Pacemaker's Functional Behaviors in Event-B

Dominique Méry, Neeraj Kumar Singh

Pacemaker's Functional Behaviors in Event-B

Dominique Méry and Neeraj Kumar Singh

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

Abstract. Test and Simulation are the only verification techniques used for any biomedical devices such as pacemaker system, implantable cardioverter/defibrillators (ICDs) etc. The construction of formal models of Pacemaker systems is a considerable practical challenge. Formal modeling of an artificial Pacemaker system is a case study proposed by the software quality research laboratory at McMaster University[1] in the Grand Challenge Initiative. Using an incremental proof-based approach, we model functionalities of the Pacemaker. The approach is illustrated by developing a new formal model of the cardiac pacemaker system. Our contribution are in this report to model the single electrode pacemaker system using Event-B and prove it. The incremental proof-based development is mainly driven by the refinement between an abstract model of the system and its detailed design through a series of refinements. A series of refinements is progressively added the functional and the timing properties to the abstract system-level specifications using some intermediate models. The properties express system architecture, action-reaction and timing behavior. This paper uses all possible operational modes of a single electrode Pacemaker system that helps to develop better hardware. Every stage of refinement includes the detail information about operating modes. The models are expressed in Event-B modeling language and validated primarily by the ProB tool in different situation such as hysteresis and rate adapting pacing under real-time constraints. In each stages of refinements include the detail information and more events are introduced. The final step of refinement completely localized the events and similar to implementation of single electrode pacemaker operating modes system. The stepwise refinement of the single electrode Pacemaker system contributes to achieve a high degree of automatic proof.

Key words: Abstract model, Event-B, Event-driven approach, Proof-based development, Refinement, Pacemaker, Electrode, Heart

1 Introduction

For centuries, people became ill and died. Today, with our increasing ability in medical technology to treat most illnesses, it's improves the human life as much as possible. People can be kept alive with mechanical supports almost

[1] http://www.cas.mcmaster.ca/wiki/index.php/Pacemaker

indefinitely. New medical technology innovation enhances the quality and effectiveness of care. Billions of patients worldwide depend on medical technology at home, at the doctor's, at hospital and in nursing homes. Development and production of medical device software and systems are common crucial issues [1] for ensuring safe advances in healthcare. As devices becomes increasingly smaller in physical terms but larger in software terms, the design, testing and device approval is becoming more expensive for medical device manufactures both in term of times and cost. The number of devices that have been recently been recalled due to software and hardware problems is increasing at an alarming rate [2].

Pacemaker is one of the hardware device for controlling the heart rate in medical domain. In a Pacemaker system, the firmware controls the hardware such that an adequate heart rate is maintained, which is necessary either because the heart's native Pacemaker is insufficiently fast or there is a block in the heart's electrical conduction system [3, 4, 2].

An adequate heart rate is required to ensure that the heart pumps enough blood into the body, according to the patient's activity level. To accomplish this adequate heart rate, the pacemaker is connected to the chambers of the heart by electrodes (pacemaker leads). Via these leads, electrical stimuli (paces) can be delivered to force a contraction of the heart's muscle cells. The pacemaker also retrieves information from the heart via these leads; contractions of the heart are measured and result in input signals on the pacemaker leads. Together with hardware notifications that indicate a delivered pace, these are the main inputs for the pacemaker firmware, which calculates whether the current heart rate is appropriate on every input signal. These calculations always result in commanding the hardware to deliver a pace to the heart on a certain moment in time. The planned pace will ensure a heart contraction and prevents that the heart stops pumping blood into the body if it does not contract spontaneously anymore.

Formal methods are based on solid mathematical principles and increase understanding of systems, increase clarity of descriptions and help solve problems and remove errors. The use of formal methods for software and hardware designs is motivated by the expectation that performing appropriate mathematical analyses can contribute to the reliability and robustness of these designs.

Only simulation and testing are usual validation techniques for the Pacemaker system specification. This is an operational way to check whether a given system realization confirms to an abstract specification. By nature, testing can be applied only after a prototype implementation of the system has been realized. Formal verification, as opposed to testing, works on models (rather than implementation) and amounts to mathematical proof of correctness of a system. Modeling in the field of medical domain is a grand challenge and Pacemaker system specification [5] is one of them in Verified Software [1, 6] that is proposed by the software quality research laboratory at McMaster University as a pilot problem. The challenge is characterised by system aspects including hardware requirements and safety issues. Such a system demands high integrity to achieve safety requirements.

