Technical Report on Formalisation of the Heart using Analysis of Conduction Time and Velocity of the Electrocardiography and Cellular-Automata

Dominique Méry, Neeraj Kumar Singh

Technical Report on Formalisation of the Heart using Analysis of Conduction Time and Velocity of the Electrocardiography and Cellular-Automata

Dominique Méry and Neeraj Kumar Singh

Université Henri Poincaré Nancy 1
LORIA, BP 239, 54506 Vandoeuvre lès Nancy, France
{mery,singhnne}@loria.fr

Abstract. Formal methods based tools and techniques have been recognised to be a promising approach to support the process of verification and validation of a critical system in early stage of the development. Specially, medical devices are very prone to show an unexpected behavior of the system in operating due to stochastic nature of the system and when a system uses traditional methods for system testing. Device-related problems are responsible for a large number of serious injuries. FDA officials has found that many deaths and injuries related to the devices are caused by product design and engineering flaws. Cardiac pacemaker and implantable cardioverter-defibrillators (ICDs) are main critical medical devices, which require close-loop modeling (integration of system and environment modeling) for verification purpose to obtain a certificate from certification bodies. No any technique is available to provide an environment modeling to verify the developed system model. This report presents a methodology to model a biological system, like heart, for modeling a biological environment. The heart model is mainly based on electrocardiography analysis, which models the heart system at cellular level. Main objective of this methodology is to model the heart system and integrate with medical device model like cardiac pacemaker to specify a close-loop model. Close-loop model of an environment and a device is an open problem in real world. Industries are striving for such kind of approach from long time to validate a system model under a virtual biological environment. Our approach involves the pragmatic combination of formal specification of a system and a biological environment to model a close-loop system to verify the correctness of a system and helps in quality improvement of the system.

Key words: Heart Model, ECG, Cellular Automata, EVENT B, Proof-based development, Refinement

1 Introduction

The human heart is well known as a mechanical device of amazing efficiency to pump blood to the circulatory system continuously throughout a lifetime. It is one of the most complex and an important biological system, which provides oxygen and nutrient to the body for sustaining life [1]. The regular impulses generated by the heart results in rhythm contractions through sequence of muscle of the heart, begins at the natural

pacemaker known as sinoatrial node (SA node), which produces the action potential for traveling across the atrioventricular (AV) node, bundle of His and Purkinje fibers throughout the ventricles. The pattern and the timing of these impulses determine the heart rhythm. Changing time intervals and conduction speeds during heart beat generates abnormal heart rhythm, which is also known as a heart rhythm impairment. A heart rhythm impairment is a main source of several diseases [2]. Using electrocardiography is a common method to diagnose related to the heart diseases [2]. Electrocardiography presents timing properties of an electrical system of the heart, which are most fundamental properties of the heart system.

Cardiac pacemaker and implantable cardioverter-defibrillators (ICDs) are two main remarkable medical and technological devices, which are recommended by doctors in case of abnormal heart rhythm. These devices are used to maintain the heart rhythm and help for life-saving in many instances. From last few years, the use of cardiac pacemaker and cardioverter-defibrillators have increased. Sometime these devices may malfunction. Device-related problems are responsible for a large number of serious injuries. A lot of deaths and injuries have been reported by the US Food and Drug Administrations (FDA) due to such kinds of device failures [3], which advocates safety and security issues for using it. FDA officials has found that many deaths and injuries related to the devices are caused by product design and engineering flaws, which are considered as the firmware problems [4, 5].

Formal-methods based tools and techniques are considered as a de-facto standard for developing the highly critical systems like avionic, automotive and medical systems. Since software plays an increasingly important role in medical devices and more generally in healthcare-related activities, regulatory agencies such as the US Food and Drug Administration and certification bodies (FDA's QSR and ISO's 13485) [6, 5, 7] need effective means for ensuring that the developed software-based healthcare system is *safe* and *reliable*. Regulatory agencies, as well as medical devices manufacturers, have been striving for a more rigorous engineering-based review strategy providing this assurance [8].

Providing assurance guarantees for medical devices makes formal approaches appealing. Formal model-based methods have been successful in targeted applications[9–12, 8, 7] of medical devices. Over the past decade, there has been considerable progress in the development of formal methods [13, 14] to improve confidence in complex software-based systems. Although formal methods are part of the standard recommendations for developing and certifying medical systems, how to integrate formal methods into the certification process is, in large part, unclear. Especially challenging is how to demonstrate that the end product of software development system, behaves securely.

1.1 Motivation

Most challenging problem is an environment modeling, for instance to validate and verify the correct behavior of the system model requires an interactive formal model (an environment formal model). For example a cardiac pacemaker or cardioverter-defibrillators (ICDs) formal models require a heart model to verify the correctness of the developed system (see Fig. 1). No any tools and techniques are available to provide an environment modeling to verify the developed system model. Medical devices

are tightly coupled with biological environment (i.e the heart), which use actuators and sensors to response the biological environment. Due to a strong relationship between medical device (i.e. pacemaker) and related biological environment (i.e. heart), it is required to model the functioning of the medical device within the biological environment. The environment model is independent to the device model, which helps to create an environment for the medical device for simulating the actual behavior of the device. The medical device model is a dependent model on the biological environment. Whenever any undesired state occurs in the biological environment, the device model must act according to the requirements. Main objective to use formal approach for modeling the medical device and biological environment to verify the correctness of the medical system.

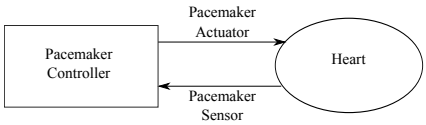


Fig. 1. Cardiac pacemaker and Heart interaction

To model a biological environment (the heart) for a cardiac pacemaker or cardioverter-defibrillators (ICDs), we propose a method for modeling a mathematical heart model based on logico-mathematical theory. The logico-mathematical based heart model is developed using refinement approach in Event-B modeling language [13, 15]. The heart model is based on electrocardiography analysis [16, 2, 17], which models the heart system at cellular level [18]. In this investigation, we present a methodology for modeling a heart model, to extract a set of biological nodes (i.e. SA node, AV node etc.), impulse propagation speed between nodes, impulse propagation time between nodes and cellular automata for propagating impulses at cellular level. This model is developed through incremental refinement, which helps to introduce several properties in an incremental way and to verify the correctness of the heart model. Main key feature of this heart model is representation of all the possible morphological states of the electrocardiogram (ECG) [17, 19]. The morphological states represent the normal and abnormal states of the electrocardiogram (ECG). The morphological representation generates any kind of heart model (patients model or normal heart model using ECG). This model can observe failure of impulse generation and failure of impulse propagation. The mathematical heart model, based on logico-mathematical theory is verified through Rodin [15] proof tool and model checker ProB [20]. This model is also verified through electro-physiologist and cardiac experts. Main objective of this heart model is to provide a biological environment (the heart) for formalizing a closed-loop system (combined model of a cardiac pacemaker and the heart).

1.2 Outline of the Report

The outline of the remaining report is as follows. Section 2 presents the related work. A brief outline of the heart system introduces in Section 3. Section 4 gives an idea of

proposed approach. Section 5 presents a summarized introduction of modeling framework for formalizing any system. Section 6 gives outline of the formal development of the heart model. Section 7 discusses the report with some lessons learned from this experience and Section 8 concludes the report with some perspectives along with future works.

2 Related Work

Heart modeling is a challenging problem in the area of real-time simulation for clinical purpose. Heart modeling problem is handled by the research community using a variety of different methods. Electrocardiogram (ECG) is an important diagnostic method to measure the heart's electrical activities, which was invented by Willem Einthoven in 1903 [21]. Electrocardiogram is used for modeling the heart [21]. At present time, a technological advancement techniques are capable to produce a high quality cellular model of an entire heart model. K.R. Jun et al. [22] have modeled a cellular automata model of activation process in ventricular muscle. They have presented 2-dimensional cellular automata model, which accounts the local orientation of the myocardial fibers and their distributed velocity, and refractory period. A three dimensional finite volume based computer mesh model of human atrial activation and current flow is represented by Harrild et. al [16]. The cellular level based this model includes both the left and right atria and the major muscle bundles of the atria. The results of this model demonstrate a normal sinus rhythm and extract the patterns of septum's activation. Due to memory and time complexity in computation of three dimensional model, an empirical approach is used to model the whole heart. The empirical approach means, it is more simple representation of the complex process at the cellular level. In this new approach, researchers have adopted some approximation to model the whole heart without compromising in the actual behavior of the heart. Berenfeld et al. [23] have developed a model that can give an insight into the local and global complex dynamics of the heart in the transition from normal to abnormal myocardial activity and help to estimate myocardial properties. Adam [24] has analyzed the wave activities during depolarization in his cardiac model, which is represented by simplifying the heart tissue.

Recently, a real time Virtual Heart Model (VHM) has been developed by Jiang et al. [25] to model the electro-physiological operation of the functioning and malfunctioning. They have used time automata model to define the timing properties of the heart. Simulink Design Verifier[1] is used as a main tool for designing the model of the Virtual Heart Model (VHM). As far as we know that Simulink Design Verifier can only check assertions. A comparative case study between a model checker SPIN and Simulink Design Verifier (SLDV) is presented by Leitner et al. [26]. This paper concludes that the Simulink Design Verifier (SLDV) is not able to test deadlock, fairness and liveness properties. However, this Virtual Heart Model (VHM) is not usable with any formal specification of a device like cardiac pacemaker. Our approach is purely based on formal techniques for modeling heart model using electrocardiography analysis. To model the heart for a cardiac pacemaker or cardioverter-defibrillators

[1] http://www.mathworks.com/products/sldesignverifier/

(ICDs), we propose a method for modeling a mathematical heart model based on logico-mathematical theory, which can be implemented in any formal methods based tools (Z, TLA$^+$, VDM etc.). Here, in this report, the model is developed using refinement approach at maximum at cellular level. The incremental refinement approach helps to introduce several properties in an incremental way and to verify the correctness of the the heart model. Main key feature of this heart model is the representation of all the possible morphological states of the electrocardiogram (ECG), which is used to represent the normal and abnormal states through observation of the failure of impulse generation and failure of impulse propagation in the heart [17, 2, 1, 19].

3 Background

3.1 The Heart System

The human heart is wondrous in its ability to pump blood to the circulatory system continuously throughout a lifetime. The heart consists of four chambers: right atria, right ventricle, left atria and left ventricle, which contract and relax periodically. Atria forms one unit and ventricles form another. The heart's mechanical system (the pump) requires at the very least impulses forms the electrical system. An electrical stimulus is generated by the sinus node (see Fig.-2), which is a small mass of specialized tissue located in the right atrium of the heart. The electrical stimulus travels down through the conduction pathways and causes the heart's lower chambers to contract and pump out blood. The right and left atria are stimulated first and contract for a short period of time before the right and left ventricles. Each contraction of the ventricles represents one heartbeat.The atria contracts for a fraction of a second before the ventricles, so their blood empties into the ventricles before the ventricles contract.

