Chapter 8
Formal Logic Based Heart-Model

Abstract A closed-loop model of a system is considered as a *de facto* standard in the area of system engineering for validating a system model. Cardiac pacemakers and implantable cardioverter-defibrillators (ICDs) are the most critical of these medical devices, requiring closed-loop modelling (integrated system and environment modelling) for verification purposes before obtaining a certificate from the certification bodies. This chapter presents a methodology for modelling a biological system, such as the heart, to enable modelling in a biological environment. The heart model is based mainly on electrocardiography analysis, which models the heart system at the cellular level. This heart model will be used for modelling the closed-loop system of a cardiac pacemaker.

8.1 Introduction

The human heart is well known as a mechanical device of amazing efficiency that pumps blood via the circulatory system continuously throughout the person's lifetime. It is one of the most complex and important biological systems, providing oxygen and nutrients to the body to sustain life [30]. The regular impulses generated by the heart result in rhythmic contractions through a sequence of muscles in the heart, beginning at the natural pacemaker known as the sinoatrial (SA) node, which produces an action potential that travels across the atrioventricular (AV) node, the bundle of His and the Purkinje fibres distributed throughout the ventricles. The pattern and the timing of these impulses determine the heart rhythm. Variable time intervals and conduction speeds during the heartbeat generate abnormal heart rhythms, which are also known as heart rhythm impairments. Heart rhythm impairment is the principal source of several diseases. Electrocardiography analysis is frequently used to diagnose various types of heart disease [24] by presenting the timing properties of the electrical system of the heart. These are the most fundamental properties of the heart.

Cardiac pacemakers and ICDs are the two main types among the remarkable range of medical and technological devices recommended by doctors in cases of abnormal heart rhythm. These devices are used to maintain the heart rhythm, and are life-saving in many instances. In the last few years, the use of cardiac pacemakers and cardioverter-defibrillators has increased. However, these devices may

N.K. Singh, *Using Event-B for Critical Device Software Systems*,
DOI 10.1007/978-1-4471-5260-6_8, © Springer-Verlag London 2013

sometimes malfunction. Device related problems have been responsible for a large number of serious injuries. Many deaths and injuries caused by device failure have been reported by the FDA [28], which advocates safety and security guidelines for using these devices. FDA officials have found that many deaths and injuries related to the devices are caused by product design and engineering flaws, which can be considered as firmware problems [10, 17].

Providing assurance guarantees for medical devices makes formal approaches appealing. Formal model based methods have been successful in targeted applications of medical devices [9, 20, 21, 25, 31, 32]. Over the past decade, there has been considerable progress in the development of formal methods for improving confidence in complex software-based systems [1, 13, 14]. Although formal methods are part of the standard recommendations for developing and certifying medical systems, the integration of formal methods into the certification process is, in large part, unclear. In particular, it is a very challenging task to ensure that the end product of the software-development system behaves securely.

8.1.1 Motivation

The most challenging problem is environment modelling. That is, to validate and to verify the correct behaviour of a system model requires an interactive formal model of the environment. For example, a formal model of a cardiac pacemaker or ICD requires a heart model to verify the correctness of the developed system (see Fig. 8.1). No tools and techniques are available to provide environment modelling that would enable verification of the developed system model. Medical devices are tightly coupled with their biological environment (i.e., the heart) and use actuators and sensors to interact with the biological environment. Because of this strong relationship between the medical device (e.g., a pacemaker) and the related biological environment (i.e., the heart), it is necessary to model the functioning of the medical device within the biological environment.

The environment model will be independent of the device model, which is helpful in creating an environment for medical devices that simulates the actual behaviour of the system. The medical device model will be dependent on the biological environment. Whenever an undesired state occurs in the biological environment, the device model must act according to the requirements. The main objective is to use a formal approach to modelling the medical device and the biological environment to verify the correctness of the medical system.

To model the biological environment (the heart) for a cardiac pacemaker or ICD, we propose a method for modelling the heart using logico-mathematical theory [33–35]. The heart model is based on electrocardiography analysis [6, 15, 24], which models the heart system at the cellular level [40]. In this investigation, we present a methodology for modelling a heart that involves extracting a set of biological nodes (SA node, AV node, etc.), impulse propagation speeds between nodes, impulse propagation times between nodes and cellular automata (CA) for propagating impulses

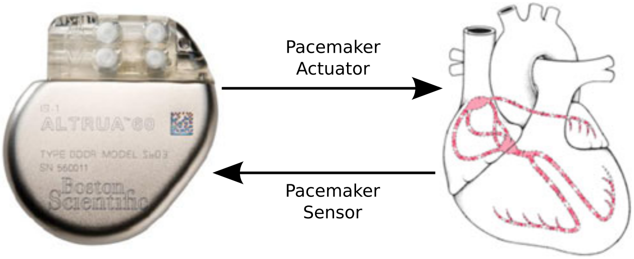


Fig. 8.1 Cardiac pacemaker and heart interaction

at the cellular level. This model is developed through incremental refinement, which introduces several properties in an incremental way and verifies the correctness of the heart model. A key feature of this heart model is the representation of all possible morphological states of the (ECG) [3, 6]. These morphological states represent both the normal and the abnormal states of the ECG. The morphological representation can generate any kind of heart model (a patient's model or a normal heart model) using the ECG. This model can observe both the failure of impulse generation and the failure of impulse propagation. The mathematical heart model, based on logico-mathematical theory, is verified using the RODIN [38] proof tool and the model checker ProB [26]. The model is also verified by electro-physiology and cardiac experts. The main objective of this heart model is to provide a biological environment (the heart) for formalising a closed-loop system (a combined model of a cardiac pacemaker and the heart).

8.1.2 Structure of This Chapter

The outline of the remaining chapter is as follows. Section 8.2 presents related work. A brief outline of the heart system is introduced in Sect. 8.3. Section 8.4 explains the proposed approach. Section 8.5 gives an outline of the formal development of the heart model. Section 8.6 discusses the results of lessons learnt from this experience, and Sect. 8.7 summarises the chapter.

8.2 Related Work

Heart modelling is a challenging problem in the area of real-time simulation for clinical purposes. It is handled by the research community using a variety of different methods. The ECG is an important diagnostic method for measuring the heart's electrical activities, and was invented by Willem Einthoven in 1903 [36]. In this study, the ECG is used in modelling the heart [36]. At the present time, technological advances have enabled the production of a high-quality cellular model of an entire heart.

K.R. Jun et al. [37] have produced a CA model of the activation process in ventricular muscle tissue. They presented a two-dimensional (2D) CA model that accounts for the local orientation of the myocardial fibres and their distributed velocity and refractory period. A three-dimensional (3D) finite-volume-based computer mesh model of human atrial activation and current flow has been presented by Harrild et al. [15]. This cellular-level-based model included both the left and right atria and the major muscle bundles of the atria. The results of using this model demonstrate a normal sinus rhythm and can extract the patterns of the septum's activation. Because of memory and time complexity in the computation of a 3D model, an empirical approach is used in modelling the whole heart. The empirical approach implies a simpler representation of the complex process at a cellular level. In this new approach, researchers have adopted some approximations in modelling the whole heart without compromising the actual behaviour of the heart. Berenfeld et al. [8] have developed a model that can give insight into the local and global complex dynamics of the heart in the transition from normal to abnormal myocardial activity, which helps to estimate myocardial properties. Adam [2] has analysed wave activities during depolarisation in his cardiac model, which is represented by a simplification of the heart tissue.