In order to analyze the problem, we consider the triptych by D. Bjoerner [7],

where, $\mathcal{D}, \mathcal{S} \rightarrow \mathcal{R}$
$\mathcal{D}$ = Healthcare domain
$\mathcal{S}$ = Model or chain of models of the Pacemaker system
$\mathcal{R}$ = Requirements of Pacemaker system

$\mathcal{D}$ is the context of the problem to solve and it defines in Event-B (parameters, constants etc.). $\mathcal{S}$ is the Pacemaker and the Heart systems. $\mathcal{R}$ is the requirements of the heart system such that when Pacemaker delivers a pacing stimulus to the heart and sensing the intrinsic activity from the heart. The operating modes of Bradycardia Therapy and formal model of Pacemaker system are based on informal requirements, which are given by Boston Scientific [5].

We have found a case study on distributed real-time model of a cardiac pacing system that is developed by H.D. Macedo, et al. [6]. Similarly, in other case study V.P. Manna, et al. [8] have developed a simple Pacemaker implementation [8]. Here we present stepwise development to model and verify such interdisciplinary requirements in Event-B [9, 10] modeling language. The correctness of each step is proved in order to achieve a reliable system. The Pacemaker models must be validated to ensure that they meet requirements for the Pacemaker. In order to handle this, validation must be carried out both by formal modelling experts (to ensure the models are free of inconsistent behaviour) and by domain experts (to ensure the models produce the appropriate behaviour for any given situation). However, some difficulty may arise due to the disparate nature of the areas of expertise these two groups possess.

In the block diagram (see Figure-1) represents the basic interface model of Pacemaker system and Heart. Our abstract specification includes events modelling pacing modes of single electrode pacemaker. The nature of the refinement that we verified using RODIN [11] proof tools are safety refinement and any behaviour (trace of events) of a refined model must be behaviours of the abstract model. Thus, since a behaviour which results in pacing and sensing mode of single electrode pacemaker with real time constrain inside the human heart is preserving by the abstract model, it is also preserved in a correctly refined models. Our refinements break the atomicity of pacemaker modes into several small process or events. Proof-based development methods [12–14] integrate formal proof techniques in the development of software systems. The main idea is to start with a very abstract model of the system under development. Details are gradually added to this first model by building a sequence of more concrete events. The relationship between two successive models in this sequence is that of *refinement* [12, 13]. The essence of the refinement relationship is that it preserves already proved system properties including safety properties and termination.

A development gives a number of proof obligations, which guarantee its correctness. Such proof obligations are discharged by the proof tool using automatic and interactive proof procedures supported by a proof engine [15, 11]. At the most abstract level it is obligatory to describe the static properties of a model's data by means of an invariant's predicate. This gives rise to proof obligations

relating to the consistency of the model. They are required to ensure that data properties which are claimed to be invariant are preserved by the events of the model. Each refinement step is associated with a further invariant which relates the data of the more concrete model to that of the abstract model and states any additional invariant properties of the (possibly richer) concrete data model. These invariants, so-called gluing invariants are used in the formulation of the refinement proof obligations.

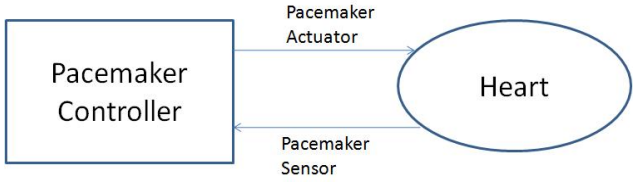


Figure-1 : Pacemaker and Heart Interface

The goal of a Event B development is to obtain a proved model and to implement the correctness-by-construction [16] paradigm. Since the development process leads to a large number of proof obligations, the mastering of proof complexity is a crucial issue. Even if a proof tool is available, its effective power is limited by classical results over logical theories and we must distribute the complexity of proofs over the components of the current development, e.g. by refinement. Refinement has the potential to decrease the complexity of the proof process, while allowing for traceability of requirements. The price to pay is to face possibly complex mathematical theories stored in contexts and difficult proofs. The re-use of developed models and the structuring mechanisms available in B help in decreasing the complexity, especially the cut and paste of proofs. These solutions are, de-facto, supporting the use of patterns which are stating general proof-based developments and are validating the expression proof-based patterns.