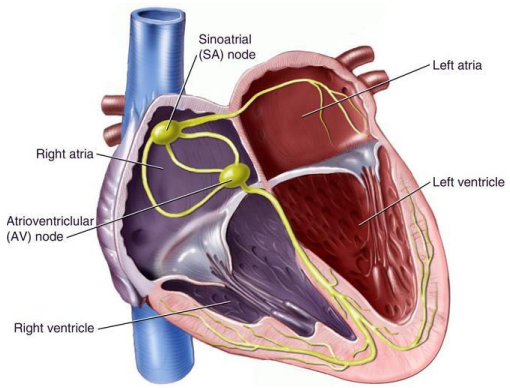


Fig. 2. Heart or Natural Pacemaker [27]

Arrhythmias are due to cardiac problems producing abnormal heart rhythms. In general, arrhythmias reduce haemodynamic performance including situations where

the heart develops an abnormal rate or rhythm or when normal conduction pathways are interrupted and a different part of the heart takes over control of the rhythm. An arrhythmia can involve an abnormal rhythm increase (tachycardia; $>$ 100 bpm) or decrease (bradycardia; $<$ 60 bpm), or may be characterized by an irregular cardiac rhythm, e.g. due to asynchrony of the cardiac chambers. The irregularity of the heartbeat, called bradycardia and tachycardia. The bradycardia indicates that the heart rate falls below the expected level while in tachycardia indicates that the heart rate go above the expected level of the heart rate. An artificial pacemaker can restore synchrony between the atria and ventricles [28–32, 1]. Beats per minute (bpm) is a basic unit to measure the rate of heart activity.

3.2 Basic overview of Electrocardiogram (ECG)

The electrocardiogram (ECG or EKG) [2, 30] is a diagnostic tool that measures and records the electrical activity of the heart precisely in form of signals. Clinicians can evaluate the conditions of a patient's heart from the ECG and perform further diagnosis. Analysis of these signals can be used for interpreting diagnosis of a wide range of heart conditions and predict related diseases. ECG records are obtained by sampling the bioelectric currents sensed by several electrodes, known as leads. A typical one-cycle ECG tracing is shown in Fig.-3. Electrocardiogram term is introduced by Willem Einthoven in 1893 at a meeting of the Dutch Medical Society. In 1924, Einthoven received the Nobel Prize for his life's work in developing the ECG [2, 33, 34, 28, 32, 29, 30, 1].

The normal electrocardiogram (ECG or EKG) is depicted in Fig.-3. All kinds of segments and intervals are represented in this ECG diagram. Depolarization and repolarization of ventricular and atrial chambers are presented by deflection of the ECG signal. All these deflections are denoted by alphabetic order (P-QRS-T). Letter P indicates atrial depolarization and the ventricular depolarization is represented by QRS complex. The ventricular repolarization is represented by T-wave. Atrial repolarization appears during the QRS complex and generates very low amplitude signal which cannot be uncovered from the normal ECG signal.

3.3 ECG Morphology

Sequential activation, depolarization, and repolarization are deflected distinctly in ECG due to anatomical difference of the atria and the ventricles. Even all sequences are easily distinguishable when they are not in correct sequence: P-QRS-T. Each beat of the heart can be observed as a series of deflections, which reflect the time evolution of electrical activity in the heart [17, 2, 19]. A single cycle of the ECG is considered as one heart beat. The ECG may be divided into the following sections:

- **P-wave:** It is a small low-voltage deflection caused by the depolarisation of the atria prior to atrial contraction as the activation (depolarisation) wave-front propagates from the SA node through the atria.
- **PQ-interval:** the time between the beginning of atrial depolarisation and the beginning of ventricular depolarisation.

- **QRS-complex:** QRS-complex are easily identifiable between P- and T-wave because it has characteristic waveform and dominating amplitude. The dominating amplitude is caused by currents generated when the ventricles depolarise prior to their contraction. Although atrial repolarisation occurs before ventricular depolarisation, the latter waveform (i.e. the QRS-complex) is of much greater amplitude and atrial repolarisation is therefore not seen on the ECG.
- **QT-interval:** the time between the onset of ventricular depolarisation and the end of ventricular repolarisation. Clinical studies have demonstrated that the QT-interval increases linearly as the RR-interval increases [4]. Prolonged QT-interval may be associated with delayed ventricular repolarisation which may cause ventricular tachyarrhythmias leading to sudden cardiac death [9].
- **ST-interval:** the time between the end of S-wave and the beginning of T-wave. Significantly elevated or depressed amplitudes away from the baseline are often associated with cardiac illness.
- **T-wave:** ventricular repolarisation, whereby the cardiac muscle is prepared for the next cycle of the ECG.

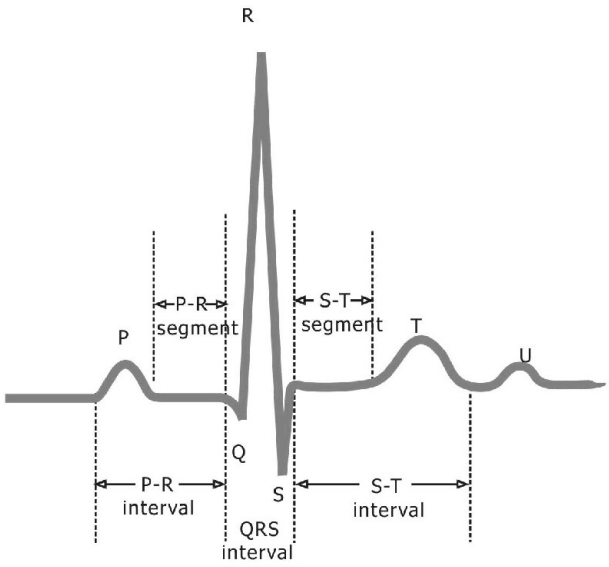
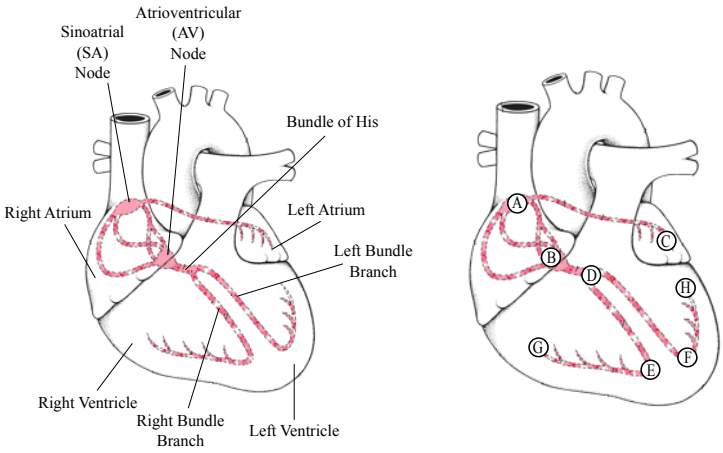


Fig. 3. A typical one-cycle ECG tracing [1]

4 Proposed Idea

Our proposed method exploits the heart model based on logico-mathematics to help the formal community to verify the correctness of developed model of any medical device

like cardiac pacemaker. The heart model is mainly based on impulse propagation time and conduction speed at cellular level. This method uses advance capabilities of the combined approach of formal verification and model validation using a model-checker in order to achieve considerable advantages for the heart system modeling. Fig. 4(a) shows the main important components and impulse conduction path in the entire heart system. The heart is a muscle with a special electrical conduction system. The system is made of two nodes (special conduction cells) and a series of conduction fibers or bundles (pathways). For modeling the heart system, we have assumed eight landmark nodes (A,B,C,D,E,F,G,H) in whole conduction network as shown in Fig. 4(b), which can control the whole heart system. We have discovered all these landmarks through literature survey [1, 2] and a long discussion with cardiologist and physiologist.



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

Fig. 4. The Electrical Conduction and Landmarks of the Heart System

Below we introduce the necessary elements to formally define our heart systems.

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:*

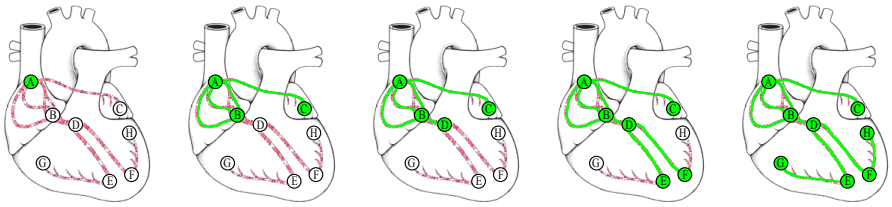
• $N = \{ A, B, C, D, E, F, G, H \}$ *is a finite set of landmark nodes in the conduction pathways of the heart system;*
• $T \subseteq N \times N = \{A \mapsto B, A \mapsto C, B \mapsto D, D \mapsto E, D \mapsto F, E \mapsto G, F \mapsto H\}$ *is a set of transitions to represent electrical impulse propagation between two landmark nodes;*
• $N_0 = A$ *is the initial landmark node (SA node);*
• $TW_{time} \in N \rightarrow TIME$ *is a weight function as time delay of each node, where* $TIME$ *is time delay in range;*
• $CW_{speed} \in T \rightarrow SPEED$ *is a weight function as impulse propagation speed of each*

transition, where $SPEED$ is propagation speed in range.

Property 1 (Impulse Propagation Time). *In the heart system, electrical impulse originates from SA node (node A) and then travels through entire conduction network and terminates to the atrial muscle fibers (node C) and at the end of Purkinje fibers into both side of the ventricular chambers (node G and node H). Impulse propagation times delay differ for each landmark nodes (N). The Impulse propagation time is represented as 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). *Similar to the impulse propagation time, the impulse propagation speed also differs for each transition ($i \rightsquigarrow j$, where $i, j \in N$). The Impulse propagation speed is represented as 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 sinoatrial (SA) node, located high in the right atrium, which is represented by the node A in Fig. 5(a). The sinoatrial (SA) node is the physiological pacemaker of the normal heart, responsible for setting the rate and rhythm. The electrical impulse spreads through the walls of the atria, causing them to contract. The conduction of the electrical impulse throughout the left and right atria is seen on the ECG as the P wave (see Fig. 3). From the sinus node, electrical impulse propagates throughout the atria and reach to the nodes B and C, but cannot propagate directly across the boundary between atria and ventricles. The electrical impulse travels outward into atrial muscle fibers and reached at the end of muscle fibers, which is represented by the node C in the conduction network (see Fig. 5(b)).