Recently, a real-time Virtual Heart Model (VHM) has been developed by Jiang et al. [22] to model the electro-physiological operation of proper functioning and malfunctioning. They used a time-automaton model to define the timing properties of the heart. Simulink Design Verifier[1] was used as the main tool for designing the VHM. A heart model based on Uppaal Model checker [7] is developed by Jee et al. [19] for developing the cardiac pacemaker model. This is a very simple heart model, which provides an environment to simulate and verify the pacemaker software in modelling phase.

Our approach is based purely on formal techniques for modelling the heart using electrocardiography analysis. To model the heart for a cardiac pacemaker or ICD, we propose a method based on logico-mathematical theory, which can be implemented using any formal-methods-based tools (Z, TLA$^+$, VDM, etc.). In this chapter, the model is developed using a maximal refinement approach at the cellular level. The incremental refinement approach helps both to introduce several properties in an incremental way and to verify the correctness of the heart model [33–35]. The key feature of this heart model is the representation of all possible morphological states of the ECG, which is used to represent both normal and abnormal states through observation of the failure of impulse generation and the failure of impulse propagation in the heart [3, 6, 24, 30].

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

8.3 Background

8.3.1 The Heart System

The human heart is wondrous in its ability to pump blood to the circulatory system continuously throughout a lifetime. The heart comprises four chambers: right atrium, right ventricle, left atrium and left ventricle, each of which contract and relax periodically. The atria form one unit and the ventricles another. The heart's mechanical system (the pump) requires impulses from its electrical system to function. An electrical stimulus is generated by the sinus node (see Fig. 8.2), which is a small mass of specialised 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 the 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 contract for a fraction of a second before the ventricles, so their blood empties into the ventricles before the ventricles contract.

Arrhythmias are caused by cardiac problems that produce 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 it may be characterised by an irregular cardiac rhythm, such as that caused by asynchrony of the cardiac chambers. Irregularities in the heartbeat are called bradycardia and tachycardia. Bradycardia indicates that the heart rate falls below the expected level whereas tachycardia indicates that the heart rate goes above the expected heart rate. An artificial pacemaker can restore synchrony between the atria and the ventricles [5, 12, 16, 25, 27, 30]. Beats per minute (bpm) is the basic unit used to measure the rate of heart activity.

8.3.2 Basic Overview of Electrocardiogram (ECG)

The ECG (or EKG) [16, 24] is a diagnostic tool that measures and records precisely the electrical activity of the heart in the 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 to diagnose a wide range of heart conditions and to predict the related diseases. ECG records are obtained by sampling the bioelectric currents sensed by several electrodes, known as leads. A normal ECG is depicted in Fig. 8.3. 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 [5, 24, 27, 30].

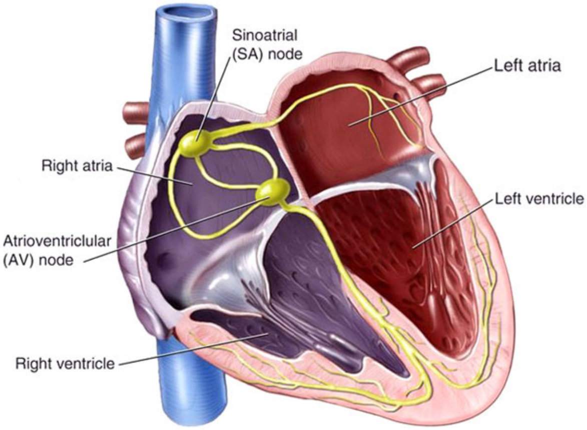


Fig. 8.2 Heart or natural pacemaker

All the segments and intervals used by clinicians are represented in this ECG diagram. Depolarisation and repolarisation of the ventricular and atrial chambers are presented by deflection in the ECG signal. These deflections are labelled in alphabetic order: P-QRS-T. Letter P indicates atrial depolarisation, and the ventricular depolarisation is represented by the QRS complex. The ventricular repolarisation is represented by T-wave. Atrial repolarisation appears during the QRS complex and generates a very low amplitude signal which cannot be uncovered from the normal ECG signal.

8.3.3 ECG Morphology

Sequential activation, depolarisation, and repolarisation are distinct deflections in the ECG, caused by anatomical differences between the atria and the ventricles. The sequences are even distinguishable when they are not in the correct sequence (P-QRS-T). Each beat of the heart can be observed as a series of deflections, which reflects the time evolution of electrical activity in the heart [3, 6, 24]. A single cycle of the ECG is considered as one heartbeat. The ECG may be divided into the following sections.

- *P-wave:* 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:* The QRS-complex is easily identifiable between the P- and T-waves because it has a 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 ven-

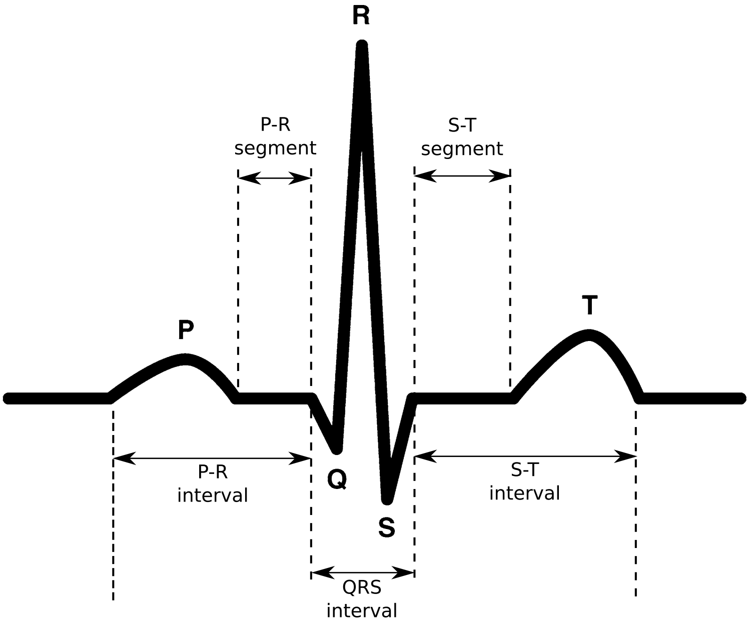


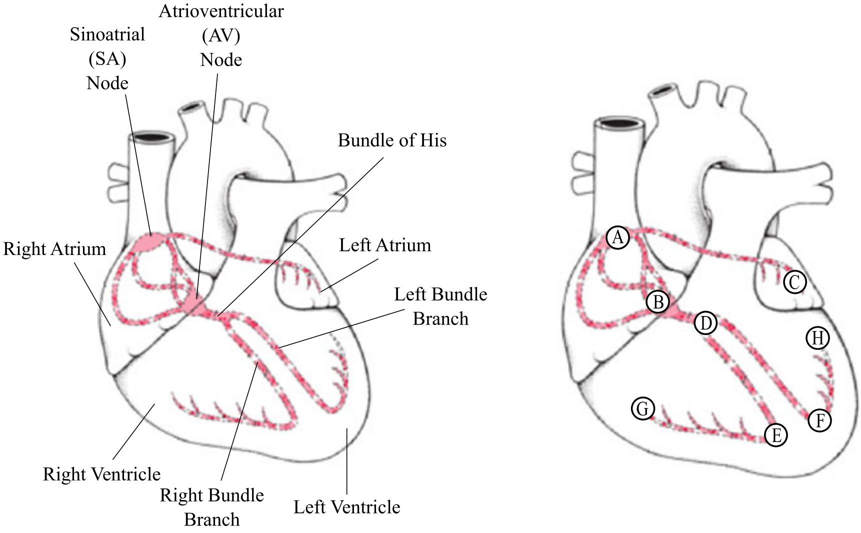
Fig. 8.3 A typical one-cycle ECG tracing

tricular 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. A prolonged QT-interval may be associated with delayed ventricular repolarisation, which may cause ventricular tachyarrhythmias leading to sudden cardiac death.
- *ST-interval:* The time between the end of the S-wave and the beginning of the 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.