The current work intends to explore problems related to the modelling of pacing and sensing modes of one electrode pacemaker in real time constrain and to evaluate the refinement process. More-over, the stepwise development of pacemaker model helps us to discover the exact behaviour of pacing and sensing behaviour of pacemaker inside the heart.

1.1 Organization of the report

The outline of the remaining report is as follows. Section 2 describes the basic overview of Pacemaker, which is outlining some general ideas of artificial Pacemaker and heart. In section 3, we describe Event-B, the action-reaction and the real-time patterns, to know the formal development of Pacemaker system with specified patterns. Section 4 explores the formal development of stepwise refinement by the Pacemaker model. Before section 5 is ended, models are validated by the ProB tool [17] and are analyzed by statistical proof. Finally, section 6 concludes the report.

2 Basic Overview of Pacemaker System

The Pacemaker system is a small electronic device for helping the heart to maintain the regular heart beat, when it is irregular. The Pacemaker is implanted in the chest during surgery. Wires called leads are put into the heart muscle. The device with the battery is placed under the skin, below the shoulder. The Pacemaker is always operating inside the heart environment so that we first review the basic elements of the heart system and then describe the essential elements of Pacemaker system.

2.1 The Heart System

The human heart is wondrous in its ability to pump for the circulatory system continuously throughout a lifetime. The heart's mechanical system (the pump) requires at the very least impulses from the electrical system. The heart consists of four compartments:the right and left atria and ventricles, which contract and relax periodically under the control of natural electrical stimuli. The atria form one unit and the ventricles another. The left ventricular free wall and the septum are much thicker than the right ventricular wall. This is logical since the left ventricle pumps blood to the systemic circulation, where the pressure is considerably higher than for the pulmonary circulation, which arises from right ventricular outflow. In the normal functioning of natural pacemaker or heart, a discharge is made at the sinus node; the discharge subsequently reaches the atrioventricular (AV) node which amplifies it, stimulating the ventricles. If the natural pacemaker is malfunctioning, a physical condition termed Bradycardia may arise in which the heart rate falls below the level expected for the patient [3]. To normalize the heart rate, an artificial pacemaker may be implanted to help the heart. The beats per minute (bpm) is a basic unit to measure the rate of the heart activity.

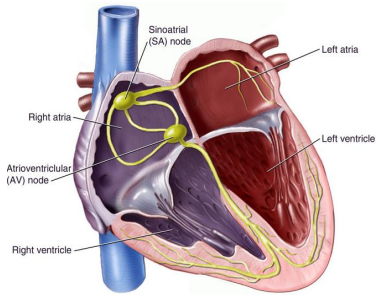


Figure-2 :Heart or Natural Pacemaker

2.2 The Pacemaker

A pacemaker is an electronic device implanted in the body to regulate the heart beat. The basic elements of Pacemaker system [18, 19] are,

Leads: One or more flexible coiled metal wire normally two, that transmits electrical signals between the heart and the Pacemaker. Each Pacemaker lead is classified by whether it is configured with one ("unipolar") or two ("bipolar") separate points of electrical contact within the heart.

The Pacemaker Generator: The Pacemaker is both the power source and the brain of the pacing and sensing systems. Such as, it contains an implanted batteries and controller as an electronic circuitry.

Device Controller-Monitor (DCM) or Programmer: An external unit that interacts with the Pacemaker device using a wireless connection. It consists of hardware platform and the Pacemaker application software.

Accelerometer: It is an electromechanical device inside the Pacemaker that measures the body motion or acceleration of motion of a body in order to allow modulated pacing.

Figure-3 :Artificial Pacemaker

In the single electrode Pacemaker, the electrode is attached to the right atrium or the right ventricle. It has several operational modes that regulate the heart functioning. The specification document [5] describes all possible operating modes for controlling the different parameters of the Pacemaker. Most of the parameters are related to real-time and action-reaction constraints for controlling the heart rate.