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

Fig. 5. Impulse Propagation through Landmark nodes

Normally, the only pathway available for electrical impulse to enter the ventricles is through a specialized region of cells called atrioventricular (AV) node. The atrioventricular node (AV node) is located at the boundary between the atria and ventricles, which

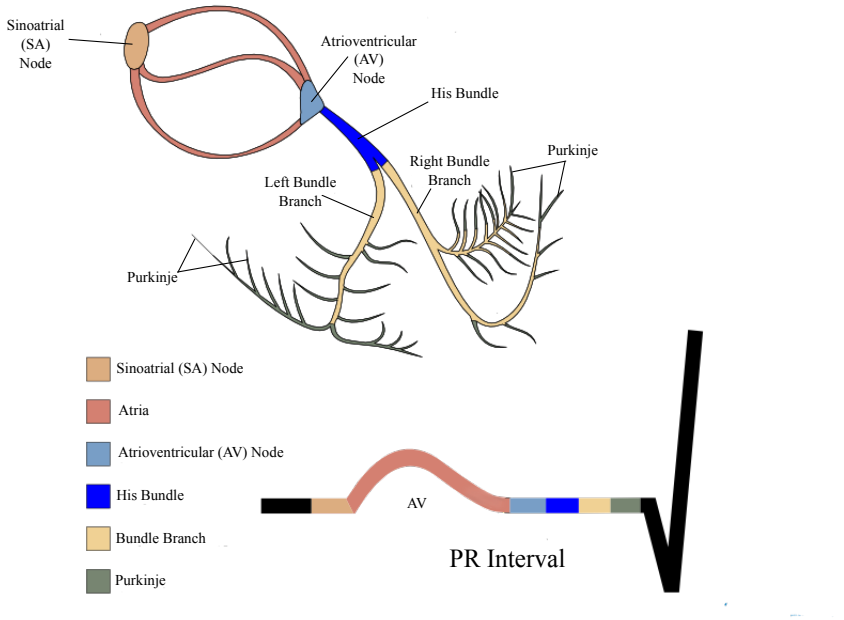


Fig. 6. Time Intervals and Impulse Propagation in the ECG signal [1]

is represented by the node B (see Fig. 4(b)). The AV node provides only conducting path from the atria to the ventricles. The AV node functions as a critical delay in the conduction system. Without this delay, the atria and ventricles would contract at the same time, and blood wouldn't flow effectively from the atria to ventricles. The delay in the AV node forms much of the PR segment on the ECG. Part of atrial repolarization can be represented by the PR segment (see Fig. 3).

Propagation from the atrioventricular (AV) node (A) to the ventricles is provided by a specialized conduction system. The distal portion of AV node is composed of a common bundle, called the bundle of His is denoted as a landmark node D (see Fig. 4(b)). The bundle of His splits into two branches in the inter-ventricular septum, the left bundle branch and the right bundle branch. The electrical impulses then enter the base of the ventricle at the Bundle of His (node D) and then follow the left and right bundle branches along the inter-ventricular septum (see Fig. 5(c)).

Two separate bundle branches propagating along each side of the septum, constituting the right and left bundle branches. We have assumed two landmark nodes E and F (see Fig. 4(b)) at the downside of the heart into both left and right bundle branches. These specialized fibers conduct the impulses at a very rapid velocity (see Table 1). The left bundle branch activates the left ventricle, while the right bundle branch activates the right ventricle (see Fig. 5(d)).

The bundle branches then divide into an extensive system of Purkinje fibers that conduct the impulses at high velocity (see Table 1) throughout the ventricles. The Purkinje fibers, stimulate individual groups of myocardial cells to contract. We have assumed

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		

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

finally two landmark nodes G and H (see Fig. 4(b)) at the end of the Purkinje fibers into both side of the ventricles. These two nodes represent end of the conduction network in the heart system. The bundles ramify into Purkinje fibers that diverge to the inner sides of the ventricular walls (see Fig. 5(e)). Upon reaching at the end of the Purkinje fibers, the electrical impulse is transmitted through the ventricular muscle mass by the ventricular muscle fibers themselves. Propagation along the conduction system takes place at a relatively high speed once it is within the ventricular region, but prior to this (through the AV node) the velocity is extremely slow [1, 2].

The electrical system provides a synchronised system between atria and ventricle, which helps in contraction of the heart muscle and optimizes haemodynamic. Changing time intervals or conducting speeds among landmarks (see Fig. 4(b) and Fig. 6) is a major cause of generating abnormalities in the heart system. Abnormalities due to electrical signal into heart can generate different kinds of arrhythmias. Slow conduction speed generates bradycardia and fast conduction speed generates tachycardia. In this model, we have taken all possible sets of range values of conduction speed and conduction time for each landmark node and conduction path. This model represents morphological structure of the ECG signal through conduction network (see Fig. 6).

4.1 Heart Block

In this section, we have considered to explain the basic heart blocks into the heart conduction system. We have formalised the the basic heart blocks in proposed methodology. Heart block is the term given to a disorder of conduction of the impulse which stimulates heart muscle contraction. The normal cardiac impulse arises in the sinoatrial (SA) node (A) situated in the right atrium and spreads to the atrioventricular (AV) node (B), whence it is conducted by specialised tissue known as the bundle of His (D) which divides into left and right bundle branches into ventricles (see Fig. 4(a)). Disturbance into conduction may demonstrate as slow conduction, intermittent condition failure, or complete conduction failure. All these kinds of conduction failures are also known as 1st, 2nd and 3rd degree blocks. We have shown different kinds of heart blocks throughout conduction network using a set of landmark nodes (see Fig. 7).

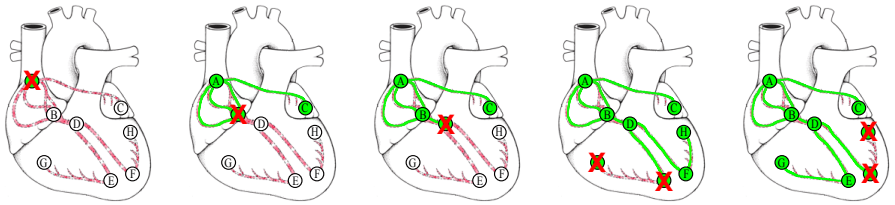
SA block: This block occur within the sinoatrial (SA) node (A) are described as sinoatrial (SA) nodal blocks, which is also known as sick sinus syndrome. Sinoatrial (SA) node is failed for impulse originating at SA nodes and heart misses one or two beats at regular or irregular intervals (see Fig. 7(a)).

AV block: In situation of atrioventricular (AV) block the sinus rhythm is normal, but there is a conduction defect between the atria and the ventricles. Main reason of this block may be in the AV node (B) or bundle of His (D), or both (B, D) (see Fig. 7(b)).

Infra-Hisian block: Blocks that occur below the atrioventricular (AV) node (B) are known as Infra-Hisian blocks (see Fig. 7(c)).

Left bundle branch block: In the normal heart, activation of both ventricles takes place simultaneously. Left bundle branch block occurs when conduction is interrupted into left branch of the bundle of His. Blocks that occur within the fascicles of the left bundle branch are known as hemiblocks (see Fig. 7(d)).

Right bundle branch block: Right bundle branch block occurs when conduction is interrupted into right branch of the bundle of His (see Fig. 7(e)).



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

Fig. 7. Impairments in Impulse Propagation due to the Heart Blocks

4.2 Cellular Automata Model

A set of spatially distributed cells form a cellular automata (CA) model, which contains uniform connection pattern among neighbouring cells and local computation laws. Cellular automata (CA) is originally proposed by Ulam and von Neumann in the 1940s to provide a formal framework for investigating the behaviour of complex, spatially distributed systems [18]. Cellular Automata is a discrete dynamic system corresponding to the space and time. The cellular autamata modeling is the uniform property in state transitions and the interconnection patterns. The model component is specified by a

single property due to same patterns instead of specifying each component separately. Cellular autamata model helps to visualize the system's dynamics [35, 36, 16, 1].

A cellular automata model can have an infinite number of cells in any dimension. Here, we consider a finite number of cells in two dimension as shown in Fig. 8. A two-dimensional cellular automata model is defined as:

Definition 2 (The Cellular Automata Model).
Cellular Automata $(CA) = < S, N, T >$ *: Discrete Time System*
S : the set of states
N: the neighbouring patterns at (0,0),
T: the transition function
In the usual case of Cellular Automata (CA) realized on a D-dimensional grid, N consists of D-tuples of indices from a coordinate set:
I: $N \subseteq I^D$*,*
Hence the cellular model for 2D becomes,
$N \subseteq I^2$.
$T : S^{|N|} \rightarrow S$
To consider automaton specified by the cellular automata (CA), let λ *and* α *be a global state and the global transition function of the cellular autamata (CA), respectively. Then* $\lambda = \{\tau | \tau : I^2 \rightarrow S\}$ *and* $\alpha(\lambda(i,j)) = T(\tau | N + (i,j))$ *for all* τ *in* λ *and (i,j) in* I^2.

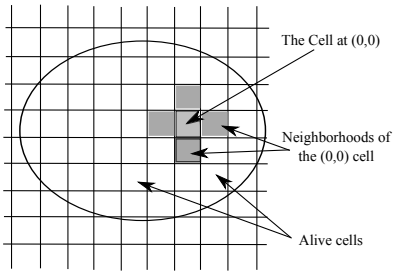


Fig. 8. A two-Dimensional Cellular Automata Model

Definition 3 (State Transition of a Cell). *The heart muscle system is composed of heterogeneous cells, the cellular automata model of the muscle system,* CAM_{CA}*, is characterized with no dependencies on the type of cells.* CAM_{CA} *is defined as follows:*
$CAM_{CA} = < S, N, T >$
$S = \{Active, Passive, Refractory\}$
$N = \{(0,0), (1,0), (-1,0), (0,1), (0,-1)\}$
$s'_{m,n} = s_{m,n}(t+1)$
$s'_{m,n} = T(s_{m,n}, s_{m+1,n}, s_{m-1,n}, s_{m,n+1}, s_{m,n-1})$
where, $s_{m,n}$ *denotes the state of the cell located at (m,n) and T is a transition function of cellular automata (*CAM_{CA}*), which is a function for the next state to be defined in Fig. 9.*

Each cell in the heart muscle should have one of the states: $Active$, $Passive$ or $Refractory$. Initially, all cells have $Passive$ state. In this state, a cell is discharged

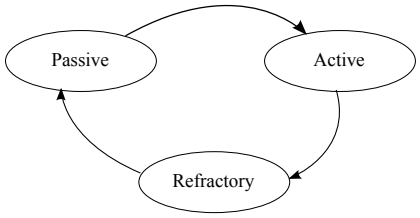


Fig. 9. State Transition of a Cell

electrically and has no influences on its neighbouring cells. When electrical impulse propagate, then the cell would be charged and eventually activated ($Active$ state). Now, the cell transmits the electrical impulse to its neighbour cells. The electrical impulse is propagated to all cells in the heart muscle. After an activation of the cell would be discharged and enter into the $Refractory$ state in which the cell cannot be reactivated After a moment, the cell changes its state to the $Passive$ state, in which the cell awaits next impulse.

5 A Summarized Introduction of The Modeling Framework

We summarize the concepts of the Event-B modeling language developed by Abrial [37, 13] and indicate the links with the tool called RODIN [15]. The modeling process deals with various languages, as seen by considering the *triptych*[2] of Bjoerner [38, 39]: $\mathcal{D}, \mathcal{S} \longrightarrow \mathcal{R}$. Here, the domain $\mathcal{D}$ deals with properties, axioms, sets, constants, functions, relations, and theories. The system model $\mathcal{S}$ expresses a model or a refinement-based chain of models of the system. Finally, $\mathcal{R}$ expresses requirements for the system to be designed. Considering the Event-B modeling language, we notice that the language can express *safety* properties, which are either *invariants* or *theorems* in a machine corresponding to the system. Recall that two main structures are available in Event-B:

- Contexts express static informations about the model.
- Machines express dynamic informations about the model, invariants, safety properties, and events.