8.4 Proposed Idea

Our proposed method exploits a heart model based on logico-mathematics to help the formal methods community to verify the correctness of a developed model of medical devices such as cardiac pacemakers. The heart model is based mainly on the impulse propagation time and conduction speed at a cellular level [33–35]. This method uses the advanced capabilities of a combined approach of formal verification



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

Fig. 8.4 The electrical conduction and landmarks of the heart system

and model validation using a model checker to achieve considerable advantages in heart system modelling.

Figure 8.4(a) shows the more significant components and the impulse conduction path in the entire heart system. The heart is a muscle with a special electrical conduction system. The system comprises two nodes (special conduction cells) and a series of conduction fibres or bundles (pathways). For modelling the heart system, we have assumed eight landmark nodes (A, B, C, D, E, F, G, H) in the whole conduction network, as shown in Fig. 8.4(b), which control the whole heart system. These landmarks were identified via a literature survey [24, 30] and extensive discussions with a cardiologist and a physiologist. Centimetres per second (cm/sec) is a basic unit to measure the conduction speed and milliseconds (ms) is a basic unit to measure the conduction time.

We now introduce the necessary elements we use to define the heart system formally.

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 $\mathrm{HSys} = (N, T, N_0, TW_{time}, CW_{speed})$ where:

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

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

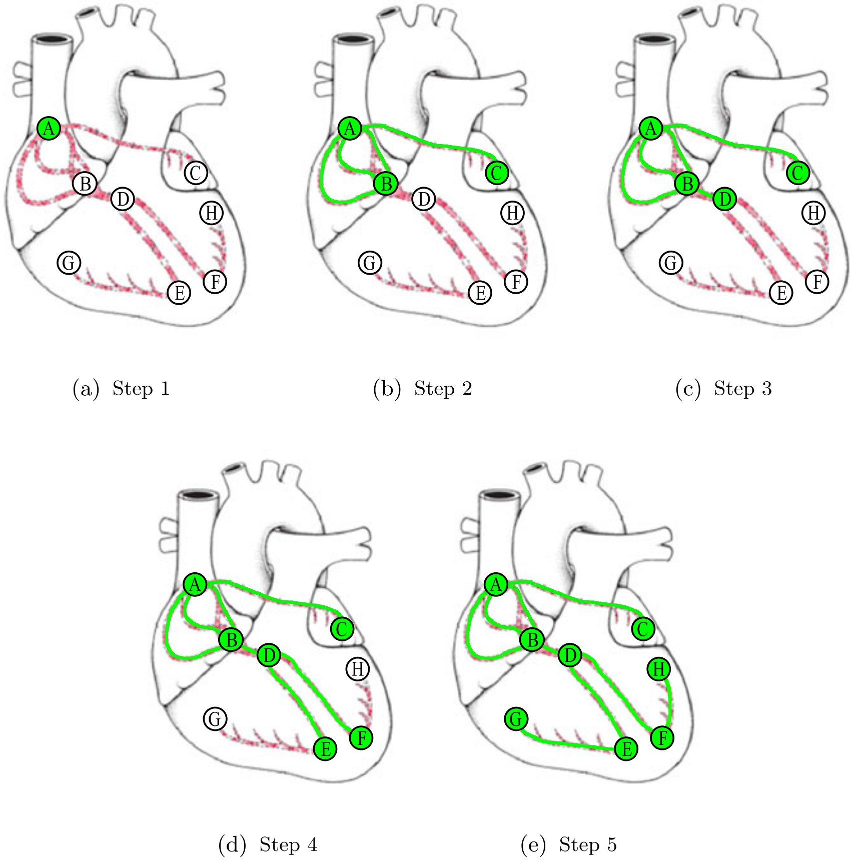
Property 1 (Impulse propagation time) In the heart system, the electrical impulse originates from SA node (node A), travels through the entire conduction network and terminates at the atrial muscle fibres (node C) and at the end of Purkinje fibres in both sides of the ventricular chambers (node G and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

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

Electrical activity is spontaneously generated by the SA node, located high in the right atrium, shown as node A in Fig. 8.5(a). The SA node is the physiological pacemaker of the normal heart, responsible for setting its 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. 8.3). From the sinus node, the electrical impulse propagates throughout the atria and reaches nodes B and C, but cannot propagate directly across the boundary between the atria and ventricles. The electrical impulse travels outward into the atrial muscle fibres and reaches the end of the fibres, shown as node C in the conduction network (see Fig. 8.5(b)).

Normally, the only pathway available for the electrical impulse is to enter the ventricles through a specialised region of cells called the AV node. The AV node is located at the boundary between the atria and ventricles, shown as node B in Fig. 8.4(b). The AV node provides the 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 would not flow effectively from the atria to the ventricles. The delay in the AV node forms much of the PR segment on the ECG. Part of the atrial repolarisation can be represented by the PR segment (see Fig. 8.3).

Propagation from the AV node (A) to the ventricles is provided by a specialised conduction system. The distal portion of the AV node is composed of a common bundle called the Bundle of His, shown as landmark node D in Fig. 8.4(b). The



(a) Step 1 (b) Step 2 (c) Step 3

(d) Step 4 (e) Step 5

Fig. 8.5 Impulse propagation through landmarks of the heart system

Bundle of His splits into two branches in the inter-ventricular septum, namely 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 follow the left and right bundle branches along the inter-ventricular septum (see Fig. 8.5(c)).

The two separate bundle branches propagating along each side of the septum constitute the left and right bundle branches.We have identified two landmark nodes E and F (see Fig. 8.4(b)) in the lower part of the heart for the left and right bundle branches. These specialised fibres conduct the impulses at a very rapid velocity (see Tables 8.1 and 8.2). The left bundle branch activates the left ventricle, whereas the right bundle branch activates the right ventricle (see Fig. 8.5(d)).

The bundle branches then divide into an extensive system of Purkinje fibres that conduct the impulses at high velocity (see Tables 8.1 and 8.2) throughout the ventricles. The Purkinje fibres stimulate individual groups of myocardial cells to contract. We have identified two final landmark nodes G and H (see Fig. 8.4(b)) at the end of the Purkinje fibres in both sides of the ventricles. These two nodes represent the end

Table 8.1 Cardiac activation time in the heart

Location in the heart	Cardiac activation time (ms)
SA node (A)	0..10
Left atria muscle fibres (C)	70..90
AV node (B)	50..70
Bundle of His (D)	125..160
Right bundle branch (E)	145..180
Left bundle branch (F)	145..180
Right Purkinje fibres (G)	150..210
Left Purkinje fibres (H)	150..230

Table 8.2 Cardiac activation velocity in the heart

Location in the heart	Conduction velocity (cm/sec)
$A \mapsto B$	30..50
$A \mapsto C$	30..50
$B \mapsto D$	100..200
$D \mapsto E$	100..200
$D \mapsto F$	100..200
$E \mapsto G$	300..400
$F \mapsto H$	300..400

of the conduction network in the heart system. The bundles branch into the Purkinje fibres that diverge across the inner sides of the ventricular walls (see Fig. 8.5(e)). On reaching the end of the Purkinje fibres, the electrical impulse is transmitted through the ventricular muscle mass by the ventricular muscle fibres 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 [24, 30].