Category	**Chambers Paced**	**Chambers Sensed**	**Response to Sensing**	**Rate Modulation**
Letters	**O**-None **A**-Atrium **V**-Ventricle **D**-Dual(A+V)	**O**-None **A**-Atrium **V**-Ventricle **D**-Dual(A+V)	**O**-None **T**-Triggered **I**-Inhibited **D**-Dual(T+I)	**R**-Rate Modulation

Table-1 : Bradycardia operating modes of Pacemaker system

Pacemaker function mode is characterized by an universally accepted code consisting of three or four characters. The code provides a description of Pacemaker pacing and sensing functions using a four-letter sequence. The sequence is referred to as the "Pacemaker mode"(see Table-1). In practice, only the first three or four-letter positions are commonly used to describe bradycardia pacing functions. The first letter of the code sequence represents that the chamber

being paced, second letter of the code sequence for the chamber being sensed, third letter of code sequence for responding to sense and final optional letter of code sequence indicates the presence of rate modulation in response to the physical activity as measured by the accelerometer. "X" is a wildcard used to denote any letter (i.e. "O", "A", "V" or "D"). *Triggered* (T) refers to deliver a pacing stimulus and *Inhibited* (I) refers to be inhibited from further pacing after sensing of an intrinsic activity from the heart chamber.

3 Event-B Patterns

The purpose of a design pattern [20] 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 is the possibility to reuse solutions from earlier developments in the current project. This will lead to a correct refinement in the chain of models, without arising proof obligations. Since the correctness (i.e proof obligations are proved) of the pattern has been proved during its development, nothing is to be proved again when using this pattern.

Pacemaker systems are characterized 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. So, the most common elements in Pacemaker system are bounded time interval for every action, reaction and action-reaction pair. The action-reaction within a time limit can be viewed as an abstraction of the Pacemaker system. We recognize the following two design patterns when modeling this kind of system according to the relationship between the action and corresponding reaction.

3.1 Action-Reaction Pattern

Under action-reaction chapter [10] two basic types of design patterns are,

Action and Weak Reaction: Once an action emits, a reaction should start in response. For a quick instance, if an action stops, the reaction should follow. Sometimes reaction does not change immediately according to the action. This is so-called pattern of action and weak reaction.

Action and Strong Reaction: The action and reaction can always keep proper synchronization then this behavior of action-reaction is known as pattern of action and strong reaction.

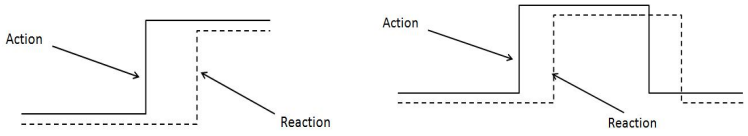


Figure-4 :Action-Reaction Patterns

Composite Weak and Strong Reactions: Action and Reaction (weak or strong) are only the basic blocks for modeling discrete event system. In most cases, system to be modeled has some complex situations to handle, because functions of a large complex system depend on some sequences of events, in which some events may be of action-reaction relation and some may occur simultaneously. The interaction between two action-reaction blocks can be modeled as composite or synchronization, which depend on that the two blocks are of weak-strong reactions or strong-strong reactions. When the weak reaction of a specific action-reaction block results eventually in the specific strong reaction of some action-reaction, it can be recognized as the composite for weak and strong reactions.

Weak synchronization of two strong reactions: As far as the synchronization of two strong action-reaction blocks is concerned, two kinds of synchronizations could be identified, which can be recognized as weak synchronization and strong synchronization. The second strong reaction can be set in on state when the first strong reaction already in on state, but there is not any constrain for how many times the first strong reaction is set to on state and what will be state of first strong reaction after the off state of second strong reaction. This is what we called weak synchronization of two strong reactions.

Strong synchronization of two Strong reactions: Another kind of synchronization between two strong action-reaction blocks is so-called strong synchronization of two strong reactions. In this pattern given the solution of the problem with the weak synchronization of two strong reactions. The strong synchronization between two strong action-reaction blocks really means that the second reaction will strictly run after the first reaction , which reacts to the first action a and changes its value into on or off regularly.

Above action-reaction patterns are the refinements of weak action-reaction patterns [21].

3.2 Time-Based Pattern

The action-reaction events of a Pacemaker system are based on real-time pattern. D. Cansell, D. Méry and Joris Rehm have introduced the time constraint pattern in IEEE 1394 and 2-Slots Simpson Algorithm case studies [22, 23]. We have applied the same time pattern to solve the time constraint of the Pacemaker system. This time pattern 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. Here we apply the time pattern in modeling to synchronize the sensing and pacing stimulus functions of the Pacemaker system in continuous progressive time constraint. In the model every events are controled under time constraints, which means action of any event activates only when time constraint satisfies on specific time. The time progress is also an

event, so there is no modification of the underlying B language. It is only a modeling technique instead of a specialized formal system. The variable *time* is 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.