Event-B model is defined either as a context or as a machine. The triptych of Bjoerner [38, 39] $\mathcal{D}, \mathcal{S} \longrightarrow \mathcal{R}$ is translated as follows: $\mathcal{C}, \mathcal{M} \longrightarrow \mathcal{R}$, where $\mathcal{C}$ is a context, $\mathcal{M}$ is a machine and $\mathcal{R}$ are the requirements. The relation $\longrightarrow$ is defined to be a logical satisfaction relation with respect to an underlying logico-mathematical theory. The satisfaction relation is supported by the RODIN platform. A machine is organizing events modifying state variables and it uses static informations defined in a context. These basic structure mechanisms are extended by the refinement mechanism which provides a mechanism for relating an abstract model and a concrete model by adding new events

[2] The term 'triptych' covers the three phases of software development: domain description, requirements prescription and software design.

or by adding new variables. This mechanism allows us to develop gradually Event-B models and to validate each decision step using the proof tool. The refinement relationship should be expressed as follows: a model M is refined by a model P, when P is simulating M. The final concrete model is close to the behavior of real system that is executing events using real source code. We give details now on the definition of events, refinement and guidelines for developing complex system models.

5.1 Modelling Actions Over States

The event-driven approach [37, 13] is based on the B notation. It extends the methodological scope of basic concepts to take into account the idea of *formal reactive models*. Briefly, a formal reactive model is characterized by a (finite) list x of *state variables* possibly modified by a (finite) list of *events*, where an invariant $I(x)$ states properties that must always be satisfied by the variables x and *maintained* by the activation of the events. In the following, we summarize the definitions and principles of formal models and explain how they can be managed by tools [15].

Generalized substitutions are borrowed from the B notation. They provide a means to express changes to state variable values. In its general form, an event has three main parts, namely a list of local parameters, a guard and a relation over values denotes pre values of variables and post values of variables. The most common event representation is (ANY t WHERE $G(t,x)$ THEN $x :\mid (R(x,x',t))$ END). The *before–after* predicate $BA(e)(x,x')$, associated with each event, describes the event as a logical predicate expressing the relationship linking the values of the state variables just before (x) and just after (x') the *execution* of event *e*. The form is semantically equivalent to $\exists\, t\cdot\, (G(t,x)\ \wedge\ R(x,x',t)$.

PROOF OBLIGATIONS

- (INV1) $Init(x)\ \Rightarrow\ I(x)$
- (INV2) $I(x)\ \wedge\ BA(e)(x,x')\ \Rightarrow\ I(x')$
- (FIS) $I(x)\ \wedge\ \mathsf{grd}(e)(x)\ \Rightarrow \exists y.BA(e)(x,y)$

Table-2 Event-B proof obligations

Proof obligations (INV 1 and INV 2) are produced by the RODIN tool [15] from events to state that an invariant condition $I(x)$ is preserved. Their general form follows immediately from the definition of the before–after predicate $BA(e)(x,x')$ of each event e (see Table-2). Note that it follows from the two guarded forms of the events that this obligation is trivially discharged when the guard of the event is false. Whenever this is the case, the event is said to be *disabled*. The proof obligation FIS expresses the feasibility of the event e with respect to the invariant I. By proving feasibility we achieve that $BA(e)(x,y)$ provides an after state whenever $grd(e)(x)$ holds. This means that the guard indeed represents the enabling condition of the event.

The intention of specifying a guard of an event is that the event may always occurs when the guard is true. There is, however, some interaction between guards and nondeterministic assignments, namely $x :\mid BA(e)(x,x')$. The predicate $BA(e)(x,x')$ of an action $x :\mid BA(e)(x,x')$ is not satisfiable or the set S of an action $v :\in S$ is empty. Both cases show violations of the event feasibility proof obligation.

We say that an assignment is feasible if there is an after-state satisfying the corresponding before-after predicate. For each event its feasibility must be proved. Note, that for deterministic assignments the proof of feasibility is trivial. Also note, that feasibility of the initialization of a machine yields the existence of an initial state of the machine. It is not necessary to require an extra initialization.

5.2 Model Refinement

The refinement of a formal model allows us to enrich the model via a *step-by-step* approach and is the foundation of our correct-by-construction approach [40]. Refinement provides a way to strengthen invariants and to add details to a model. It is also used to transform an abstract model to a more concrete version by modifying the state description. This is done by extending the list of state variables (possibly suppressing some of them), by refining each abstract event to a set of possible concrete version, and by adding new events. The abstract (x) and concrete (y) state variables are linked by means of a *gluing invariant* $J(x, y)$. A number of proof obligations ensure that (1) each abstract event is correctly refined by its corresponding concrete version, (2) each new event refines $skip$, (3) no new event takes control for ever, and (4) relative deadlock freedom is preserved. Details of the formulation of these proofs follows.

We suppose that an abstract model AM with variables x and invariant $I(x)$ is refined by a concrete model CM with variables y and gluing invariant $J(x, y)$. Event e is in abstract model AM and event f is in concrete model CM. Event f refines event e. $BA(e)(x, x')$ and $BA(f)(y, y')$ are predicates of events e and f respectively, we have to prove the following statement, corresponding to proof obligation (1):

$$I(x) \ \wedge \ J(x, y) \ \wedge \ BA(f)(y, y') \ \Rightarrow \ \exists x' \cdot (BA(e)(x, x') \ \wedge \ J(x', y'))$$

The new events introduced in a refinement step can be viewed as hidden events not visible to the environment of a system and are thus outside the control of the environment. In Event-B, requiring a new event to refine $skip$ means that the effect of the new event is not observable in the abstract model. Any number of executions of an internal action may occur in between each execution of a visible action. Now, proof obligation (2) states that $BA(f)(y, y')$ must refine $skip$ ($x' = x$), generating the following simple statement to prove (2):

$$I(x) \ \wedge \ J(x, y) \ \wedge \ BA(f)(y, y') \ \Rightarrow \ J(x, y')$$

In refining a model, an existing event can be refined by strengthening the guard and/or the before–after predicate (effectively reducing the degree of nondeterminism), or a new event can be added to refine the skip event. The feasibility condition is crucial to avoiding possible states that have no successor, such as division by zero. Furthermore, this refinement guarantees that the set of traces of the refined model contains (up to stuttering) the traces of the resulting model. The refinement of an event e by an event f means that the event f simulates the event e.

The Event-B modeling language is supported by the RODIN platform [15] and has been introduced in publications [13, 37], where the many case studies and discussions about the language itself and the foundations of the Event-B approach. The language of *generalized substitutions* is very rich, enabling the expression of any relation between states in a set-theoretical context. The expressive power of the language leads to a requirement for help in writing relational specifications, which is why we should provide guidelines for assisting the development of Event-B models.

5.3 Tools Environments for Event-B

The Event-B modeling language is supported by the Atelier B [41] environment and by the RODIN platform [15]. Both environments provide facilities for editing machines, refinements, contexts and projects, for generating proof obligations corresponding to a given property, for proving proof obligations in an automatic or/and interactive process and for animating models. The internal prover is shared by the two environments and there are hints generated by the prover interface for helping the interactive proofs. However, the refinement process of machines should be progressive when adding new elements to a given current model and the goal is to distribute the complexity of proofs through the proof-based refinement. These tools are based on logical and semantical concepts of Event-B models (machines, contexts,refinement) and our methodology for modeling medical devices can be built from them.

5.4 Patterns in Event-B Modeling

Considering design patterns [42, 13], the purpose is to capture structures and to make decisions within a design that are common to similar modeling and analysis tasks. They can be re-applied when undertaking similar tasks in order to reduce the duplication of effort. The design pattern approach helps for reusing existing models in current Event-B projects. This approach allows developers to reuse existing models in a way that preserves the correctness of the models and reduces the proving efforts.

A real-time systems are characterizing by their functions, which can be expressed by analyzing *action-reaction* and *real-time* patterns. Sequences of inputs are recognized, and outputs can be emitted in response within a fixed time interval. We recognize the following two design patterns when modeling real-time system according to the relationship between the action and corresponding reaction.

The time constraint pattern in IEEE 1394 proposed by Rehm [43] is fully based on timed automaton. The timed automaton is a finite state machine that is useful to model components of real-time systems. In a model, timed automata interacts with each other and defines a timed transition system. Besides ordinary action transitions that can represent input, output and internal actions. A timed transition system has time progress transitions. Such time progress transitions result in synchronous progress of all clock variables in the model. The time progress is also an event, so there is no change of the underlying Event-B language. It is only a modeling technique instead of a specialized formal system. The timed variable is ranging in $\mathbb{N}$ $(natural\ numbers)$ but time constraint can be written in terms involving unknown constants or expressions

between different times. Finally, the timed event observations can be constrained by other events which determine future activations.

6 Formalization of the Heart

To formalize the heart model, we have used Event-B modeling language, while the proposed idea can be formalized with any kind of formal method tools like Z, ASM, TLA$^+$ and VDM etcetera. Here we have used Event-B [15, 13] modeling language, which supports refinement approach [44] that helps to verify the correctness of the system in an incremental way.

Heart model development is expressed in an abstract and general way. We describe the incremental development of the heart model in several phases using step-wise refinements. The initial model formalizes the system requirements and environmental assumptions, whereas the subsequent models introduce design decisions for the resulting system. Following summary informations present global view of the heart system development, which help to understand the whole modeling approach.

Initial Model : This is an observation model, which specifies heart state in form of $true$ and $false$, where $true$ represents normal rhythm and $false$ represents abnormal rhythm of the heart.

Refinement 1 : This is a conduction model of the heart, which specifies beginning of the impulse propagation at SA node and ending of the impulse propagation at Purkinje fibers in both left and right ventricles.

Refinement 2 : This model specifies impulse propagation between landmark nodes with global clock counter to model a real-time system to satisfy the temporal properties of impulse propagation.

Refinement 3 : This is perturbation model of the heart, which specifies perturbation in the heart conduction system and helps to discover exact block into the heart conduction system.

Refinement 4 : This is a simulation model of the heart, which introduces impulse propagation at cellular level using cellular automata.

6.1 The Context and Initial Model

Event-B models are described in two major components: *context* and *machine*. Context contains the static part of a model whereas machine contains the dynamic part. The context uses sets and constants to define axioms and theorems. Axioms and theorems represent the logical theory of the elements of the system. The logical theory lists static properties of constants related to the system and provides an axiomatization of system environment. Context can be extended by other context and referenced by a set of machines, while machine can be refined by machines.

We choose the electrical features for modeling the heart system. To model the heart system, we identify a set of electrical impulse propagation nodes *ConductionNode* of the heart conduction network (see Fig. 4(a)). These nodes are basic landmarks, which express the normal and abnormal behavior of the heart system. We have discovered all these landmarks through literature survey [1, 2] and a long discussion with cardiologist

and physiologist. Three constants define impulse propagation time *ConductionTime*, impulse propagation path *ConductionPath*, and impulse propagation velocity *ConductionSpeed*. Static properties are defined in the context model for specifying an electrical impulse propagation network of the heart system, impulse propagation time for each landmark node, and impulse propagation speed for every path. The path is represented by pair of landmark nodes (see Definition 1, Properties 1 and 2, Table 1).

$$
\begin{array}{l}
axm1 : partition(ConductionNode, \{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}, \{G\}, \{H\}) \\
axm2 : ConductionTime \in ConductionNode \rightarrow \mathbb{P}(0 \mathinner{..} 230) \\
axm3 : ConductionPath \subseteq ConductionNode \times ConductionNode \\
axm4 : ConductionSpeed \in ConductionPath \rightarrow \mathbb{P}(5 \mathinner{..} 400)
\end{array}
$$

As you see axioms are extracted from the definitions and are validated by cardiologist and physiologist.