The electrical system provides a synchronised system from atria to ventricles, which aids the contraction of the heart muscle and optimises the haemodynamics. Changed time intervals or conducting speeds between landmarks (see Fig. 8.4(b) and Fig. 8.6) are a major cause of abnormalities in the heart system. Abnormalities in electrical signals in the heart can generate various kinds of arrhythmias. A slow conduction speed generates bradycardia and a fast conduction speed generates tachycardia. In this model, we consider the ranges of all possible values for conduction speeds and conduction times for each landmark node and conduction path. This model represents the morphological structure of the ECG signal through the conduction network (see Fig. 8.6).

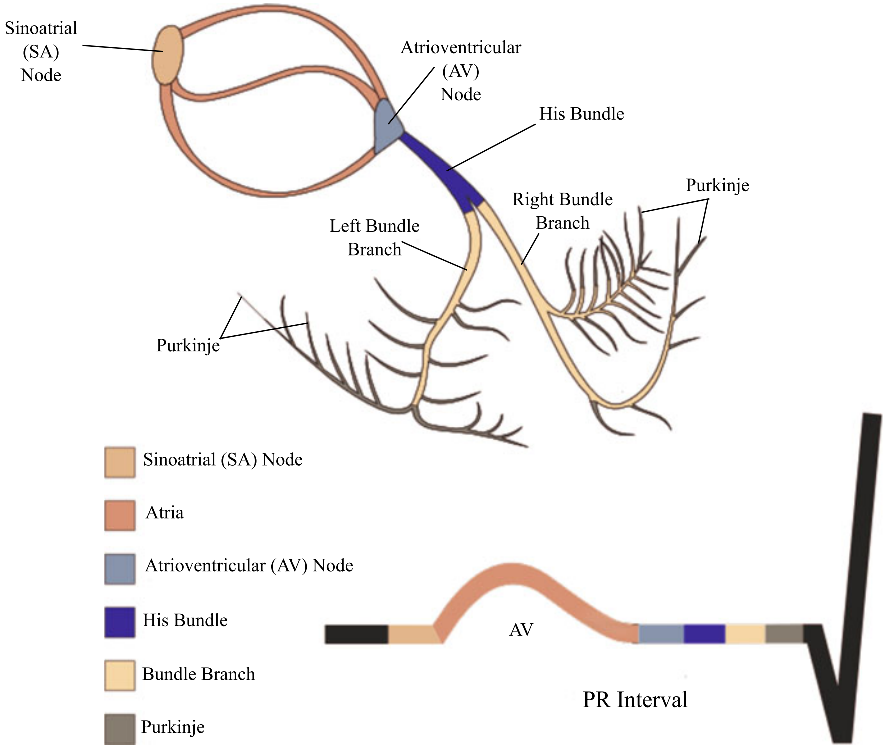


Fig. 8.6 Time intervals and impulse propagation in the ECG signal (adapted from [30])

8.4.1 Heart Block

In this section, we explain the basic heart blocks in the heart conduction system. We have formalised these basic heart blocks in the proposed methodology. Heart block is the term given to a disorder of conduction of the impulse that stimulates heart muscle contraction. The normal cardiac impulse arises in the SA node (A), situated in the right atrium, and spreads to the AV node (B), whence it is conducted by specialised tissue known as the Bundle of His (D), which divides into the left and right bundle branches in the ventricles (see Fig. 8.4(a)). Disturbances in conduction may appear as slow conduction, intermittent conduction failure or complete conduction failure. These three kinds of conduction failure are also known as 1st, 2nd and 3rd degree blocks. We can show these different kinds of heart block throughout the conduction network in terms of our set of landmark nodes (see Fig. 8.7).

SA Block

This block occurs within the SA node (A) and is described as an SA nodal block or sick sinus syndrome. The SA node fails to originate an impulse, and the heart misses one or two beats at regular or irregular intervals (see Fig. 8.7(a)).

AV Block

For an AV block, the sinus rhythm is normal, but there is a conduction defect between the atria and the ventricles. The main cause of this block may be in the AV node (B) or the Bundle of His (D), or both (see Fig. 8.7(b)).

Infra-Hisian Block

Blocks that occur below the AV node (B) are known as Infra-Hisian blocks (see Fig. 8.7(c)). This block describes block of the distal conduction system and it includes Type 2 second degree heart block.

Left Bundle Branch Block

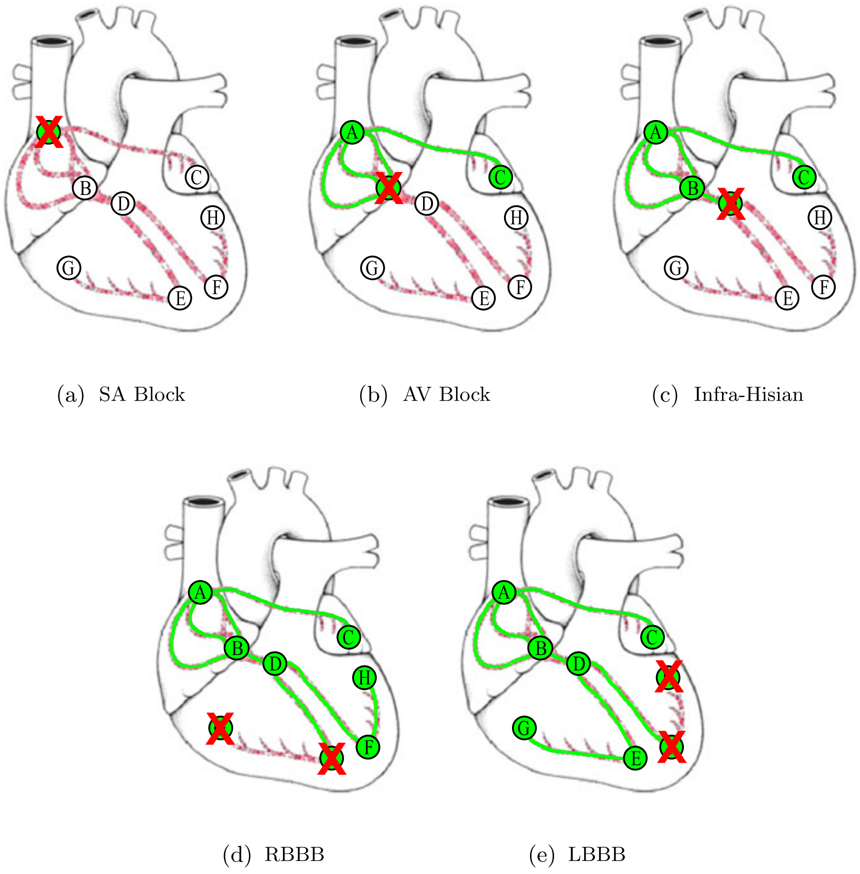
In the normal heart, activation of both ventricles takes place simultaneously. A left bundle branch block occurs when conduction into the left branch of the Bundle of His is interrupted. Blocks that occur within the fascicles of the left bundle branch are known as hemiblocks (see Fig. 8.7(d)).