4 Formal Development

The Pacemaker system development is expressed in a general way. We describe the incremental development of action-reaction and real-time pattern based single electrode Pacemaker system. We are applying the action-reaction and real-time patterns to model the single electrode pacemaker system. Each mode of pacemaker has specific properties to control the rate of natural heart. In order to understand the basic timing of a pacemaker one must understand the terminology commonly used to describe the events that occur. All single chamber pacemakers have three basic timed events:-

Automatic Interval: The period of time between two sequential paced beats uninterrupted by a sensed beat. It is also referred to as the base pacing interval and may be converted to bpm and expressed as the base pacing rate.

Escape Interval: The period of time after a sensed event until the next paced event. The escape interval is usually the same as the automatic interval. It may be different if a feature called "hysteresis" is enabled.

Refractory Period: This is a period of time after a paced or sensed event during which the pacemaker sensing is disabled. An event occurring during a refractory period will not be sensed, or will be "tagged" by the pacemaker as a refractory sensed event and used by the device for evaluation of possible abnormal rhythms (e.g., atrial fibrillation). The reason for having a refractory period in a ventricular pacemaker is to prevent sensing of the evoked QRS and T-wave that occurs immediately after the paced event. In atrial pacemakers the refractory period also prevents sensing of the far-field R-wave or T-wave. In some devices the first part of the refractory period may be an adjustable "Blanking Period", during which no sensing at all occurs, followed by the remainder of the refractory period during which sensing occurs for diagnostic purposes only [4, 24, 18].

We contribute the stepwise hierarchical development (see Figure-5) of all possible operating modes of the Pacemaker system in five models. All operating modes in different models are separately shown for each chamber (atria or ventricular). The abstract model formalizes our system requirements whereas the subsequent models introduces all detail information for the resulting system. The hierarchical development specifies all the possible operating modes with different requirements in multiple refinements. Every refinement level shows the relationship between different refined operating modes. In the block diagram (see Figure-5) triple dots (...) represents that there is no refinement at that level. In

refinements (2 and 3) have the same operating modes with different optional parameters while in refinement 4 has new modulated operating modes.

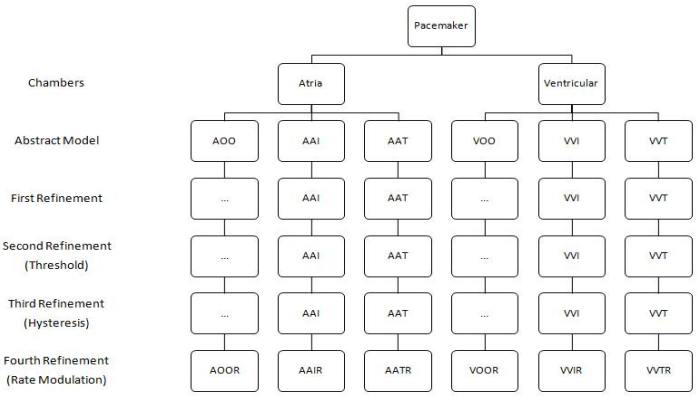


Figure-5 : Refinement Structure of Pacemaker Operating Modes

Abstract Model: In the abstract model of stepwise development of single electrode pacemaker contains the definition and properties of different time interval parameters (URL (Upper Rate Limit), LRL (Lower Rate Limit),...etc.) and pacemaker actuator status (ON and OFF). The first model contains the four basic events $Pace\_ON$, $Pace\_OFF$, tic and Set_Pace_Int, which are elementary events of single chamber pacing mode (AOO,VOO). Two extra new events $Pace\_OFF\_with\_Sensor$ and $Sense\_ON$ is introduced in single chamber pacing modes(AAI,VVI). Similarly two more events $Pace\_ON\_with\_Sensor$ and $Sense\_ON$ is introduced in single chamber pacing modes(AAT,VVT). Remaining other modes of single electrode pacemaker (AOOR, VOOR, AAIR,AATR,VVIR and VVTR) are refinement of basic single chamber pacing mode, which are describing in following continuous refinements. In the basic abstract model of pacemaker we introduce the action-reaction and real-time pattern for describing the pacing and sensing mode of single electrode pacemaker.

First Refinement: In the first refinement of the model we introduce only some extra invariants in an abstract model to stable the system and make more strong for proper pacing and sensing under real time constraints.