6.2 Abstract Model

We define an abstract model for indicating the heart state according to the observation impulse propagation on the conduction nodes. The machine model represents dynamic behavior of the heart system through step wise impulse propagation into atria and ventricular chambers. To define the dynamic properties, we have introduced four variables *ConductionNodeState*, *CConductionTime*, *CConductionSpeed* and *HeartState* in invariants. The variable *ConductionNodeState* is defined as function, which shows boolean states of the landmark nodes. When the electrical impulse passes through landmark nodes (see Fig. 4(b)), then the visited nodes become *TRUE* and the unvisited landmark nodes are represented by *FALSE*. The variables *CConductionTime* and *CConductionSpeed* represent current impulse propagation time and velocity in the conduction network. The last variable *HeartState* represents boolean states *TRUE* or *FALSE*. *TRUE* represents normal condition of the heart while *FALSE* represents abnormal condition of the heart.

$$
\begin{array}{l}
inv1 : ConductionNodeState \in ConductionNode \rightarrow BOOL \\
inv2 : CConductionTime \in ConductionNode \rightarrow 0 \mathinner{..} 300 \\
inv3 : CConductionSpeed \in ConductionPath \rightarrow 0 \mathinner{..} 500 \\
inv4 : HeartState \in BOOL
\end{array}
$$

In the abstract specification of the heart model, there are three events *HeartOK* to represent normal state of the heart system, *HeartKO* to express abnormal state of the heart system, and *HeartConduction* to update the value of each landmark nodes of the conduction network in terms of visited landmark nodes (*ConductionNodeState*), impulse propagation intervals (*CConductionTime*) and impulse propagation velocities (*CConductionSpeed*).

The event *HeartOK* specifies a set of required conditions for a normal state of the heart system. First guard *grd1* represents that all landmark nodes should be visited in single cycle of impulse propagation; second guard states that current impulse propagation time of each landmark node should lie within the pre-specified range of the impulse

propagation time and the last guard states that current impulse propagation velocity of the each path should be lie in between pre-defined impulse propagation velocity. When all the guards satisfy then the heart state represents normal condition as *TRUE*.

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 : $\forall i, j \cdot i \mapsto j \in ConductionPath \Rightarrow$
$CConductionSpeed(i \mapsto j) \in ConductionSpeed(i \mapsto j)$
THEN
act1 : $HeartState := TRUE$
END

The event *HeartKO* specifies as an opposite set of guards than the normal state of the heart system for specifying abnormal conditions of the heart state. First guard *grd1* represents that if any landmark node is not visited in a single cycle of the impulse propagation; or second guard states that the current impulse propagation time of any landmark node is not lied within the pre-specified range of the impulse propagation time; or the last guard states that the current impulse propagation velocity of any path is not lied in between pre-defined impulse propagation velocity, then the heart system represents an abnormal state as *FALSE*. Different kinds of heart diseases affect the electrical impulse propagation time and velocity in the heart system [2]. It means that heart changes and we model diseases as possible behaviors.

EVENT HeartKO
WHEN
grd1 : $\exists i \cdot i \in ConductionNode \wedge ConductionNodeState(i) = FALSE)$
$\vee$
$(\exists j \cdot j \in ConductionNode \wedge CConductionTime(j) \notin ConductionTime(j))$
$\vee$
$(\exists m, n \cdot m \mapsto n \in ConductionPath \wedge CConductionSpeed(m \mapsto n)$
$\notin ConductionSpeed(m \mapsto n))$
THEN
act1 : $HeartState := FALSE$
END

The event *HeartConduction* abstractly formalises the heart behavior through updating the value of impulse propagation time, impulse propagation velocity and visited state of the landmark nodes in-deterministically. This event is used to model more concrete behavior of the heart system in the next level of refinement.

EVENT HeartConduction
BEGIN
act1 : $ConductionNodeState :\in ConductionNode \rightarrow BOOL$
act2 : $CConductionTime :\in ConductionNode \rightarrow 0 \mathinner{..} 300$
act3 : $CConductionSpeed :\in ConductionPath \rightarrow 0 \mathinner{..} 500$
act4 : $HeartState :\in BOOL$
END

6.3 Refinement 1: Introducing Steps in the Propagation

In the abstract model, we have presented that the impulse propagation time, velocity and visited landmark nodes have been updated in an atomic step when electrical impulse fire from the sinus (SA) node and moves towards the Purkinje fibers into ventricles (G, H nodes) and in the left atria muscle fibers (C node). Our main objective is to model step by step impulse propagation through all landmark nodes, where the electrical impulse must pass through a number of intermediate landmark nodes before reaching to the terminal nodes (C, G, H). This refinement is a very simple refinement, where we introduce two extra events *SinusNodeFire* and *HeartConductionEnd* as the refinement of the event *HeartConduction*. The event *SinusNodeFire* models the behavior of a sinoatrial (SA) node, which originates electrical impulse for traversing throughout the heart system using conduction network (see Fig. 4). Guards of this event state that if all landmark nodes are unvisited (means FALSE state) and current impulse propagation time of each node is 0, and impulse propagation velocity of each path is 0, then the conduction node state *ConductionNodeState* of landmark node A (SA node) sets TRUE and current impulse propagation time of SA node (A) sets to 0.

EVENT SinusNodeFire Refines HeartConduction
WHEN
grd1 : $\forall n \cdot n \in ConductionNode \Rightarrow ConductionNodeState(n) = FALSE$
grd2 : $\forall n \cdot n \in ConductionNode \Rightarrow CConductionTime(n) = 0$
grd3 : $\forall n, m \cdot n \in ConductionNode \land m \in ConductionNode \land$
$n \mapsto m \in ConductionPath \Rightarrow CConductionSpeed(n \mapsto m) = 0$
THEN
act1 : $ConductionNodeState(A) := TRUE$
act2 : $CConductionTime(A) := 0$
END

The next event *HeartConductionEnd* represents end state of the impulse propagation into Purkinje fibers of ventricles (G, H nodes) and left atria muscle (node C). This event resets all variables for generating next impulse at SA node. The actions of the event reset all conduction node state as *FALSE*, current impulse propagation time of all landmark nodes reset to 0, current impulse propagation velocity of all landmark nodes reset to 0, and the heart state set as *FALSE*. All these actions are required before originating the next electrical impulse from the SA node (A).

EVENT HeartConductionEnd Refines HeartConduction
BEGIN
act1 : $ConductionNodeState := \{A \mapsto FALSE, B \mapsto FALSE, C \mapsto FALSE, D \mapsto FALSE, E \mapsto FALSE, F \mapsto FALSE, G \mapsto FALSE, H \mapsto FALSE\}$
act2 : $CConductionTime := \{A \mapsto 0, B \mapsto 0, C \mapsto 0, D \mapsto 0, E \mapsto 0, F \mapsto 0, G \mapsto 0, H \mapsto 0\}$
act3 : $CConductionSpeed := \{A \mapsto B \mapsto 0, A \mapsto C \mapsto 0, B \mapsto D \mapsto 0, D \mapsto E \mapsto 0, D \mapsto F \mapsto 0, E \mapsto G \mapsto 0, F \mapsto H \mapsto 0\}$
act4 : $HeartState := FALSE$
END

6.4 Refinement 2: Impulse Propagation

In the second refinement, we introduce several events as a refinement of the event *Heart-Conduction* to model the impulse propagation into the heart conduction network. New events are formalizing impulse flow between two landmark nodes separately; for instance, electrical impulse moves from SA node (A) to AV node (B). This level of refinement introduces seven events for modeling the whole conduction path from originating nodes (A) to the ending nodes (C, G, H). A variable *CCSpeed_CCTime_Flag* is introduced as a boolean type to capture the value of current impulse propagation time and current impulse propagation velocity. A new variable *Cycle_Length* declares the time interval of the single heart beat, which may change in every cycle of electrocardiogram (ECG). This refinement also introduces a logical clock to synchronise all states of the heart system and checks the heart states under a required time length in the conduction network. A new variable *tic* is defined as current *clock counter*. Invariants (*inv4-inv10*) are introduced as safety properties, which define that if the heart state is *TRUE* then the impulse propagation time and the impulse propagation velocity are always lied within the standard range of time and velocity during the impulse conduction throughout the conduction network (see Fig. 4(b)).

$inv1 : CCSpeed\_CCTime\_Flag \in BOOL$
$inv2 : Cycle\_Length \in 500..2000$
$inv3 : tic \in \mathbb{N}$

$inv4 : HeartState = TRUE \Rightarrow CConductionTime(B) \in ConductionTime(B) \wedge$
$CConductionSpeed(A \mapsto B) \in ConductionSpeed(A \mapsto B)$
$inv5 : HeartState = TRUE \Rightarrow CConductionTime(C) \in ConductionTime(C) \wedge$
$CConductionSpeed(A \mapsto C) \in ConductionSpeed(A \mapsto C)$
$inv6 : HeartState = TRUE \Rightarrow CConductionTime(D) \in ConductionTime(D) \wedge$
$CConductionSpeed(B \mapsto D) \in ConductionSpeed(B \mapsto D)$
$inv7 : HeartState = TRUE \Rightarrow CConductionTime(E) \in ConductionTime(E) \wedge$
$CConductionSpeed(D \mapsto E) \in ConductionSpeed(D \mapsto E)$
$inv8 : HeartState = TRUE \Rightarrow CConductionTime(F) \in ConductionTime(F) \wedge$
$CConductionSpeed(D \mapsto F) \in ConductionSpeed(D \mapsto F)$
$inv9 : HeartState = TRUE \Rightarrow CConductionTime(G) \in ConductionTime(G) \wedge$
$CConductionSpeed(E \mapsto G) \in ConductionSpeed(E \mapsto G)$
$inv10 : HeartState = TRUE \Rightarrow CConductionTime(H) \in ConductionTime(H) \wedge$
$CConductionSpeed(F \mapsto H) \in ConductionSpeed(F \mapsto H)$

Events are introduced in this refinement to model the impulse propagation from SA node towards the Purkinje fibers landmark nodes (G, H) and atria fibers nodes (C). Each event is synchronised through progressive electrical impulse propagation in the conduction network. We have given formalization of only one event *HeartConduction_A_B* to understand the basic formalization step of all other events. All other events of impulse propagation in the conduction network among landmark nodes have been modeled in a similar fashion.

