Table 1. Cardiac Activation Time and Cardiac Velocity [19]

Location in the heart	**Cardiac Activation Time (ms.)**	**Location in the heart**	**Conduction Velocity (cm/sec.)**
SA Node (A)	0..10	A $\mapsto$ B	30..50
Left atria muscle fibers (C)	70..90	A $\mapsto$ C	30..50
AV Node (B)	50..70	B $\mapsto$ D	100..200
Bundle of His (D)	125..160	D $\mapsto$ E	100..200
Right Bundle Branch (E)	145..180	D $\mapsto$ F	100..200
Left Bundle Branch (F)	145..180	E $\mapsto$ G	300..400
Right Purkinje fibers (G)	150..210	F $\mapsto$ H	300..400
Left Purkinje fibers (H)	150..230		

Heart block is the term given to a disorder of conduction of the impulse that stimulates heart muscle contraction. The normal cardiac impulse arises in the SA node (A), situated in the right atrium, and spreads to the AV node (B), whence it is conducted by specialized tissue known as the Bundle of His (D), which divides into the left and right bundle branches in the ventricles (see Fig. 2(a)). Disturbances in conduction may appear as slow conduction, intermittent conduction failure or complete conduction failure. These three kinds of conduction failure are also known as 1st, 2nd and 3rd degree blocks. We can show these different kinds of heart block throughout the conduction network in terms of our set of landmark nodes (see Fig. 4).

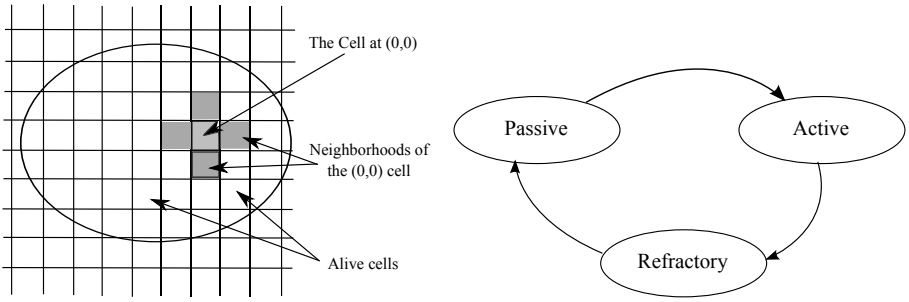
A set of spatially distributed cells form a Cellular Automata (CA) model, which contains a uniform connection pattern among neighbouring cells and local computation laws. CA are discrete dynamic systems corresponding to space and time, which provide uniform properties for state transitions and interconnection patterns. The cardiac muscle cells of the heart are presented in the following states: $Active$, $Passive$ or $Refractory$. Initially, all cells are $Passive$, where each cell is discharged electrically and has no



(a) SA Block (b) AV Block (c) Infra-Hisian (d) RBBB (e) LBBB

Fig. 4. Impairments in Impulse Propagation due to the Heart Blocks [18]

influence on its neighbouring cells. When an electrical impulse propagates, the cell becomes charged and eventually activated ($Active$ state). The $Active$ cell transmits an electrical impulse to its neighbour cells. The electrical impulse is propagated to all the cells in the heart muscle. After activation, the cell becomes discharged and enters the $Refractory$ state within which the cell can not be reactivated. After a time, the cell changes its state to the $Passive$ state to await the next impulse.



(a) A two-Dimensional Cellular Automata Model (b) State Transition of a Cell

Fig. 5. Two-Dimensional Cellular Automata and State Transition Model [18]

3 Closed-Loop Model of Heart and Cardiac Pacemaker

This section describes a closed-loop formal model of a cardiac pacemaker and of the heart system, where the cardiac pacemaker responses according to the functional behavior of the heart [18, 23]. The main objective of this model is to verify the complex properties of the cardiac pacemaker under the virtual environment. Fig. 1 represents a block diagram of the cardiac pacemaker and of the heart system, where the cardiac pacemaker responses, when it senses intrinsic activities from the heart. In this system specification, the heart model simulates the functional behavior of the normal and abnormal heart rate. The heart model activities are always monitored by the cardiac pacemaker and it responses according to the user needs.