Second Refinement: This refinement is relatively more complex then the last refinement in which we introduce the threshold variable. A pacemaker has a stimulation threshold measuring unit which measures a stimulation threshold voltage value of a heart and a pulse generator for delivering stimulation pulses to the heart. The pulse generator is controlled by a control unit to deliver the stimulation pulses with respective amplitudes related to the measured threshold value and a safety margin. The new event Thr_value introduce to take the value of $threshold$ variable.

Third Refinement: In this refinement, we introduce the concept of $Hysteresis$ in pacing and sensing mode of single electrode pacemaker. Hysteresis, from the Greek for "to lag behind," means a delay of effect behind the cause. In pacemakers, this means delaying pacing to maximize patient benefit. The application of

a hysteresis interval to provide consistent pacing of the atrial or ventricle, or to prevent constant pacing of the atrial or ventricle.

Fourth Refinement: It is the final and last refinement of the single electrode pacemaker system. In this refinements we introduce the rate adapting pacing technique to the pacemaker. This refinement of pacemaker also give some new pacing and sensing mode (AAIR,VVIR,...etc.) of the pacemaker. The rate adapting mode of pacemaker can progressively pace faster than the lower rate, but no more than the upper sensor rate limit, when it determines that heart rate needs to increase. This typically occurs with exercise in patients that cannot increase their own heart rate. The amount of rate increase is determined by how much exertion, the pacemaker thinks the patient is performing. This increased pacing rate is sometimes referred to as the "sensor indicated rate". When exertion has stopped the pacemaker will progressively decrease the paced rate down to the lower rate.

4.1 The Context and Initial Model

In the abstract model of single electrode pacemaker, we introduce the basic notions of action-reaction and real-time constraint patterns. We have already explained the different types of action-reaction patterns. Here, in modelling of single electrode pacemaker we have applied the strong action-reaction patterns in step wise refinements in all modes. In this abstraction, we begin with an abstract model of a single electrode pacemaker system focusing on pacing and sensing modes properties and operations control the pumping rate of natural pacemaker or human heart. However, some pacing modes (AAIR,VVIR) are not distinguished in this level. Instead they are emerged to next refinements. Thus, in this level, for every modes of pacemaker are treated in the same way as common basic modes, which are essential for the single electrode pacemaker. The model consists of several modules, each corresponding to an operating mode of the pacemaker but some operating modes get through the refinements of model.

Abstraction of AOO and VOO modes We begin by defining the Event-B context. In the context, we define the constants LRL and URL that relate to the Lower Rate Limit (minimum number of pace pulses delivered per minute) and Upper Rate Limit (how fast the Pacemaker will allow the heart to be paced). These constants are extracted from the documentation [5].

$$
\begin{array}{l}
axm1 : LRL \in 30 \mathinner{..} 175 \land LRL = 60 \\
axm2 : URL \in 50 \mathinner{..} 175 \land URL = 120 \\
axm3 : URL\_Time\_Int \in \mathbb{N}_1 \land URL\_Time\_Int = 60000/URL \\
axm4 : LRL\_Time\_Int \in \mathbb{N}_1 \land LRL\_Time\_Int = 60000/LRL
\end{array}
$$

The two new constants URL_Time_Int and LRL_Time_Int represent the corresponding URI (Upper Rate Interval) and LRI (Lower Rate Interval), respectively. The numeric conversions are needed because the time unit is milliseconds. The pacemaker (or pacing) rate is programmed in milliseconds. To

convert a heart rate from beats per minute (bpm) to milliseconds, divide 60,000 by the heart rate. For example, a heart rate of 70 bpm equals 857 milliseconds. To obtain the URI and LRI, we use the $axm3$ and $axm4$. Additionaly, we define an enumerated set *status* of an electrode as ON and OFF states.