Right Bundle Branch Block

A right bundle branch block occurs when conduction into the right branch of the Bundle of His is interrupted (see Fig. 8.7(e)).

8.4.2 Cellular Automata Model

A set of spatially distributed cells form a CA model, which contains a uniform connection pattern among neighbouring cells and local computation laws. CA were originally proposed by Ulam and von Neumann [40] in the 1940s to provide a formal framework for investigating the behaviour of complex, spatially distributed systems. CA are discrete dynamic systems corresponding to space and time. CA modelling involves uniform properties for state transitions and interconnection patterns. The model components are specified by a single property caused by the same patterns



(a) SA Block

(b) AV Block

(c) Infra-Hisian

(d) RBBB

(e) LBBB

Fig. 8.7 Impairments in impulse propagation due to the heart blocks

instead of specifying each component separately. CA models help to visualise a system's dynamics [15, 29, 30, 39]. A CA model can have an infinite number of cells along any dimension. Here, we consider a finite number of cells in two dimensions, as shown in Fig. 8.8. A 2D CA model is defined as:

Definition 2 (The CA model)

$$\text{(CA)} = \langle S, N, T \rangle\colon \text{ discrete time system}$$

$$S\colon \text{ the set of states}$$

$$N\colon \text{ the neighbouring patterns at } (0, 0)$$

$$T\colon \text{ the transition function}$$

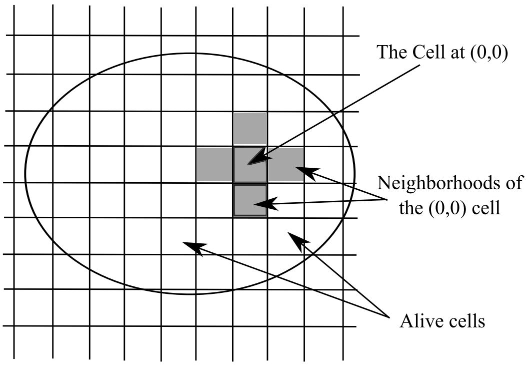


Fig. 8.8 A two-dimensional cellular automata model

In the usual case of CA realised on a D-dimensional grid, N consists of D-tuples of indices from a coordinate set:

$$I\colon\ N \subseteq I^D$$

The 2D cellular model therefore becomes

$$\begin{aligned} &N \subseteq I^2 \\ &T : S^{|N|} \to S \end{aligned}$$

To consider an automaton specified as a CA, let λ and α be the global state and the global transition function of the CA, respectively. Then, $\lambda = \{\tau | \tau : I^2 \to S\}$ and $\alpha(\lambda(i, j)) = T(\tau | N + (i, j))$ for all τ in λ and (i, j) in I^2.

Definition 3 (State transition of a cell) The heart muscle system is composed of heterogeneous cells, the CA model of the muscle system, CAM_{CA}, is characterised by having no dependency on the type of cells. CAM_{CA} is defined as follows:

$$\begin{aligned} &CAM_{CA} = \langle S, N, T\rangle \\ &S = \{Active, Passive, Refractory\} \\ &N = \{(0,0), (1,0), (-1,0), (0,1), (0,-1)\} \\ &s'_{m,n} = s_{m,n}(t+1) \\ &s'_{m,n} = T(s_{m,n}, s_{m+1,n}, s_{m-1,n}, s_{m,n+1}, s_{m,n-1}) \end{aligned}$$

where, $s_{m,n}$ denotes the state of the cell located at (m, n) and T is a transition function for CAM_{CA} that specifies the next state, as shown in Fig. 8.9.

Each cell in the heart muscle should be in one of the states *Active*, *Passive* or *Refractory*. Initially, all cells are *Passive*. In this state, the cell is discharged electrically and has no influence on its neighbouring cells. When an electrical impulse propagates, the cell becomes charged and eventually activated (*Active* state). The cell then transmits an electrical impulse to its neighbour cells. The electrical impulse is propagated to all the cells in the heart muscle. After activation, the cell

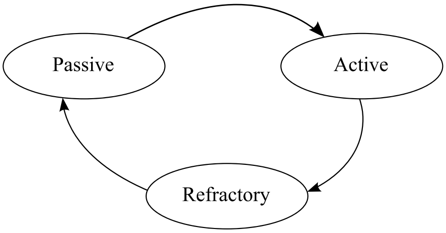


Fig. 8.9 State transition of a cell

becomes discharged and enters the *Refractory* state within which the cell cannot be reactivated. After a time, the cell changes its state to the *Passive* state to await the next impulse.

8.5 Functional Formal Modelling of the Heart

To formalise the heart model, we have used the Event-B modelling language [1, 38], although the proposed idea can be formalised using any kind of formal-methods tool such as Z, ASM, TLA$^+$ or VDM. Event-B modelling language supports the refinement approach [4] that helps to verify the correctness of the system in an incremental way.

The heart model development is expressed in an abstract and general way. The initial model formalises 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 modelling approach.

Initial model: This is an observation model, which specifies a heart state in the form of *true* and *false*, where *true* represents a normal rhythm and *false* represents an 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 fibres 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 a 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 the cellular level using cellular automata.

8.5.1 The Context and Initial Model

Event-B models are described in terms of two major components: *context* and *machine*. The context contains the static part of the model, whereas the 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 a system. The logical theory lists the static properties of constants related to the system and provides an axiomatisation of the system environment. The context can be extended by other contexts and referenced by a set of machines, while a machine can be refined by other machines.

We need to choose electrical features for modelling 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. 8.4(a)). These nodes are basic landmarks, which enable expression of the normal and abnormal behaviour of the heart system. These landmarks were identified through a literature survey [24, 30] and fruitful discussions with a cardiologist and a physiologist. Three constants define the impulse propagation time, namely *ConductionTime*, impulse propagation path *ConductionPath* and impulse propagation velocity *ConductionSpeed*. Static properties are defined in the context model to specify the electrical impulse propagation network of the heart system, the impulse propagation time for each landmark node and the impulse propagation speed for every path. Paths are represented by a set of pairs of landmark nodes (see Definition 1, Properties 1 and 2 and Tables 8.1 and 8.2).

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

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

8.5.2 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 a dynamic behaviour of the heart system through the step-wise impulse propagation into the 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 a function, which shows boolean states of the landmark nodes. When the electrical impulse passes through the landmark nodes (see Fig. 8.4(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 a boolean state *TRUE* or *FALSE*. *TRUE* represents the normal condition of the heart while *FALSE* represents an abnormal condition of the heart.

```
$inv1 : ConductionNodeState \in ConductionNode \rightarrow BOOL$
$inv2 : CConductionTime \in ConductionNode \rightarrow 0..300$
$inv3 : CConductionSpeed \in ConductionPath \rightarrow 0..500$
$inv4 : HeartState \in BOOL$
```

In the abstract specification of the heart model, there are three events, namely *HeartOK* to represent a normal state of the heart, *HeartKO* to express abnormal state of the heart and *HeartConduction* to update the value of each landmark node 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. The first guard *grd*1 states that all landmark nodes should be visited in a single cycle of impulse propagation. The second guard states that the current impulse propagation time of each landmark node should lie within the pre-specified range of the impulse propagation times. The final guard states that the current impulse propagation velocity of each path should lie between pre-defined impulse propagation velocities. If all guards are satisfied then the heart state indicates the normal condition as being *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 to those for the normal state of the heart system to specify abnormal conditions of the heart. These guards state that if any landmark node is not visited in a single cycle of the impulse propagation, or if any the current impulse propagation time of any landmark node does not lie within the pre-specified range of the impulse propagation times, or if the current impulse propagation velocity of any path does not lie within the pre-defined range of impulse propagation velocities, then the heart system is in an abnormal state represents by its normal condition being *FALSE*. Different kinds of heart diseases affect the electrical impulse propagation time and velocity in the heart system [24]. These changes affect the actual heart rhythm and help to identify the possible abnormal behaviours of the heart.