EVENT HeartConduction_A_B Refines HeartConduction
WHEN
grd1 : $ConductionNodeState(A) = TRUE$
grd2 : $ConductionNodeState(B) = FALSE$
grd3 : $CConductionTime(B) \in ConductionTime(B)$
grd4 : $CConductionSpeed(A \mapsto B) \in ConductionSpeed(A \mapsto B)$
grd5 : $CCSpeed\_CCTime\_Flag = FALSE$
THEN
act1 : $ConductionNodeState(B) := TRUE$
act2 : $CCSpeed\_CCTime\_Flag := TRUE$
END

A new event *Update_CCSpeed_CCtime* is a refinement of the event *HeartConduction*. This event is used to capture the current electrical impulse propagation time *CConductionTime* and the current electrical impulse propagation speed *CCconductionSpeed* during a progressive conduction flow into the heart system in the conduction network.

EVENT Update_CCSpeed_CCtime Refines HeartConduction
ANY $i, j, CSpeed, CTime$
WHERE
grd1 : $i \in ConductionNode$
grd2 : $j \in ConductionNode$
grd3 : $i \mapsto j \in ConductionPath$
grd4 : $CSpeed \in 0 \,..\, 500$
grd5 : $CTime \in 0 \,..\, 300$
grd6 : $CCSpeed\_CCTime\_Flag = TRUE$
grd7 : $HeartState = FALSE$
grd8 : $tic = CTime$
THEN
act1 : $CConductionTime(j) := CTime$
act2 : $CConductionSpeed(i \mapsto j) := CSpeed$
act3 : $CCSpeed\_CCTime\_Flag := FALSE$
END

The electrical impulse propagates at every millisecond. But the impulse propagation time and velocity are different for each landmark nodes. The progressive increment of the independent logical clock is model through event *tic*, that increments time in 1 ms. The event *Clock_Counter* progressively increases the current clock counter tic under pre-defined cycle length $Cycle\_Length$. The predicate in guard $(grd1)$ of event *Clock_Counter* represents an upper bound time limit. The current clock counter tic is reset to 0 by the event *HeartConductionEnd*. An extra guard is added in the event *HeartConductionEnd* as $tic = Cycle\_Length$ to reset the all parametric values of the heart system for starting a fresh new impulse propagation cycle.

EVENT Clock_Counter
WHEN
grd1 : $tic < Cycle\_Length$
THEN
act1 : $tic := tic + 1$
END

We have defined the event *Clock_Counter* as a type of $Convergent$ and the system variant is define as $Cycle\_length - tic$, which generates the convergence proof obligations to verify that the time is progressing with the electrical impulse propagation. It means that the electrical impulse is propagating in the conduction network corresponding to the clock counter.

6.5 Refinement 3: Perturbation the Conduction

It introduces a set of possible blocks in the heart conducting system. These blocks can occur into the conduction network and give trouble into electrical impulse propagation. A set of landmark nodes partition the different regions for all possible heart blocks. For introducing the heart blocks, we introduce an enumerated set *HeartBlockSets* in a new context model as a static property of the heart system.

$axm1 : partition(HeartBlockSets, \{SA\_nodal\_blocks\}, \{AV\_nodal\_blocks\},$
$\{Infra\_Hisian\_blocks\}, \{LBBB\_blocks\}, \{RBBB\_blocks\}, \{None\})$

To model the heart block system, we define a variable *HeartBlocks* as $HeartBlocks \in HeartBlockSets$. New events are introduced to show different kinds of heart blocks during impulse propagation into conduction network. Events are *HeartConduction_Block_A_B_C* to formalise the sinoatrial (SA) nodal block, *HeartConduction_Block_B* to represent atrioventricular (AV) nodal block, *HeartConduction_Block_B_D* to specify Infra-Hisian block, *HeartConduction_Block_D_E_G* to present Left bundle branch block, and *HeartConduction_Block_D_F_H* to specify the Right bundle branch block.

Conduction disturbance in the heart during which an impulse formed within the sinus node (A) is blocked or delayed from depolarizing the atria. There are different kinds of SA blocks [1, 2]. To model SA block, we introduce an event *HeartConduction_Block_A_B_C*, which formalises the SA block. In this event, guard ($grd1$) represents that the landmark nodes (A or C) are not visited means FALSE state, or the current impulse propagation time of B and C nodes are not lied within the standard range, or the current impulse propagation velocity of the pairs $A \mapsto B$ and $A \mapsto C$ are not lied within the standard range. When guard is triggered, then actions of this event state that the heart state is FALSE and the heart block is a sinoatrial (SA) nodal block.

EVENT HeartConduction_Block_A_B_C Refines HeartKO
WHEN
grd1 : $(ConductionNodeState(A) = FALSE)$
$\lor$
$(ConductionNodeState(C) = FALSE)$
$\lor$
$(CConductionTime(B) \notin ConductionTime(B))$
$\lor$
$(CConductionTime(C) \notin ConductionTime(C))$
$\lor$
$(CConductionSpeed(A \mapsto B) \notin ConductionSpeed(A \mapsto B))$
$\lor$
$(CConductionSpeed(A \mapsto C) \notin ConductionSpeed(A \mapsto C))$
THEN
act1 : $HeartState := FALSE$
act2 : $HeartBlocks := SA\_nodal\_blocks$
END

Any interruption in the conduction of electrical impulses from the atria to the ventricles; it can occur at the level of the atria, the atrioventricular node, the bundle of His, or the Purkinje system. It is a type of heart block in which the blocking is at the atrioventricular (AV) junction. It is known as first degree when atrioventricular (AV) conduction time is prolonged; it is called second degree or partial when some but not all atrial impulses reach the ventricle; and it is called third degree or complete when no atrial impulses at all reach the ventricle, so that the atria and ventricles act independently of each other. There are different kinds of AV blocks [1, 2]. To model AV block,

we introduce an event *HeartConduction_Block_B*, which formalises the AV block. The conduction node state *ConductionNodeState* of landmark node (B) is *FALSE*, which represents a condition for AV block using guard ($grd1$) and actions state that the heart state is *FALSE* and such kinds of heart block is known as atrioventricular (AV) nodal block.

EVENT HeartConduction_Block_B Refines HeartKO
WHEN
grd1 : $(ConductionNodeState(B) = FALSE)$
THEN
act1 : $HeartState := FALSE$
act2 : $HeartBlocks := AV\_nodal\_blocks$
END

Infra-Hisian block describes block of the distal conduction system (node D). There are different kinds of Infra-Hisian blocks [1, 2]. To model Infra-Hisian block, an event *HeartConduction_Block_B_D* is used to formalise the desired conditions for a such kind of blocks through landmark nodes (B, D). Guard ($grd1$) represents that the landmark node (D) is *FALSE*, means it is not visited, or current impulse propagation time of node D is not lied within the standard range, or current propagation velocity of pair $B \mapsto D$ is not lied within the standard range. Actions of this event state that the heart state is FALSE and the heart block is the Infra-Hisian block.

EVENT HeartConduction_Block_B_D Refines HeartKO
WHEN
grd1 : $(ConductionNodeState(D) = FALSE)$
$\vee$
$(CConductionTime(D) \notin ConductionTime(D))$
$\vee$
$(CConductionSpeed(B \mapsto D) \notin ConductionSpeed(B \mapsto D))$
THEN
act1 : $HeartState := FALSE$
act2 : $HeartBlocks := Infra\_Hisian\_blocks$
END

The bundle of His divides into a right bundle branch and a left bundle branch, which lead to your heart's lower chambers (the ventricles). For the left and right ventricles to contract at the same time, an electrical impulse must travel down the right and left bundle branches at the same speed. If there is a block in one of these branches, the electrical impulse must travel to the ventricle by a different route. When this happens, the rate and rhythm of your heartbeat are not affected, but the impulse is slowed. Even ventricle will still contract, but it will take longer because of the slowed impulse. This slowed impulse causes one ventricle to contract a fraction of a second slower than the other [1, 2]. The medical terms for bundle branch block are derived from which branch is affected. If the block is located in the right bundle branch, it is called Right bundle branch block. If the block is located in the left bundle branch, it is called Left bundle branch block.

To model the Right bundle branch block, we introduce an event in a similar fashion like past events. A new event *HeartConduction_Block_D_E_G* formalises the Right bundle branch; a guard of this event states that the landmark nodes (E or G) are not visited means FALSE state, or the current impulse propagation time of E and G nodes are not lied within the standard ranges, or the current impulse propagation velocity of the pairs $D \mapsto E$ and $E \mapsto G$ are not lied within the standard range; then the actions of this event state that the heart state is FALSE and the heart block is the Right bundle branch block.

EVENT HeartConduction_Block_D_E_G Refines HeartKO
WHEN
grd1 : $(ConductionNodeState(E) = FALSE)$
$\vee$
$(ConductionNodeState(G) = FALSE)$
$\vee$
$(CConductionTime(E) \notin ConductionTime(E))$
$\vee$
$(CConductionTime(C) \notin ConductionTime(C))$
$\vee$
$(CConductionSpeed(D \mapsto E) \notin ConductionSpeed(D \mapsto E))$
$\vee$
$(CConductionSpeed(E \mapsto G) \notin ConductionSpeed(E \mapsto G))$
THEN
act1 : $HeartState := FALSE$
act2 : $HeartBlocks := RBBB\_blocks$
END

To model the Left bundle branch block, we introduce an event like Right bundle branch event. This new event *HeartConduction_Block_D_F_H* formalises the Left bundle branch. A guard of this event states that the landmark nodes (F or H) are not visited means FALSE state, or the current impulse propagation time of F and H nodes are not lied within the standard range, or the current impulse propagation velocity of the pairs $D \mapsto F$ and $F \mapsto H$ are not lied within the standard range. Then the actions of this event state that the heart state is FALSE and the heart block is the Left bundle branch block.

EVENT HeartConduction_Block_D_F_H Refines HeartKO
WHEN
grd1 : $(ConductionNodeState(F) = FALSE)$
$\lor$
$(ConductionNodeState(H) = FALSE)$
$\lor$
$(CConductionTime(F) \notin ConductionTime(F))$
$\lor$
$(CConductionTime(H) \notin ConductionTime(H))$
$\lor$
$(CConductionSpeed(D \mapsto F) \notin ConductionSpeed(D \mapsto F))$
$\lor$
$(CConductionSpeed(F \mapsto H) \notin ConductionSpeed(F \mapsto H))$
THEN
act1 : $HeartState := FALSE$
act2 : $HeartBlocks := LBBB\_blocks$
END

6.6 Refinement 4: Getting a Cellular Model

This last refinement introduces cellular level modeling into the heart model. The cellular level modeling is used to model the electrical impulse propagation at cell level. The formalisation uses cellular autamata theory to model the micro-structure based cell model. To formalise the cellular automata, we introduce mathematical properties (see Definition 2 and 3) in context model. In the biological system, each cell has one of the following states: $Active$, $Passive$ or $Refractory$. To define cell states, we declare an enumerated set *CellStates*. We have assumed grid of cells in square format. Due to square geometry of the cells, we define a constant *NeighbouringCells* to represent a set of coordinated positions of the neighbouring cells. A new function *NEXT* is used to define neighbouring cell's state. This function maps from power-set of *NeighbouringCells* to the cell's state *CellStates*. A new function *CellS* is defined as to map from *NeighbouringCells* to *CellStates*. This function maps various states like $Active$, $Passive$ and $Refractory$ to the neighbouring cells.