In our previous work, we have already developed the formal model of the cardiac pacemaker [24] and of the heart system [18]. This paper presents a closed-loop model of the cardiac pacemaker, where the heart is used as an environment. For developing this closed-loop model, we borrow formal specifications from the previously developed and verified formal models of the cardiac pacemaker [24] and heart system [18]. However, to develop the closed-loop model, we have done substantial changes in the existing models to specify the desired behavior of the system. Moreover, we develop the whole system from scratch using progressive refinements. Each refinement level introduces both cardiac pacemaker and heart system behaviors. To check the correctness of the closed-loop system, we have introduced safety properties using invariants, and discharged all the generated proof obligations at each refinement level. Due to space limitations, the following section formalizes the closed-loop system abstractly.

3.1 The Context and Initial Model

To formalize the heart behavior, we capture the electrical features. We identify a set of landmark nodes from the conduction network (see Fig. 2(a)) of the heart. These landmark nodes are also known as the electrical impulse propagation nodes *ConductionNode*, which enable expression of the normal and abnormal behaviors of the heart system. We find the direct connections among the impulse propagation nodes, which constitute the impulse propagation path. The impulse propagation time and the impulse propagation velocity for each pair of nodes vary due to different types of muscles in the heart. To formalize the heart system, we define three constants impulse propagation time *ConductionTime*, impulse propagation path *ConductionPath* and impulse propagation velocity *ConductionSpeed*. All these constants are initial components, which are defined through a set of axioms ($axm1$-$axm4$). To formalize the cardiac pacemaker, we define a set of constants (LRL, URL, ARP, VRP, $PVARP$ etc.), which express timing intervals. These timing intervals are used as a set of configuration parameters. To model a boolean behavior of the sensor and actuator, we define an enumerated set $status$. Axioms for the cardiac pacemaker are defined by $axm5$ and $axm6$. All these constants and axioms have been extracted from the definitions (see Section 2) and technical specification [25], that are validated by the cardiologist and the physiologist.

$$
\begin{array}{l}
axm1 : partition(ConductionNode, \{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}, \{G\}, \{H\}) \\
axm2 : ConductionTime \in ConductionNode \rightarrow \mathbb{P}(0 \mathinner{..} 230) \\
axm3 : ConductionPath \subseteq ConductionNode \times ConductionNode \\
axm4 : ConductionSpeed \in ConductionPath \rightarrow \mathbb{P}(5 \mathinner{..} 400) \\
axm5 : LRL \in 30 \mathinner{..} 175 \land URL \in 50 \mathinner{..} 175 \land PVARP \in 150 \mathinner{..} 500 \\
axm6 : ARP \in 150 \mathinner{..} 500 \land VRP \in 150 \mathinner{..} 500 \land status = \{ON, OFF\}
\end{array}
$$

To define an abstract model of the closed-loop system, we develop the combined model of the cardiac pacemaker and of the heart, where the cardiac pacemaker acts according to the heart behavior. The environment model of the heart behaves according to the observations of the impulse propagation in the conduction nodes. We define a set

of variables to model the heart and pacemaker models, where four variables (*ConductionNodeState*, *CConductionTime*, *CConductionSpeed* and *HeartState*) are used to model the heart behavior, and six variables($PM\_Actuator\_A$,$PM\_Actuator\_V$, PM_Sensor_A, $PM - \_Sensor\_V$, $Pace\_Int$ and sp) are used to express the cardiac pacemaker behavior. All these variables are defined using a list of invariants ($inv1$-$inv7$). The cardiac pacemaker variables are introduced for modeling actuators, sensors and timing intervals. A list of invariants ($inv8$,$inv9$ and $inv10$) presents safety properties. The invariant $inv8$ states that, when the clock counter sp is less than VRP and atrioventricular (AV) counter state AV_Count_State is FALSE, then the pacemaker's actuators and sensors of both chambers are OFF. Similarly, the next invariants ($inv9$ and $inv10$) represent the required properties of ON state of the pacemaker's actuators in both chambers.