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

The event *HeartConduction* formalises the heart behaviour in an abstract manner by updating the values for impulse propagation time, impulse propagation velocity and visited state of the landmark nodes non-deterministically. This event is used to model more concrete behaviour of the heart system at the next level of refinement.

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

8.5.3 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 fibres into ventricles (G, H nodes) and in the left atria muscle fibres (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 behaviour of a sinoatrial (SA) node, which originates electrical impulse for traversing throughout the heart system using the conduction network (see Fig. 8.4). The 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 a 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 fibres of ventricles (G, H nodes) and left atria muscle (node C). This event resets all variables for generating next impulse at the SA node. The actions of this 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 sets 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
```

8.5.4 Refinement 2: Impulse Propagation

In the second refinement, we introduce several events as a refinement of the event *HeartConduction* to model the impulse propagation into the heart conduction network. New events formalise 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 modelling 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 a time interval for the single heart beat, which may change in every cycle of an 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 (*inv*4–*inv*10) are introduced as safety properties, which define that if the heart state is *TRUE* then the impulse propagation time and the impulse propagation velocity be within the standard range of time and velocity during the impulse conduction throughout the conduction network (see Fig. 8.4(b)).

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

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

Events are introduced in this refinement to model the impulse propagation from SA node towards the Purkinje fibres landmark nodes (G, H) and atria fibres nodes (C). Each event is synchronised through progressive electrical impulse propagation in the conduction network. We have given formalisation of only one event *HeartConduction_A_B* to understand the basic formalisation steps of all other events. All other events of impulse propagation in the conduction network among landmark nodes have been modelled 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 node. The progressive increment of the independent logical clock is modelled 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 all the 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 defined 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.

8.5.5 Refinement 3: Perturbation in the Conduction

It introduces a set of possible blocks in the heart conducting system. These blocks can occur in the conduction network and give trouble in 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.

$$\boxed{\begin{array}{l} axm1 : partition(HeartBlockSets, \{SA\_nodal\_blocks\}, \{AV\_nodal\_blocks\}, \\ \qquad \{Infra\_Hisian\_blocks\}, \{LBBB\_blocks\}, \{RBBB\_blocks\}, \{None\}) \end{array}}$$

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 the 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 depolarising the atria. There are different kinds of SA blocks [24, 30]. To model SA block, we introduce an event *HeartConduction_Block_A_B_C*, which formalises the SA block. In this event, guard (*grd*1) 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 lain within the standard range, or the current impulse propagation velocity of the pairs $A \mapsto B$ and $A \mapsto C$ are not lain within the standard range. When a 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 atria, atrioventricular node, bundle of His, or Purkinje system. It is a type of heart block in which a 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 at 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 [24, 30]. To model the AV block, we introduce an event *HeartConduction_Block_B*, which formalises the AV block. The conduction node state *ConductionNodeState* of a landmark node (B) is *FALSE*, which represents a condition for the AV block using guard (grd1) and actions state that the heart state is *FALSE* and such kind of heart block is known as the 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 a block of the distal conduction system (node D). There are different kinds of Infra-Hisian blocks [24, 30]. 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 the current impulse propagation time of a node D is not lain within the standard range, or the current propagation velocity of a pair $B \mapsto D$ is not lain within the standard range. The 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 : $(ConductionNodeState(D) = FALSE) \lor$
           $(CConductionTime(D) \notin ConductionTime(D)) \lor$
           $(CConductionSpeed(B \mapsto D) \notin ConductionSpeed(B \mapsto D))$
  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 the 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 [24, 30]. 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; 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 lain within the standard ranges, or the current impulse propagation velocity of the pairs $D \mapsto E$ and $E \mapsto G$ are not lain 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 : $(ConductionNodeState(E) = FALSE) \lor$
           $(ConductionNodeState(G) = FALSE) \lor$
           $(CConductionTime(E) \notin ConductionTime(E)) \lor$
           $(CConductionTime(C) \notin ConductionTime(C)) \lor$
           $(CConductionSpeed(D \mapsto E) \notin ConductionSpeed(D \mapsto E)) \lor$
           $(CConductionSpeed(E \mapsto G) \notin ConductionSpeed(E \mapsto G))$
  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. 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 lain within the standard range, or the current impulse propagation velocity of the pairs $D \mapsto F$ and $F \mapsto H$ are not lain 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 : $(ConductionNodeState(F) = FALSE) \vee$
           $(ConductionNodeState(H) = FALSE) \vee$
           $(CConductionTime(F) \notin ConductionTime(F)) \vee$
           $(CConductionTime(H) \notin ConductionTime(H)) \vee$
           $(CConductionSpeed(D \mapsto F) \notin ConductionSpeed(D \mapsto F)) \vee$
           $(CConductionSpeed(F \mapsto H) \notin ConductionSpeed(F \mapsto H))$
  THEN
    act1 : $HeartState := FALSE$
    act2 : $HeartBlocks := LBBB\_blocks$
  END
```

8.5.6 Refinement 4: Getting a Cellular Model

This last refinement introduces cellular level modelling into the heart model. The cellular level modelling is used to model the electrical impulse propagation at the cell level. The formalisation uses cellular automata theory to model the micro-structure based cell model. To formalise the cellular automata, we introduce mathematical properties (see Definitions 2 and 3) in a context model. In a 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 a 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 the power-set of *NeighbouringCells* to a 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 behaviour 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. 8.9). 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 the *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, the last property ($axm10$) presents that if a cell at (x, y) is *Passive*, then and if all the neighbouring cells in 2D is not *Active*, then a set of neighbouring cells must be in *Passive*.

$$\boxed{\begin{array}{l}
axm7 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \wedge CellS(\{x, y\}) = ACTIVE \\
\qquad\qquad \Rightarrow NEXT(param) = REFRACTORY \\
axm8 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \wedge CellS(\{x, y\}) = \\
\qquad\qquad REFRACTORY\ \Rightarrow NEXT(param) = PASSIVE \\
axm9 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \wedge \{x, y\} \in param \wedge \\
\qquad\qquad CellS(\{x, y\}) = PASSIVE\ \Rightarrow ((CellS(\{x+1, y\}) = ACTIVE \vee \\
\qquad\qquad CellS(\{x-1, y\}) = ACTIVE\ \vee CellS(\{x, y+1\}) = ACTIVE \vee \\
\qquad\qquad CellS(\{x, y-1\}) = ACTIVE)\ \Rightarrow NEXT(param) = ACTIVE) \\
axm10 : \forall\, param \cdot param \in \mathbb{P}(NeighbouringCells) \wedge \{x, y\} \in param \wedge \\
\qquad\qquad CellS(\{x, y\}) = PASSIVE\ \Rightarrow ((CellS(\{x+1, y\}) \neq ACTIVE \wedge \\
\qquad\qquad CellS(\{x-1, y\}) \neq ACTIVE \wedge CellS(\{x, y+1\}) \neq ACTIVE \wedge \\
\qquad\qquad CellS(\{x, y-1\}) \neq\ ACTIVE \Rightarrow NEXT(param) = PASSIVE)
\end{array}}$$

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 a cell cannot be reactivated after a moment, a cell changes its state to the *Passive* state, in which the cell awaits next impulse (see Fig. 8.9).