$axm1 : partition(CellStates, \{PASSIVE\}, \{ACTIVE\}, \{REFRACTORY\})$
$axm2 : x \in \mathbb{Z}$
$axm3 : y \in \mathbb{Z}$
$axm4 : NeighbouringCells =$
$\{\{x, y\}, \{x + 1, y\}, \{x - 1, y\}, \{x, y + 1\}, \{x, y - 1\}\}$
$axm5 : NEXT \in \mathbb{P}(NeighbouringCells) \rightarrow CellStates$
$axm6 : CellS \in NeighbouringCells \rightarrow CellStates$

A set of properties *(axm7-axm10)* is introduced to specify the desired behavior of the biological cell automata in two-dimensions. All these properties implement the state transition of a cell and formalise the transitions automaton (see Fig. 9). The first property ($axm1$) states that if the neighbouring cells are in $Active$ state then the NEXT

state of the cell must be $Refractory$. The second property ($axm8$) represents that if the neighbouring cells are in $Refractory$ state then the NEXT state of the cell must be $Passive$. Third property ($axm9$) states that if a cell at (x, y) is $Passive$, then if all the neighbouring cells in 2D is $Active$, then a set of neighbouring cells must be in $Active$. Similarly, last property ($axm10$) presents that if a cell at (x, y) is $Passive$, then and if all the neighbouring cells in 2D is $Active$, then a set of neighbouring cells must be in $Active$.

$axm7 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \land CellS(\{x, y\}) = ACTIVE \Rightarrow NEXT(param) = REFRACTORY$

$axm8 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \land CellS(\{x, y\}) = REFRACTORY \Rightarrow NEXT(param) = PASSIVE$

$axm9 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \land \{x, y\} \in paramCellS(\{x, y\}) = PASSIVE \Rightarrow ((CellS(\{x + 1, y\}) = ACTIVE \lor CellS(\{x - 1, y\}) = ACTIVE \lor CellS(\{x, y + 1\}) = ACTIVE \lor CellS(\{x, y - 1\}) = ACTIVE) \Rightarrow NEXT(param) = ACTIVE)$

$axm10 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \land \{x, y\} \in param \land CellS(\{x, y\}) = PASSIVE \Rightarrow ((CellS(\{x + 1, y\}) \neq ACTIVE \land CellS(\{x - 1, y\}) \neq ACTIVE \land CellS(\{x, y + 1\}) \neq ACTIVE \land CellS(\{x, y - 1\}) \neq ACTIVE \Rightarrow NEXT(param) = PASSIVE)$

Each cell in the heart muscle must have one of the states: $Active$, $Passive$ or $Refractory$. Initially, all cells have $Passive$ state. In this state, a cell is discharged electrically and has no influences on its neighbouring cells. When electrical impulse propagates, then the cell would be charged and eventually activated ($Active$ state). Now, the cell transmits the electrical impulse to its neighbour cells. The electrical impulse is propagated to all cells in the heart muscle. After an activation the cell would be discharged and enter into the $Refractory$ state in which the cell cannot be reactivated After a moment, the cell changes its state to the $Passive$ state, in which the cell awaits next impulse (see Fig. 9).

To model the dynamic behavior of the cell automata, we declare four variables m, n, *Transition* and *NextCellState*. Two variables m and n represent current position of the active cell during impulse propagation. The variable *Transition* is defined as boolean to set the transition state $TRUE$ or $FALSE$ to model the behavior of a tissue. Last variable *NextCellState* is used to store the values of next neighbouring positions after every transitions.

$inv1 : m \in \mathbb{Z}$

$inv2 : n \in \mathbb{Z}$

$inv3 : Transition \in BOOL$

$inv4 : NextCellState \in CellStates$

To implement the dynamic behavior of the cell in two-dimensions, we introduce two events *HeartConduction_Cellular* to make transition TRUE for electrical conduction at cell level and *HeartConduction_Next_UpdateCell* to calculate status of neighbouring cells and update the current position (m, n) of the cell. The event *HeartConduction_Cellular* is used to set the boolean states of the variable *Transition*. First guard

of this event states that any path ($p \mapsto q$) is one of the pair from a set of pairs of the conduction network. Next guard ($grd2$) states that the current impulse propagation speed and velocity flag *CCSpeed_CCTime_Flag* is TRUE and a set of coordinate positions ($param$) of neighbouring cells is represented in third guard. Fourth guard states that the current cell position (m, n) is $Passive$ and last guard represents that the cell transition state $Transition$ is *FALSE*. If all guards satisfy then the transition state of a cell becomes *TRUE*.

EVENT HeartConduction_Cellular
ANY $p, q, param$
WHERE
grd1 : $p \mapsto q \in ConductionPath$
grd2 : $CCSpeed\_CCTime\_Flag = TRUE$
grd3 : $param = \{\{m, n\}, \{m+1, n\}, \{m-1, n\}, \{m, n+1\}, \{m, n-1\}\}$
grd4 : $\{m, n\} \in dom(CellS) \wedge CellS(\{m, n\}) = PASSIVE$
grd5 : $NextCellState = CellS(\{m, n\})$
grd6 : $Transition = FALSE$
THEN
act1 : $Transition := TRUE$
END

The event *HeartConduction_Next_UpdateCell* is used to calculate state of neighbouring cells and update the position of a current cell (m, n). First guard of this event represents a set of coordinate positions ($param$) of neighbouring cells and next guard ($grd2$) states that selected neighbouring cells are a set of cells ($dom(NEXT)$). Last guard presents transition state *Transition* is $TRUE$. Action of this event calculates a set of the next neighbouring cells in $act1$. Next action ($act2$) sets $FALSE$ of a transition state. Last two actions update the value of a current cell (m, n) to continuously impulse propagation into whole heart using the conduction network.

EVENT HeartConduction_Next_UpdateCell
ANY $param$
WHERE
grd1 : $param = \{\{m, n\}, \{m+1, n\}, \{m-1, n\}, \{m, n+1\}, \{m, n-1\}\}$
grd2 : $param \in dom(NEXT)$
grd3 : $Transition = TRUE$
THEN
act1 : $NextCellState := NEXT(param)$
act2 : $Transition := FALSE$
act3 : $m :\in \{m-1, m, m+1\}$
act4 : $n :\in \{n-1, n, n+1\}$
END

Finally, we have completed the formal specifications of the heart modeling. In the next section, we present model validation of the heart model using Event-B model checker ProB tool.

6.7 Model Validation and Analysis

There are two main validation activities in Event-B and both are complementary for designing a consistent system in medical domain:

- *consistency checking*, which is used to show that the events of a machine preserve the invariant, and *refinement checking*, which is used to show that one machine is a valid refinement of another. A list of automatically generated proof obligations should be discharged by the proof tool of the RODIN platform.
- *model analysis*, which is done by the ProB tool and consists in exploring traces or scenarios of our consistent Event-B models. For instance, the ProB may discover possible deadlocks or hidden properties that are not expressed by generated proof obligations.

This section conveys the validity of the model by using ProB tool [20, 45] and Proof Statistics. "Validation" refers to the activity of gaining confidence that the developed formal models are consistent with the requirements. We have used the ProB tool [20] that supports *automated consistency checking* of Event-B machines via model checking [46] and constraint-based checking [47]. Animation using ProB worked very well and we have then used ProB to validate the Event-B machine. This tool assists us to validate the heart model according to the conduction network and a set of landmark nodes. It is the complementary use of both techniques to develop formal models of critical systems, where high safety and security are required. The heart model is carefully verified through animations and under supervision of physiologist and cardiologist. We have validated different kinds of scenario cases of normal and abnormal heart conditions and we have also tested morphological behavior [17, 19] of the ECG during impulse propagation from SA node (A) to Purkinje fibers (F, H) in ventricles. Logical based mathematical model of the heart can generate all possible scenarios of a normal and an abnormal heart conditions in the electrocardiogram due to changing in time and velocity among landmark nodes. ProB was very useful in the development of the heart model. It was able to animate all of our models and verify an absence of error (no counter example exist) and a deadlock. Such kind of errors would have been more difficult to uncover with the prover of RODIN tool.

Model	Total number of POs	Automatic Proof	Interactive Proof
Abstract Model	29	22(76%)	7(24%)
First Refinement	9	6(67%)	3(33%)
Second Refinement	159	155(97%)	4(3%)
Third Refinement	10	1(10%)	9(90%)
Fourth Refinement	11	10(91%)	1(9%)
Total	218	194(89%)	24(11%)

Table 3. Proof Statistics

Table 3 is expressing the proof statistics of the development in the RODIN tool. These statistics measure the size of the model, the proof obligations generated and dis-

charged by the RODIN prover, and those are interactively proved. The complete development of the heart model results in 218(100%) proof obligations, in which 194(89%) are proved automatically by the RODIN tool. The remaining 24(11%) proof obligations are proved interactively using RODIN tool. In the heart model, many proof obligations are generated due to an introduction of the new functional behaviors. In order to guarantee the correctness of these functional behaviors, we have established various invariants in the incremental refinements. Most of the proofs are interactively discharged in the 3rd refinement of the heart model. These proofs are quite simple, and achieve with the help of simplifying predicates. Few proof obligations are also proved interactively in other refinements. The incremental refinement of the heart system helps to achieve a high degree of automatic proof.

7 Discussion

This report presents a methodology to model a biological system, like heart, by modeling a biological environment. Main objective of this methodology is to model the heart system and integrate with medical device model like cardiac pacemaker to model the close-loop system for certifying the medical system through certification bodies [6, 5] for the safe operations. Close-loop model of an environment and device modeling is an open problem in the real world. Industries are striving for such kind of approach from long time to validate the system model under the biological environment. We have discovered lots of informations through literature survey and long discussion with cardiologist and physiologist experts, and reach to the conclusion that how in an efficient and optimum way to model the heart system at cellular level architecture. Due to complexity of the cellular level calculation (see Sec. 2) past models are failed to model the heart system. We have proposed the heart model in an abstract way to simulate the desired behavior of the heart system to avoid the complexity. More important, the heart model is based on logico-mathematical theory, which is our primary objective to model the heart system only using simple logico-mathematics. Main reason to use logical-mathematics is to model the heart system, which can be used with formal specification of the medical devices for verification as a close loop system for certification purpose. Medical experts have elaborated every minor details to understand the complexity of the biological system, specially a heart system is most complex organ in the whole body. The proposed approach contains only main part to specify the system behavior and rest of informations are hidden. We have spent a lot of time to discover the exact abstract model of the heart system, which complies with medical experts. We have used Event-B modeling language to model the system and verify the correctness of the heart system uses ProB model checker, while any other formal specification language as well as model checkers can be used to model the heart system based on our proposed methodology.

8 Conclusion and Future Challenges

8.1 Conclusion

This report has presented a methodology for modeling a mathematical heart model based on logico-mathematical theory. The heart model is based on electrocardiography

analysis, which models the heart system at cellular level. This is the most challenging problem to validate and verify the correct behavior of the developed system model under biological environment (i.e. heart). We have proposed a method to develop a heart model based on logico-mathematical theory. For formalizing the heart system, we have used Event-B modeling language [15, 13] to develop the proof-based formal model. Our approach for formalizing and reasoning about impulse propagation into whole heart system through the conduction network (see Fig. 4(a)). The heart model suggests that such an approach can yield a viable model that can be subjected to useful validation against medical device softwares at an early stage in the development process (i.e. cardiac pacemaker).