$$
\begin{array}{l}
inv1 : ConductionNodeState \in ConductionNode \rightarrow BOOL \\
inv2 : CConductionTime \in ConductionNode \rightarrow 0 \,..\, 300 \\
inv3 : CConductionSpeed \in ConductionPath \rightarrow 0 \,..\, 500 \\
inv4 : HeartState \in BOOL \\
inv5 : PM\_Actuator\_A \in status \wedge PM\_Actuator\_V \in status \\
inv6 : PM\_Sensor\_A \in status \wedge PM\_Sensor\_V \in status \\
inv7 : Pace\_Int \in URI \,..\, LRI \wedge sp \in 1 \,..\, Pace\_Int \\
inv8 : sp < VRP \wedge AV\_Count\_STATE = FALSE \Rightarrow \\
\qquad PM\_Actuator\_V = OFF \wedge PM\_Sensor\_A = OFF \wedge \\
\qquad PM\_Sensor\_V = OFF \wedge PM\_Actuator\_A = OFF \\
inv9 : PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int \vee (sp < Pace\_Int \wedge \\
\qquad AV\_Count > V\_Blank \wedge AV\_Count \geq FixedAV) \\
inv10 : PM\_Actuator\_A = ON \Rightarrow (sp \geq Pace\_Int - FixedAV)
\end{array}
$$

The abstract specification of the closed-loop model contains several events related to the cardiac pacemaker and to the heart system. There are many events, namely *HeartOK* to represent a normal state of the heart, *HeartKO* to express an abnormal state of the heart, *HeartConduction* to trace the current updated value of each landmark node in the conduction network, *Actuator_ON_V*, *Actuator_OFF_V*, *Actuator_ON_A* and *Actuator_OFF_A* to represent ON and OFF states of the pacemaker's actuators for both chambers, *Sensor_ON_A*, *Sensor_OFF_A*, *Sensor_ON_V*, and *Sensor_OFF_V* to represent ON and OFF states of the pacemaker's sensors for both chambers, and *tic* to represent clock counter. Due to space limitations, we describe few events in detail.

The event *HeartOK* expresses desired behavior of the normal heart, where a set of guards formulates the required conditions. The first guard ($grd1$) states that all the landmark nodes must be visited for one cycle during impulse propagation using conduction network. The second guard specifies that the current impulse propagation time for each landmark node should be ranged in the pre-specified ranges (*Property 1*). Similarly, the last guard states that the current impulse propagation velocity of each path should range between pre-defined impulse propagation velocities (*Property 2*). The action predicate ($act1$) denotes the normal state of the heart, when these guards are satisfied.

EVENT HeartOK
WHEN
$grd1 : \forall i \cdot i \in ConductionNode \Rightarrow ConductionNodeState(i) = TRUE$
$grd2 : \forall i \cdot i \in ConductionNode \Rightarrow CConductionTime(i) \in ConductionTime(i)$
$grd3 : \forall i, j \cdot \left(\begin{array}{l} i \mapsto j \in ConductionPath \\ \Rightarrow \\ CConductionSpeed(i \mapsto j) \in ConductionSpeed(i \mapsto j) \end{array} \right)$
THEN
$act1 : HeartState := TRUE$
END

In the two electrodes pacemaker, we use two sensors and two actuators for capturing the required behavior of the cardiac pacemaker. In this section, we show only actuator and sensor events of the ventricle chamber. Moreover, other events related to the sensor and actuator of the atrial chamber are identical. Events $Actuator\_ON\_V$ and $Sensor\_ON\_V$ are excerpt from the abstract model to describe ON state of the actuator and sensor of the cardiac pacemaker. A list of guards of both events enables to set ON state of both actuator and sensor, allowing to pace and to sense in the ventricular chamber under the desired conditions using real-time constraints. A detailed formalization of the other events related to the cardiac pacemaker are described in [24, 26].

EVENT Actuator_ON_V
WHEN
grd1 : $PM\_Actuator\_V = OFF$
grd2 : $(sp = Pace\_Int) \lor$
$(sp < Pace\_Int \land$
$AV\_Count > V\_Blank \land$
$AV\_Count \geq FixedAV)$
grd3 : $sp \geq VRP \land sp \geq PVARP$
$\land sp \geq URI$
THEN
act1 : $PM\_Actuator\_V := ON$
act2 : $last\_sp := sp$
END

EVENT Sensor_ON_V
WHEN
grd1 : $PM\_Sensor\_V = OFF$
grd2 : $(sp \geq VRP \land sp < Pace\_Int - FixedAV \land$
$PM\_Sensor\_A = ON)$
$\lor$
$(sp \geq Pace\_Int - FixedAV \land$
$AV\_Count\_STATE = TRUE)$
grd3 : $PM\_Actuator\_A = OFF$
THEN
act1 : $PM\_Sensor\_V := ON$
END