In the single electrode Pacemaker system, the Pacemaker delivers a pacing stimulus in the atria or the ventricular chambers. In our initial model, we formalize the functional behaviors of the Pacemaker system , where a new variable $Pacemaker\_Actuator$ represents the presence or absence of pulse. The variable sp (since pace) represents the current *clock counter* and a variable $last\_sp$ represents the last interval (in ms.) between two paces. The invariant ($inv4$) states that the clock counter sp should be less than Lower Rate Interval (LRI). The variable $Pace\_Int$ is an interval between two paces that is initialized by the system before start of the pacing. The invariant ($inv7$) represents the safety properties: the Pacemaker delivers a pacing stimulus into the heart chamber between URI and LRI. Similarly, next invariants ($inv8$ and $inv9$) represent the state of Pacemaker's actuator under heart environment as safety properties and state that it is never activated between two heart beats.

$$
\begin{array}{l}
inv1 : Pacemaker\_Actuator \in status \\
inv2 : sp \in \mathbb{N} \\
inv3 : last\_sp \in \mathbb{N} \\
inv4 : sp \leq LRL\_Time\_Int \\
inv5 : Pace\_Int\_set \in BOOL \\
inv6 : Pace\_Int \in URL\_Time\_Int \mathinner{..} LRL\_Time\_Int \\
inv7 : last\_sp \geq URL\_Time\_Int \wedge last\_sp \leq LRL\_Time\_Int \\
inv8 : Pace\_Int\_set = FALSE \wedge sp > 0 \wedge sp < Pace\_Int \\
\qquad \Rightarrow \\
\qquad Pacemaker\_Actuator = OFF \\
inv9 : Pace\_Int\_set = FALSE \wedge sp > Pace\_Int \\
\qquad \Rightarrow \\
\qquad Pacemaker\_Actuator = ON
\end{array}
$$

In the single electrode pacemaker the pacemaker either paced in atria or ventricular in modes of AOO and VOO respectively. The above described all axiom and constants are common for AOO and VOO modes. We have introduced the new events and variables in forthcoming models as refinement in an incremental development of the single electrode pacemaker system. In abstract specification of the pacemaker modes include events modeling, pacing into the heart under real time constraints, stop the pacing into the heart under time constraint and progressive increments in the clock cycle to control all the atomic events of pacemaker. There are four significant events in our abstract model of AOO and VOO modes as follows:-

The event ($Pace\_ON$) represents a pacing operation of the single electrode pacemaker into the heart either in atrial or ventricular chambers using AOO or VOO pacing modes respectively. The guard ($grd1$) states that the pacemaker

actuator should be in OFF state and next guard ($grd2$) states that clock counter (sp) should satisfy the condition $sp \geq Pace\_Int$. When guard of event satisfy then action will take the effect and pacemaker will discharge the pulse to the heart and assign the value of clock counter variable (sp) to other last clock counting variable ($last\_sp$).

EVENT Pace_ON
WHERE
grd1 : $Pacemaker\_Actuator = OFF$
grd2 : $sp \geq Pace\_Int$
THEN
act1 : $Pacemaker\_Actuator := ON$
act2 : $last\_sp := sp$
END

The event $Pace\_OFF$ is used to stop the pulse discharging to the heart and set the value "1" to current clock counter variable sp. The guard ($grd1$ and $grd2$) of this event state that the pacemaker should be in "ON" state and clock counter value should be grater then variable $Pace\_Int$.

EVENT Pace_OFF
WHERE
grd1 : $Pacemaker\_Actuator = ON$
grd2 : $sp \geq Pace\_Int$
THEN
act1 : $Pacemaker\_Actuator := OFF$
act2 : $sp := 1$
END

The event (tic) is an important event which controls the all other events of pacemaker. The guard ($grd1$) states that the value of clock counter should be in between 1 to $Pace\_Int$. The action of this event progressively increase the value of current clock counter under time constraints.

EVENT tic
WHERE
grd1 : $sp > 0 \wedge sp < Pace\_Int$
THEN
act1 : $sp := sp + 1$
END

The event Set_Pace_Int is used as keep event in abstract model for choosing the value of ($Pace\_Int$) variable. The value of variable $Pace\_Int$ can be only changed when the flag variable $Pace\_Int\_set$ is in TRUE state.

EVENT Set_Pace_Int
WHERE
grd1 : $Pace\_Int\_set = TRUE$
THEN
act1 : $Pace\_Int$
$:|$
$(Pace\_Int' \in URL\_Time\_Int \mathinner{..} LRL\_Time\_Int)$
END

Abstraction of AAI and VVI modes In the abstract model of AAI and VVI modes all the constants, variables and events are common as initial model of AOO and VOO modes. We introduce a new constant refractory period (RF[2]) that represents a period during which Pacemaker timing in the heart chamber is not affected by events that occur with in (no sensing with initiation of a new Lower Rate Interval). We have added two new variables $Pacemaker\_Sensor$ and $last\_ss$ in the abstract model of AAI and VVI mode. The new variable $Pacemaker\_Sensor$ defines as a enumerated type that represents the presence or absence of sensing pulse from the heart chamber and the variable $last\_ss$ represents the last interval (in ms.) between two sense pulses. The safety property is represented by invariant ($inv3$) that states, there is not any sensing activity in duration of refractory period (RF).