More precisely, we would like to stress the original contribution of our work. We have proposed a method for modeling a mathematical heart model based on logico-mathematical theory. Main objectives of this proposed idea are as follows:

- To obtain the certification for providing higher safety integrity level.
- To verify the system under patient model (in formal represents).
- To analyse the biological environment (the heart) in a mathematical way.
- To analyse the interaction between heart model and cardiac pacemaker or ICDs.

For quick understanding, we have formalised the given characteristics and physiological behavior of the heart. The formalisation is highlighting a different aspect of the problem, making different assumptions about the impulse propagation and establishing different properties related to the cell automaton. We have outlined how an incremental refinement approach of the heart system allows to achieve a high degree of automatic proof using RODIN tool. Our different developments reflect not only many facets of the problem, but also that there is a 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 are performed by ProB model checker regarding safety conditions. As part of our reasoning, we have proved that the initialisation of the system is a valid one and we have calculated preconditions of operations. The latter has been executed to guarantee that our intention to have total operations has been fulfilled. At every stage of refinement we introduced the new behavior of the system and proved the *consistency* and *refinement checking*. We have introduced the more general invariants at refinement level that the initialisation of the whole system is valid. Finally, we have validated the heart system using the ProB model checker as validation tool and verify the correctness of exact behavior of our proved heart system with the help of physiologist and cardiologist experts.

8.2 Future Challenges

Our most important goal is that this formal model helps to obtain a certification for the medical devices related to the heart system such as cardiac pacemaker and ICDs. As far as, it can be also used as a diagnostic tool to diagnose the patient with the help patient model. This has been the first attempt ever in heart modeling based on logico-mathematical theory. We have successfully model the electrical impulse propagation

of the heart system. Main cause of any heart's diseases is trouble in heart conduction network [2, 17]. Medical devices are tightly coupled with biological environment (i.e the heart), which use actuators and sensors to response the biological environment. Due to a strong relationship between medical device (i.e. pacemaker) and related biological environment (the heart), it is required to model the functioning of the medical device within the biological environment. The environment model is independent to the device model, which helps to create an environment for the medical device for simulating the desired behavior of a device. The medical device model is a dependent model on the biological environment. Whenever any undesired state occur in the biological environment, the device model must act according the requirements. As a future work, our main objective is to integrate pacemaker formal specification [48, 49] and the heart formal specification to model the closed-loop system for verifying the desired behavior of the cardiac pacemaker for certification purpose.

Acknowledgement. We are grateful to cardiologist experts Dr. Yves Juillière and Dr. Frédérique Claudot and biomedical experts Didier Fass of the Université Henri Poincaré Nancy, who share their experiences for helping to design the methodology and verifying the correctness of proposed approach. Work of Neeraj Kumar Singh is supported by grant awarded by the Ministry of University and Research.

References

1. Jaakko Malmivuo, R.P. In: Bioelectromagnetism. Oxford University Press (1995) ISBN 0-19-505823-2.
2. Khan, M.G.: Rapid ECG Interpretation. Humana Press (2008)
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) (2001) 793–799
4. Center for Devices and Radiological Health: Safety of Marketed Medical Devices,US FDA (2006)
5. 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
6. Keatley, K.L.: A review of the fda draft guidance document for software validation: guidance for industry. Qual Assur **7**(1) (1999) 49–55
7. 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) (2006) 33–38
8. Méry, D., Singh, N.K.: Trustable formal specification for software certification. In Margaria, T., Steffen, B., eds.: Leveraging Applications of Formal Methods, Verification, and Validation. Volume 6416 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2010) 312–326
9. Bowen, J., Stavridou, V.: Safety-critical systems, formal methods and standards. Software Engineering Journal **8**(4) (Jul 1993) 189–209
10. Jetley, R.P., Carlos, C., Iyer, S.P.: A case study on applying formal methods to medical devices: computer-aided resuscitation algorithm. STTT **5**(4) (2004) 320–330
11. Jetley, R., Purushothaman Iyer, S., Jones, P.: A formal methods approach to medical device review. Computer **39**(4) (April 2006) 61–67

12. Méry, D., Singh, N.K.: Real-time animation for formal specification. In Aiguier, M., Bretaudeau, F., Krob, D., eds.: Complex Systems Design & Management. Springer Berlin Heidelberg (2010) 49–60
13. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2010)
14. Fitzgerald, J. In: The Typed Logic of Partial Functions and the Vienna Development Method. Springer (2007) 431–465 See [39].
15. Project RODIN: Rigorous open development environment for complex systems. http://rodin-b-sharp.sourceforge.net/ (2004)
16. Harrild, D.M., Henriquez, C.S., Atria, T.H., Harrild, D.M., Henriquez, C.S.: Cs, a computer model of normal conduction. In: in the Human Atria, Circ Res, vol 87. (2000) 25–36
17. Bayes de Luna A., Batcharov, V.N., Malik, M. In: "The morphology of the Electrocardiogram" in The ESC Textbook of Cardiovascular Medicine. Blackwell Publishing Ltd. (2006) 1–36
18. John von Neumann: Theory of Self-Reproducing Automata. University of Illinois Press, 1966 Edited by Arthur W. Burks
19. Jean-Yves Artigou, Jean-Jacques Monsuez, S.f.c. In: Cardiologie et maladies vasculaires. Elsevier Masson (2006)
20. ProB: The ProB animator and model checker for the B method. http://www.stups.uni-duesseldorf.de/ProB/overview.php/
21. Plonsey, R., Barr, R.C.: Mathematical modeling of electrical activity of the heart. Journal of Electrocardiology **20**(3) (1987) 219 – 226
22. Kye-Rok Jun, Y.R.S., Kim, T.G.: A cellular automata model of activation process in ventricular muscle
23. Berenfeld, O., Abboud, S.: Simulation of cardiac activity and the ecg using a heart model with a reaction-diffusion action potential. Medical Engineering & Physics **18**(8) (1996) 615 – 625
24. Adam, D.: Propagation of depolarization and repolarization processes in the myocardium-an anisotropic model. Biomedical Engineering, IEEE Transactions on **38**(2) (feb. 1991) 133 –141
25. Jiang, Z., Pajic, M., Connolly, A.T., Dixit, S., Mangharam, R.: Real-time heart model for implantable cardiac device validation and verification. In: 22st Euromicro Conference on Real-Time Systems, (IEEE ECRTS'10). (07/2010 2010)
26. Leitner, F., Leue, S.: Simulink Design Verifier vs. SPIN a Comparative Case Study, Proceedings of FMICS (2008)
27. : Heart image weblink : http://media.summitmedicalgroup.com/media/db/ relayhealth-images/nodes.jpg
28. Barold, S.S., Stroobandt, R.X., Sinnaeve, A.F. In: Cardiac Pacemakers Step by Step. Futura Publishing (2004) ISBN 1-4051-1647-1.
29. Ellenbogen, K.A., Wood, M.A. In: Cardiac Pacing and ICDs. 4th Edition, Blackwell (2005) ISBN-10 1-4051-0447-3.
30. Hesselson, A. In: Simplified Interpretations of Pacemaker ECGs. Blackwell Publishers (2003) ISBN 978-1-4051-0372-5.
31. 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) (2006) 33–38
32. Love, C.J. In: Cardiac Pacemakers and Defibrillators. Landes Bioscience Publishers (2006) ISBN 1-57059-691-3.
33. Report: Recommendations for pacemaker prescription for symptomatic bradycardia. British Heart Journal **66**(2) (1991) 185–189

34. Writing Committee Members, and Epstein, Andrew E. and DiMarco, John P. and Ellenbogen, Kenneth A. and Estes, N.A. Mark, III and Freedman, Roger A. and Gettes, Leonard S. and Gillinov, A. Marc and Gregoratos, Gabriel and Hammill, Stephen C. and Hayes, David L. and Hlatky, Mark A. and Newby, L. Kristin and Page, Richard L. and Schoenfeld, Mark H. and Silka, Michael J. and Stevenson, Lynne Warner and Sweeney, Michael O.: ACC/AHA/HRS 2008 Guidelines for Device-Based Therapy of Cardiac Rhythm Abnormalities, Developed in Collaboration With the American Association for Thoracic Surgery and Society of Thoracic Surgeons. Circulation **117**(21) (2008) 2820–2840
35. Makowiec, D.: The heart pacemaker by cellular automata on complex networks. In: Proceedings of the 8th international conference on Cellular Automata for Reseach and Industry. ACRI '08, Berlin, Heidelberg, Springer-Verlag (2008) 291–298
36. Vangheluwe, H., Vansteenkiste, G.C.: The cellular automata formalism and its relationship to devs. In: Proceedings of the 14th European Simulation Multiconference on Simulation and Modelling: Enablers for a Better Quality of Life, SCS Europe (2000) 800–810
37. Cansell, D., Méry, D. In: The event-B Modelling Method: Concepts and Case Studies. Springer (2007) 33–140 See [39].
38. Bjorner, D.: Software Engineering 1-2-3. Texts in Theoretical Computer Science. An EATCS Series. Springer (2006)
39. Bjørner, D., Henson, M.C., eds.: Logics of Specification Languages. EATCS Textbook in Computer Science. Springer (2007)
40. Leavens, G.T., Abrial, J.R., Batory, D., Butler, M., Coglio, A., Fisler, K., Hehner, E., Jones, C., Miller, D., Peyton-Jones, S., Sitaraman, M., Smith, D.R., Stump, A.: Roadmap for enhanced languages and methods to aid verification. In: Fifth Intl. Conf. Generative Programming and Component Engineering (GPCE 2006), ACM (October 2006) 221–235
41. ClearSy Aix-en-Provence (F): Atelier B. (2002) Version 3.6.
42. Gamma, E., Helm, R., Johnson, R., Vlissides, R., Gamma, P.: Design Patterns : Elements of Reusable Object-Oriented Software design Patterns. Addison-Wesley Professional Computing (1994)
43. Rehm, J.: Pattern Based Integration of Time applied to the 2-Slots Simpson Algorithm. In: Integration of Model-based Formal Methods and Tools in IFM'2009, Düsseldorf Allemagne (02 2009)
44. Back, R., von Wright, J.: Refinement Calculus A Systematic Introduction. Graduate Texts in Computer Science. Springer (1998)
45. Leuschel, M., Butler, M. LNCS. In: ProB: A Model Checker for B. Springer (2003) 855–874
46. E. M. Clarke, O.G., Peled, D. In: Model Checking. MIT Press (1999) ISBN 978-0262032704.
47. Jackson, D.: Alloy: a lightweight object modelling notation. ACM Trans. Softw. Eng. Methodol. **11**(2) (2002) 256–290
48. Méry, D., Singh, N.K.: Functional behavior of a cardiac pacing system. International Journal of Discrete Event Control Systems **1**(2) (2011) 129–149
49. EB2ALL: Automatic code generation from Event-B to many Programming Languages. http://eb2all.loria.fr/ (2011)