In our previous models [18, 23, 24, 26]. of the cardiac pacemaker and of the heart system, we use the tic event to model a clock, separately. However, in the closed-loop model, we use a *single* event tic to specify a common clock for both cardiac pacemaker and heart environment models. The event tic models the clock behavior, where time is progressively increased using the current clock counter sp. It controls the time line of pacing and sensing events. A guard ($grd1$) of this event provides the required conditions to increase the clock counter sp by 1 (ms.).

EVENT tic
WHEN
grd1 : $(sp < VRP)$
$\lor$
$(sp \geq VRP \land sp < Pace\_Int - FixedAV \land$
$PM\_Sensor\_A = ON \land PM\_Sensor\_V = ON$
THEN
act1 : $sp := sp + 1$
END

3.2 Chain of Refinements

So far, we have described our abstract model of the closed-loop model. Each refinement level is used to introduce a new set of functional properties for modeling the normal and abnormal behaviors of the heart and of the pacemaker. Rather than presenting a chain of refinement stages in detail, we give an overview of the remaining refinement stages, sufficient to explain the rationale of each refinement stage in formalizing the system. For more detailed information, see in [23, 24, 18, 26].

Refinement 1: Introducing *threshold* in Cardiac Pacemaker and Impulse Propagation in the Heart System. This refinement step is known as a conduction model, which introduces the impulse propagation in the conduction network of the heart. The impulse propagation originates from the SA node and passes through all the landmark nodes and reaches at the Purkinje fibers of the ventricles. We formalize the conduction model by the introduction of a set of events, which supports piecewise development of the impulse propagation. The electrical impulse passes through several intermediate landmark nodes and finally sinks to the terminal nodes (C, G, H). The conduction model uses the clock counter to model the real-time system to satisfy the required temporal properties for the impulse propagation. A set of new events simulates the desired behavior of the impulse propagation into the heart conduction network, where each new refined event formalizes impulse flow between two landmark nodes; for instance, the electrical impulse moves from SA node (A) to AV node (B).

In the refinement of the closed-loop system, the cardiac pacemaker development introduces sensors behavior for both atrial and ventricular chambers, which models the sensing activities using some standard threshold values. The threshold values are different for both atrial and ventricle chambers. The heart conduction behavior is continuously monitored by the cardiac pacemaker model. The monitored value is compared with the standard threshold value under the required timing intervals to allow or inhibit to pace into the heart chamber for controlling the desired behaviors of the heart.

Refinement 2: Introduction of Hysteresis for Cardiac Pacemaker Model and Perturbation of the Conduction for the Heart Model. This refinement step introduces an abnormal behavior in the closed-loop model through introduction of the blocking activities, and *hysteresis* operating mode in the cardiac pacemaker model. The blocking behavior in the heart network is known as perturbation model, which specifies perturbations in the heart conduction system and helps to discover exact blocks into the heart conduction network. We introduce a set of events through progressive refinement to simulate the desired blocking behavior. The blocking behavior generates troubles into electrical impulse propagation. Different types of heart blocks are presented through the partition of the landmark nodes in the conduction network.

The cardiac pacemaker model uses the refinement to introduce a new feature related to the operating modes. This new feature is known as the *hysteresis* operating mode, which prevents the constant pacing and allows a patient to have his/her own underlying rhythm as much as possible. The *hysteresis* is a programmed feature whereby the pacemaker paces at a faster rate than the sensing rate. This refinement introduces a new event, which allows to set *hysteresis* mode, and the cardiac pacemaker operates according to the desired rate.

Refinement 3: Introduction of Rate Modulation for the Cardiac Pacemaker Model and a Cellular Model for the Heart System. This is the final refinement of the closed-loop system, which introduces the cellular level modeling for the heart system and the rate modulation for the cardiac pacemaker. The final refinement of the heart system provides a simulation model, which introduces the impulse propagation at the cellular level using cellular automata. The electrical impulse propagates at the cells level. A set of constants and mathematical properties is introduced using axioms, and a set of events is used to formalize the desired behaviors of the heart using cellular automata, which are described in [18].