To model the dynamic behaviour 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 behaviour of a tissue. Last variable *NextCellState* is used to store the values of next neighbouring positions after every transition.

$$\boxed{\begin{array}{l}
inv1 : m \in \mathbb{Z} \\
inv2 : n \in \mathbb{Z} \\
inv3 : Transition \in BOOL \\
inv4 : NextCellState \in CellStates
\end{array}}$$

To implement the dynamic behaviour of a cell in two-dimensions, we introduce two events *HeartConduction_Cellular* to make transition TRUE for the electrical conduction at the cell level and *HeartConduction_Next_UpdateCell* to calculate status of the 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*. The 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. The 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 the state of neighbouring cells and to update the position of the current cell (m, n). The first guard of this event represents a set of coordinate positions (*param*) of neighbouring cells and the next guard (grd2) states that the selected neighbouring cells are a set of cells $(dom(NEXT))$. The last guard presents a transition state *Transition* is *TRUE*. Action of this event calculates a set of the next neighbouring cells in act1. The next action (act2) sets *FALSE* of a transition state. The last two actions update the value of the current cell (m, n) to continuously impulse propagating in the 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 modelling. In the next section, we present model validation of the heart model using Event-B model checker ProB tool.

Table 8.3 Proof statistics

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 %)

8.5.7 Model Validation and Analysis

There are two main validation activities in Event-B, and both are complementary for designing a consistent system in the medical domain; *consistency checking* and *model analysis*. This section validates the model by using ProB tool [26] 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 that supports *automated consistency checking* of Event-B machines via model checking [11] and constraint-based checking [18]. 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 various scenario cases of normal and abnormal heart conditions, and we have also tested morphological behaviour [3, 6] of the ECG during impulse propagation from the SA node (A) to the Purkinje fibres (F, H) in the ventricles. The logic-based mathematical model of the heart can generate all possible scenarios of normal and abnormal heart conditions in the ECG caused by changes in time and velocity among landmark nodes. ProB was very useful in animating all models and in verifying the absence of error (no counter-examples exist) and deadlock.

Table 8.3 expresses the proof statistics of the development using the Rodin tool. These statistics measure the size of the model, the proof obligations generated and discharged by the Rodin prover and those are interactively proved. The complete development of the heart model results in 218 (100 %) proof obligations, within which 194 (89 %) are proved automatically by the Rodin tool. The remaining 24 (11 %) proof obligations are proved interactively using Rodin tool. For the heart model, many proof obligations are generated because of the introduction of the new functional behaviours. To guarantee the correctness of these functional behaviours, we have established various invariants in the incremental refinements. Most of the proofs are interactively discharged in the third refinement of the heart model. These proofs are quite simple, and have been discharged with the help of simplifying predicates. Few proof obligations are proved interactively in other refinements. The incremental refinement of the heart system helps to achieve a high degree of automatic proof.

8.6 Discussion

This chapter presents a methodology for modelling a biological system, such as the heart, by modelling a biological environment. The main objective of this methodology is to model the heart system and integrate it with the model of a medical device such as a cardiac pacemaker, thereby modelling the closed-loop system to enable certification of the medical system via the certification bodies [17, 23] for safe operation. To build a closed-loop model using both environment and device modelling is considered as a standard approach in the area validation, given that designing an environment model is a challenging problem in the real world. Industry has long sought such an approach to validating system models in a biological environment. We have discovered much information via a literature survey and long discussions with experts in cardiology and physiology, and have concluded how best to model the heart system as a cellular-level architecture in an efficient and optimum way. Because of the complexity of the cellular-level calculations (see Sect. 8.2), previous models have failed to model the heart system.

We have proposed modelling the heart in an abstract way to simulate the desired behaviour of the heart system while avoiding the complexity. More importantly, the heart model is based on logico-mathematical theory. Our primary objective was to model the heart system using only simple logico-mathematical methods. The heart model is an environmental model for medical devices that may improve their development in the early phases. As such, it will contribute only one element of the verification process. Other verification steps will also be required. Medical experts have elaborated every minor detail in an effort to understand the complexity of the biological system, particularly because the heart system is the most complex organ in the body. The proposed approach contains only a main part of the specification of the system behaviour, with the remaining information being hidden. We have spent much time identifying an exact abstract model of the heart system that satisfies medical experts. We have used the Event B modelling language to model and verify the system. The ProB model checker was used to verify the correctness of the heart model via animation. Any other formal specification language and model checker could be used to model the heart system based on our proposed methodology.

8.7 Summary

This chapter has presented a methodology for producing a mathematical model of the heart based on logico-mathematical theory [33–35]. This model is the first computational model that considers the heart as an electrical conduction system. Given that a cardiac pacemaker interacts with the heart exactly at this level (i.e., electrical impulses), this model is a very promising “environmental model” to be used in parallel with a pacemaker model to form a closed-loop system. This model therefore has an immediate use in “the grand challenges in formal methods” where an industrial pacemaker specification has been elected as a benchmark. To formalise the heart system, we have used the Event-B modelling language [1, 38] to develop the proof-based formal model. Our approach involves formalising and reasoning about impulse propagation in the whole heart system through the conduction network (see Fig. 8.4(a)). More precisely, we would like to stress the original contribution of our work. We have proposed a method for modelling a human heart based on logico-mathematical theory. The main objectives of this proposed idea are as follows:

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

In summary, we have formalised the known characteristics and physiological behaviour of the heart. The formalisation highlights various aspects of the problem, making different assumptions about impulse propagation and establishing different properties related to the CA. We have outlined how an incremental refinement approach to the heart system enables a high degree of automatic proof using the Rodin tool. Our various developments reflect not only many facets of the problem, but also the learning process involved in understanding the problem and its ultimate possible solutions.

The consistency of our specification has been checked through reasoning, and validation experiments were performed using the ProB model checker with respect to safety conditions. As part of our reasoning, we have proved that the initialisation of the system is valid, and we have calculated the preconditions for operations. These have been executed to guarantee that our intention to have total operations has been fulfilled. At each stage of the refinement, we have introduced a new behaviour for the system and proved its *consistency* and *refinement checking*. We have introduced more general invariants at the refinement level, showing that the initialisation of the whole system is valid. Finally, we have validated the heart system using the ProB model checker as a validation tool and have verified the correctness of the exact behaviour of our heart system with the help of physiology and cardiology experts.

References

1. Abrial, J.-R. (2010). *Modeling in Event-B: System and software engineering* (1st ed.). New York: Cambridge University Press.
2. Adam, D. R. (1991). Propagation of depolarization and repolarization processes in the myocardium—an anisotropic model. *IEEE Transactions on Biomedical Engineering*, *38*(2), 133–141.
3. Artigou, J. Y., & Monsuez, J. J. (2007). *Cardiologie et maladies vasculaires*. Paris: Elsevier Masson.
4. Back, R. J. R. (1981). On correct refinement of programs. *Journal of Computer and System Sciences*, *23*(1), 49–68.
5. Barold, S. S., Stroobandt, R. X., & Sinnaeve, A. F. (2004). *Cardiac pacemakers step by step*. London: Futura. ISBN 1-4051-1647-1.