In the final model of the cardiac pacemaker, we describe a rate adapting pacing technique. The rate adapting pacing technique gives freedom to select automatically desired pacing rate according to the physiological needs. Automatic selection of the desired pacing rate helps to increase or to decrease the pacing rate and assists a patient for controlling the heart rate according to the different day to day activities. In the rate modulation mode, the pacemaker operates faster than the lower rate, but no more than the upper sensor rate limit, when it determines then the heart rate needs to increase. For instance, when a patient does an exercise, the heart rate cannot increase automatically to fulfill the required pumping rate. The rate modulation sensor is used to determine the maximum exertion performed by the patient. This increased pacing rate refers to the *sensor indicated rate*. Reducing the physical activities helps to progressively decrease the pacing rate down to the lower rate. A set of new refined events models increasing and decreasing pacing rate of the cardiac pacemaker.

3.3 Proof Statistics

Table 2 contains the proof statistics of the development of the closed-loop model of the cardiac pacemaker with the heart system. These statistics measure the size of the model, the proof obligations (POs) generated and discharged by the RODIN prover and those that are interactively proved. The complete development of the closed-loop model model results in 3049 (100%) POs, within which 2147 (70%) are proved automatically by the RODIN tool.

Table 2. Proof Statistics

Model	Total number of POs	Automatic Proof	Interactive Proof
Closed-loop model of One-electrode pacemaker			
Abstract Model	304	258(85%)	46(15%)
First Refinement	1015	730(72%)	285(28%)
Second Refinement	72	8(11%)	64(89%)
Third Refinement	153	79(52%)	74(48%)
Closed-loop model of Two-electrode pacemaker			
Abstract Model	291	244(84%)	47(16%)
First Refinement	1039	766(74%)	273(26%)
Second Refinement	53	2(4%)	51(96%)
Third Refinement	122	60(49%)	62(51%)
Total	3049	2147(70%)	902(30%)

The remaining 902 (30%) POs are proved interactively using the RODIN tool. Integration of the heart model and the cardiac pacemaker model generates lots of extra POs. The main reason of these new POs is to use shared variables in both models to link between the heart and pacemaker models. A set of invariants corresponding to the shared variables generates new POs. For example, the current clock counter variable (*sp*) is shared, which has been used in events of the heart and pacemaker models. The combined invariants of the heart and pacemaker models generates new POs corresponding to the current clock counter variable (*sp*). The whole system represents functional properties of the cardiac pacemaker operating modes under the biological environment in the heart. The heart model represents normal and abnormal states of the heart, which is

estimated by the physiological analysis. To guarantee the correctness of these functional behaviors, we have established various invariants in the incremental refinements.

Model checking [27] is a complementary technique for validation and verification of a formal specification. The model checker investigates expected system behaviors under the required safety properties and confirms the correctness of the closed-loop system. The use of model checker helps to discover some unexpected behaviors, and assists to verify all the operating modes of the cardiac pacemaker in the heart environment model. A tool ProB [28] is used to animate the closed-loop model and able to prove the absence of errors [26].

4 Closed-Loop Modeling Requirements

This section presents a set of requirements for modeling the closed-loop system in order to guarantee the safety properties [2]. These requirements are useful for verifying the closed-loop system.

4.1 Patient Safety in Closed-Loop

The closed-loop system must meet a set of requirements related to the physiological needs. The heart's state indicates the patient's condition, which presents conditional properties. In the closed-loop system, the heart states are connected to the heart model parameters, which are not affected by pacemaker therapy. The integration of the heart model and pacemaker model allows us to evaluate whether the pacemaker provides an appropriate therapy for any arrhythmias.

4.2 Behavioral Requirements

The closed-loop system exposes several conditions for both normal and abnormal heart functionalities, which are represented through node automata (Fig. 2(b)) using ranges of impulse propagation speed and impulse propagation time. The condition is a boolean value for meaning whether the heart state is true. The cardiac pacemaker presents pacing and sensing activities under specified conditions. Some behavioral requirements are given as follows: 1) Atrial and ventricular paces should not occur during atrial and ventricular refractory period, respectively. This requirement is an important safety property, which is verified in the closed-loop model. Any pacing during the refractory period creates derangements in timing for the atria and ventricles. 2) Intrinsic activities of the atria and ventricles should be sensed by different leads. The intrinsic activities are essential input for the pacemaker. The pacemaker should ensure that the intrinsic activities are sensed accurately. 3) Natural pacing in the atria and ventricles, and artificial pacing and sensing activities of the pacemaker must be coordinated to ensure efficient pumping for maintaining the heart rhythm.