6. Bayes, B. V. N., de Luna, A., & Malik, M. (2006). The morphology of the electrocardiogram. In *The ESC textbook of cardiovascular medicine* (pp. 1–36). Oxford: Blackwell.
7. Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., & Yi, W. (1996). UPPAAL—a tool suite for automatic verification of real-time systems. In *Proceedings of the DIMACS/SYCON workshop on hybrid systems III: Verification and control* (pp. 232–243). Secaucus: Springer.
8. Berenfeld, O., & Abboud, S. (1996). Simulation of cardiac activity and the ECG using a heart model with a reaction-diffusion action potential. *Medical Engineering & Physics*, *18*(8), 615–625.
9. Bowen, J., & Stavridou, V. (1993). Safety-critical systems, formal methods and standards. *Software Engineering Journal*, *8*(4), 189–209.
10. CDRH (2006). Safety of marketed medical devices. Center for Devices and Radiological Health, US FDA.
11. Clarke, E. M., Grumberg, O., & Peled, D. (2001). *Model checking*. Cambridge: MIT Press.
12. Ellenbogen, K. A., & Wood, M. A. (2005). *Cardiac pacing and ICDs* (4th ed.). Oxford: Blackwell. ISBN 1-4051-0447-3.
13. Fitzgerald, J. (2007). The typed logic of partial functions and the Vienna development method. In D. Bjørner & M. C. Henson (Eds.), *EATCS textbook in computer science*. *Logics of specification languages* (pp. 431–465). Berlin: Springer.
14. Fitzgerald, J., Larsen, P. G., Pierce, K., Verhoef, M., & Wolff, S. (2010). Collaborative modelling and co-simulation in the development of dependable embedded systems. In *Lecture notes in computer science*. *Proceedings of the 8th international conference on integrated formal methods* (pp. 12–26). Berlin: Springer.
15. Harrild, D. M., & Henriquez, C. S. (2000). A computer model of normal conduction in the human atria. *Circulation Research*, *87*, 25–36.
16. Hesselson, A. (2003). *Simplified interpretations of pacemaker ECGs*. Oxford: Blackwell. ISBN 978-1-4051-0372-5.
17. High Confidence Software and Systems Coordinating Group (2009). *High-confidence medical devices: Cyber-physical systems for 21st century health care* (Technical report). NITRD. http://www.nitrd.gov/About/MedDevice-FINAL1-web.pdf.
18. Jackson, D. (2002). Alloy: A lightweight object modelling notation. *ACM Transactions on Software Engineering and Methodology*, *11*(2), 256–290.
19. Jee, E., Wang, S., Kim, J.-K., Lee, J., Sokolsky, O., & Lee, I. (2010). A safety-assured development approach for real-time software. In *16th IEEE international conference on embedded and real-time computing systems and applications*, RTCSA (pp. 133–142).
20. Jetley, R. P., Carlos, C., & Purushothaman Iyer, S. (2004). A case study on applying formal methods to medical devices: Computer-aided resuscitation algorithm. *International Journal on Software Tools for Technology Transfer*, *5*(4), 320–330.
21. Jetley, R., Purushothaman Iyer, S., & Jones, P. (2006). A formal methods approach to medical device review. *Computer*, *39*(4), 61–67.
22. Jiang, Z., Pajic, M., Connolly, A. T., Dixit, S., & Mangharam, R. (2010). Real-time heart model for implantable cardiac device validation and verification. In *22st Euromicro conference on real-time systems*, IEEE ECRTS'10, July 2010.
23. Keatley, K. L. (1999). A review of the FDA draft guidance document for software validation: Guidance for industry. *Quality Assurance*, *7*(1), 49–55.
24. Khan, M. G. (2008). *Rapid ECG interpretation*. Clifton: Humana Press.
25. Lee, I., Pappas, G. J., Cleaveland, R., Hatcliff, J., Krogh, B. H., Lee, P., et al. (2006). High-confidence medical device software and systems. *Computer*, *39*(4), 33–38.
26. Leuschel, M., & Butler, M. (2003). *Lecture notes in computer science*. *ProB: A model checker for B* (pp. 855–874). Berlin: Springer.
27. Love, C. J. (2006). *Cardiac pacemakers and defibrillators*. Georgetown: Landes Bioscience. ISBN 1-57059-691-3.
28. Maisel, W. H., Sweeney, M. O., Stevenson, W. G., Ellison, K. E., & Epstein, L. M. (2001). Recalls and safety alerts involving pacemakers and implantable cardioverter-defibrillator generators. *Journal of the American Medical Association*, *286*(7), 793–799.

29. Makowiec, D. (2008). The heart pacemaker by cellular automata on complex networks. In *Proceedings of the 8th international conference on cellular automata for research and industry*, ACRI'08 (pp. 291–298). Berlin: Springer.
30. Malmivuo, J. (1995). *Bioelectromagnetism*. Oxford: Oxford University Press. ISBN 0-19-505823-2.
31. Méry, D., & Singh, N. K. (2010). Real-time animation for formal specification. In M. Aiguier, F. Bretaudeau, & D. Krob (Eds.), *Complex systems design & management* (pp. 49–60). Berlin: Springer.
32. Méry, D., & Singh, N. K. (2010). Trustable formal specification for software certification. In T. Margaria & B. Steffen (Eds.), *Lecture notes in computer science: Vol. 6416. Leveraging applications of formal methods, verification, and validation* (pp. 312–326). Berlin: Springer.
33. Méry, D., & Singh, N. K. (2011). Technical report on formalisation of the heart using analysis of conduction time and velocity of the electrocardiography and cellular-automata. MOSEL-LORIA-INRIA-CNRS: UMR7503-Université Henri Poincaré-Nancy I-Université Nancy II-Institut National Polytechnique de Lorraine. http://hal.inria.fr/inria-00600339/en/.
34. Méry, D., & Singh, N. K. (2012). Closed-loop modeling of cardiac pacemaker and heart. In *Foundations of health informatics engineering and systems*.
35. Méry, D., & Singh, N. K. (2012). Formalization of heart models based on the conduction of electrical impulses and cellular automata. In Z. Liu & A. Wassyng (Eds.), *Lecture notes in computer science: Vol. 7151. Foundations of health informatics engineering and systems* (pp. 140–159). Berlin: Springer.
36. Plonsey, R., & Barr, R. C. (1987). Mathematical modeling of electrical activity of the heart. *Journal of Electrocardiology*, *20*(3), 219–226.
37. Seong, Y. R., Jun, K.-R., & Kim, T. G. (1994). A cellular automata model of activation process in ventricular muscle. In *SCSC'94* (pp. 769–774).
38. RODIN (2004). Rigorous open development environment for complex systems. http://rodin-b-sharp.sourceforge.net.
39. Vangheluwe, H., & Vansteenkiste, G. C. (2000). 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* (pp. 800–810). Ghent: SCS Europe.
40. von Neumann, J. (1966). *Theory of self-reproducing automata*. Chicago: University of Illinois Press. A. W. Burks (Ed.).