4.3 Clinical Requirements with Closed-Loop

Clinical requirements depend on the patient needs such as normal sinus rhythm, bradycardia, heart block and tachycardia. These requirements are common critical conditions, which can vary between patients because of different physiological needs.

In this paper, the heart model is as abstract as possible to capture all possible scenarios of the heart, which is completely based on the conduction speed and conduction time. Whenever these two parameters change or lie out of the range, then the ECG signal deforms and we cannot obtain the desired ECG signal, which represents an abnormal heart state. Moreover, we have introduced heart blocking behavior using stepwise refinement. Rather then considering any particular behavior of the heart, we have abstractly formalized the heart. For instance, we have not processed any special treatment in our model to capture the retrograde conduction (travel backwards). We have considered the perfect heart condition (see HeartOK, where we have only a forward conduction network). The retrograde conduction results in many different symptoms, primarily those symptoms resulting from the delayed, non-physiologic timing of atrial contraction in relation to ventricular contraction. According to our model, if the retrograde conduction affects the timing cycle or conduction speed, then the heart presents an abnormal state. The normal state of the closed-loop model is presented according to the timing and speed of the conduction requirements. In case of abnormal state of the heart, the cardiac pacemaker paces and senses according to the patient's needs. In this closed-loop system, the cardiac pacemaker can take effect, when the heart presents an abnormal state, which helps to maintain the patient heart rhythm. We have considered heart state (OK or KO) for each cycle. If the cycle has any abnormality, the heart will be in abnormal state and the pacemaker takes over to maintain the heart rhythm. In addition, this closed-loop model helps to identify the pacemaker requirements according to the heart behavior.

5 Discussion

This paper presents an approach for modeling the closed-loop system. The prime objective of this approach is to provide a new modeling technique, which helps to combine the formal models of a critical system and related environment. For example, the cardiac pacemaker operates in the biological heart system. The closed-loop modeling is an effective approach, which guarantees the correctness of the operating behavior of the critical system. Moreover, this approach provides a viable mechanism for obtaining the certification standards for the system development. To build a closed-loop model using both environment and device modeling, is considered as a standard approach for validation, given that designing an environment model is a challenging problem in the real world. Industry has long sought such an approach to validating system models in a biological environment. We have proposed the closed-loop modeling approach, which is based on our previous research related to the cardiac pacemaker [24] and to the heart model [18].

A Virtual Heart Model (VHM) based on Simulink has been developed by Jiang et al. [2], which can be used for testing a pacemaker. However, a major constraint of their approach is that the VHM and pacemaker both use the Simulink, which is not based on any formal technique such as a theorem prover or model checker. Therefore, it is not feasible to integrate their VHM with any formal methods based cardiac pacemaker model in order to build a closed-loop system. A wide range of work related to the formal verification of the pacemaker has been presented [24, 29, 30], but none of these has used the heart environment model for verification purpose. We have proposed

modeling the heart in an abstract way to simulate the desired behavior of the heart system whilst avoiding the complexity, which is based on logico-mathematical theory [18]. Our proposed approach for modeling the closed-loop system of the heart and pacemaker is better than existing modeling approach. The closed-loop model of the heart and pacemaker is developed using a refinement-based approach and has been used to verify the system properties under patient conditions.

6 Conclusion

We present a method for modeling pacemakers within the closed-loop context of a heart model. The heart model is based on logico-mathematical theory and is the first computational model [18] that considers the heart as an electrical conduction system. Given that a cardiac pacemaker interacts with the heart exactly at this level (i.e., electrical impulses), this model is a very promising *environmental model* to be used in parallel with a pacemaker model to form a closed-loop system. It therefore has an immediate use in *the grand challenges in formal methods* where an industrial pacemaker specification has been elected as a benchmark. To model the closed-loop system of the heart and cardiac pacemaker, we have used the Event-B modeling language[31, 15]. Our approach involves formalizing and reasoning about behavior of a cardiac pacemaker under normal and abnormal heart conditions. A set of general and patient condition-specific temporal requirements is specified for the closed-loop system. Based on these requirements, we have presented an interactive and physiologically relevant closed-loop model for verifying basic and complex operations of the cardiac pacemaker. With the use of model checkers, we demonstrate that the proposed system is capable of testing common and complex heart conditions across a variety of pacemaker modes. This system is a step towards a modeling approach for medical cyber-physical systems with the patient-in-the-loop. The main objectives of the proposed idea are as follows:

- To meet the certification standards
- To verify a critical system like a cardiac pacemaker or implantable cardioverter-defibrillators in a patient model (using a formal representation)
- To analyse the interaction between the heart model and a cardiac pacemaker or implantable cardioverter-defibrillators.

Applying the closed-loop approach for developing the cardiac pacemaker has many benefits, including the exposure of errors which might have not been detected without the environment model. A list of guidlines proposed by regulatory standards (NITRD, IEEE, and IEC/ISO) allows adoption of the closed-loop modeling using formal techniques to establish mechanisms for verifying the specification against the user requirements and certification standards, and to ensure that designs and programs satisfy their requirements specifications.

We have outlined how an incremental refinement approach to the closed-loop model of the heart and pacemaker system enables a high degree of automatic proof using the RODIN tool. Our various developments reflect not only many facets of the problem, but also the learning process involved in understanding the problem and its ultimate possible solutions. The consistency of our specification has been checked through reasoning,

and validation experiments were performed using the ProB model checker with respect to safety conditions. At each stage of the refinement, we have introduced a new behavior for the system and proved its consistency and performed refinement checking. We have introduced more general invariants at the refinement level, showing that the initialization of the whole system is valid. Finally, we have verified the correctness of the exact behavior of our closed-loop system with the help of physiology and cardiology experts.

As a part of our future efforts we plan to generate the automatic test cases from this closed-loop model, permitting system testing. In addition, it would be beneficial to consider a more complex pacemaker model such as the three electrodes pacemaker. Finally, as future work we plan to implement the developed closed-loop formal model. With this approach, our goal is to generate this closed-loop model, moving from a formal model to a Simulink model, which is the most common approach for realizing a real-time system. The final implemented system will comply with developed closed-loop formal models.

Acknowledgement. We are grateful to cardiologist experts Prof. Yves Juillière (MD, Cardiology) and Dr. Frédérique Claudot (PhD) and biomedical experts Dr. Didier Fass (PhD) of the Université de Lorraine, who shared their experience with us. We are thankful to the anonymous reviewers for their helpful and detailed comments.

References

1. Sandler, K., Ohrstrom, L., Moy, L., McVay, R.: Killed by code: Software transparency in implantable medical devices (2010)
2. Jiang, Z., Pajic, M., Mangharam, R.: Model-based closed-loop testing of implantable pacemakers. In: 2011 IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS), pp. 131–140 (April 2011)
3. Maisel, W.H., Sweeney, M.O., Stevenson, W.G., Ellison, K.E., Epstein, L.M.: Recalls and safety alerts involving pacemakers and implantable cardioverter-defibrillator generators. JAMA: The Journal of the American Medical Association 286(7), 793–799 (2001)
4. US FDA Center for Devices and Radiological Health: Medical devices; current good manufacturing practice (cgmp) final rule; quality system regulation (1996)
5. US FDA Center for Devices and Radiological Health: Guidance for the content of premarket submissions for software contained in medical devices (May 2005)
6. Center for Devices and Radiological Health: Safety of Marketed Med. Devices, FDA (2006)
7. A Reseach and Development Needs Report by NITRD: High-Confidence Medical Devices : Cyber-Physical Systems for 21st Century Health Care, `http://www.nitrd.gov/About/MedDevice-FINAL1-web.pdf`
8. Keatley, K.L.: A review of the fda draft guidance document for software validation: guidance for industry. Qual. Assur. 7(1), 49–55 (1999)
9. Lee, I., Pappas, G.J., Cleaveland, R., Hatcliff, J., Krogh, B.H., Lee, P., Rubin, H., Sha, L.: High-confidence medical device software and systems. Computer 39(4), 33–38 (2006)
10. Méry, D., Singh, N.K.: Trustable formal specification for software certification. In: Margaria, T., Steffen, B. (eds.) ISoLA 2010, Part II. LNCS, vol. 6416, pp. 312–326. Springer, Heidelberg (2010)
11. Bowen, J., Stavridou, V.: Safety-critical systems, formal methods and standards. Software Engineering Journal 8(4), 189–209 (1993)
12. Jetley, R.P., Carlos, C., Iyer, S.P.: A case study on applying formal methods to medical devices: computer-aided resuscitation algorithm. International Journal on Software Tools for Technology Transfer 5(4), 320–330 (2004)

13. Jetley, R., Purushothaman Iyer, S., Jones, P.: A formal methods approach to medical device review. Computer 39(4), 61–67 (2006)
14. Méry, D., Singh, N.K.: Real-time animation for formal specification. In: Aiguier, M., Bretaudeau, F., Krob, D. (eds.) Complex Systems Design & Management, pp. 49–60. Springer, Heidelberg (2010)
15. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2010)
16. Fitzgerald, J.S.: The typed logic of partial functions and the vienna development method. In: Bjørner, D., Henson, M.C. (eds.) Logics of Specification Languages. Monographs in Theoretical Computer Science. An EATCS Series, pp. 453–487. Springer, Heidelberg (2008)
17. Lieber, R., Fass, D.: Human systems integration design: Which generalized rationale? In: Kurosu, M. (ed.) Human Centered Design, HCII 2011. LNCS, vol. 6776, pp. 101–109. Springer, Heidelberg (2011)
18. Méry, D., Singh, N.K.: Formalization of heart models based on the conduction of electrical impulses and cellular automata. In: Liu, Z., Wassyng, A. (eds.) FHIES 2011. LNCS, vol. 7151, pp. 140–159. Springer, Heidelberg (2012)
19. Malmivuo, J., Plonsey, R.: Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields, 1st edn. Oxford University Press, USA (1995) ISBN 0-19-505823-2
20. Khan, M.G.: Rapid ECG Interpretation. Humana Press (2008)
21. Bayes de Luna, A., Batcharov, V.N., Malik, M.: The morphology of the Electrocardiogram. In: John Camm, A., Lascher, T.F., Serruys, P.W. (eds.) The ESC Textbook of Cardiovascular Medicine. Blackwell Publishing Ltd. (2006)
22. Artigou, J.Y., Monsuez, J.J., Société française de cardiologie: Cardiologie et maladies vasculaires. Elsevier Masson (2006)
23. Méry, D., Singh, N.K.: Technical Report on Formalisation of the Heart using Analysis of Conduction Time and Velocity of the Electrocardiography and Cellular-Automata. Technical report, LORIA UMR7503 - Université de Lorraine (May 2011)
24. Méry, D., Singh, N.K.: Functional behavior of a cardiac pacing system. International Journal of Discrete Event Control Systems 1(2), 129–149 (2011)
25. Boston Scientific: Pacemaker system specification, Technical report (2007), http://www.cas.mcmaster.ca/sqrl/SQRLDocuments/PACEMAKER.pdf
26. Singh, N.K.: Reliability and Safety of Critical Device Software Systems. PhD in Computer Science, Université Henri Poincaré - Nancy 1, France (November 2011), http://www.scd.uhp-nancy.fr/docnum/SCD_T_2011_0129_SINGH.pdf
27. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press (1999)
28. Leuschel, M., Butler, M.: Prob: A model checker for B. In: Araki, K., Gnesi, S., Mandrioli, D. (eds.) FME 2003. LNCS, vol. 2805, pp. 855–874. Springer, Heidelberg (2003)
29. Macedo, H.D., Larsen, P.G., Fitzgerald, J.: Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System Using VDM. In: Cuellar, J., Maibaum, T. (eds.) FM 2008. LNCS, vol. 5014, pp. 181–197. Springer, Heidelberg (2008)
30. Gomes, A.O., Oliveira, M.V.M.: Formal specification of a cardiac pacing system. In: Cavalcanti, A., Dams, D.R. (eds.) FM 2009. LNCS, vol. 5850, pp. 692–707. Springer, Heidelberg (2009)
31. Project RODIN: Rigorous open development environment for complex systems (2004), http://rodin-b-sharp.sourceforge.net/