$axm1 : RF \in 150 \mathinner{..} 500$
$inv1 : Pacemaker\_Sensor \in status$
$inv2 : last\_ss \in \mathbb{N}$
$inv3 : last\_ss \geq RF \wedge last\_ss \leq LRL\_Time\_Int$

In this abstract model the event $Pace\_ON$ is similar to $Pace\_ON$ event of AOO and VOO modes. One new event ($Pace\_OFF\_with\_Sensor$) is added in the abstraction of AAI or VVI modes and some new guards and actions added in all other events of AOO and VOO modes. In the event $Pace\_OFF$ of AAI and VVI modes, we have added the new guard ($grd3$) which states that pacemaker sensor should be in ON state and action ($act2$) part states that the sensor will stop the sensing of the pulse from atria or ventricular.

EVENT Pace_OFF
WHERE
$\oplus$ $grd3 : Pacemaker\_Sensor = ON$
THEN
$\oplus$ $act2 : Pacemaker\_Sensor := OFF$
END

[2] RF : Atria Refractory Period (ARP) or Ventricular Refractory Period (VRP)

The event ($Pace\_OFF\_with\_Sensor$) is a new event in this abstraction of AAI and VVI modes. The guards ($grd1, grd2$ and $grd3$) state that when pacemaker actuator is in OFF state , Pacemaker's sensor is in ON state and clock counter is higher than refractory period (RF) then it stores the value of current clock counter sp to the new variable $last\_ss$, resets the clock counter to 1 and stop the Pacemaker's sensor for sensing the heart chamber (atria or ventricular). The LRI consist of two portions, the ventricular refractory period (VRP), and the alert period. The VRP is initiated at the start of the LRI with each sensed or paced event. It is a period during which pacemaker timing will not be affected by events that occur within it. The alert period follows and is the interval during which sensing can occur, inhibit pacing, and initiate a new LRI.

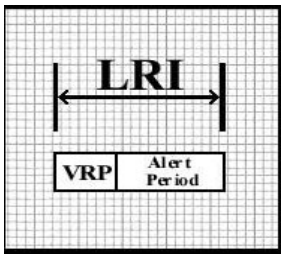
EVENT Pace_OFF_with_Sensor
WHERE
grd1 : $Pacemaker\_Actuator = OFF$
grd2 : $Pacemaker\_Sensor = ON$
grd3 : $sp \geq RF$
THEN
act1 : $last\_ss := sp$
act2 : $sp := 1$
act3 : $Pacemaker\_Sensor := OFF$
END

In the abstract model of AAI and VVI modes only modify the guard of tic event. The action part of this event is remain same as previous abstraction of AOO and VOO modes. The modified guard has been given as follows:-

EVENT tic
WHERE
grd1 : $(sp > 0 \wedge sp < RF)$
$\vee$
$(sp \geq RF \wedge sp < Pace\_Int \wedge$
$Pacemaker\_Sensor = ON)$
THEN
act1 : $sp := sp + 1$
END

The event $Sense\_ON$ is a new event of pacemaker AAI and VVI modes. The event $Sense\_ON$ starts the sensing process of Pacemaker's sensor when the sensor is in OFF state and progressive clock counter sp is higher than refractory period (RF) and lower than pace interval $Pace\_Int$. We have added some new guards in the events $Pace\_OFF$ and tic of AAI and VVI operating modes for controlling the sensor under progressive clock counter sp.

EVENT Sense_ON
WHERE
grd1 : $Pacemaker\_Sensor = OFF$
grd2 : $sp \geq RF$
grd3 : $sp < Pace\_Int$
THEN
act1 : $Pacemaker\_Sensor := ON$
END



Refractory period and Alert Period between two pulses

Abstraction of AAT and VVT modes In the abstract model of AAT and VVT modes, all the constants and variables are similar to AAI and VVI modes respectively. Similarly all the events of AAT and VVT modes same as AAI and VVI modes but a new event $Pace\_ON\_with\_Sensor$ used in place of $Pace\_OFF\_with\_Sensor$ event. The guards ($grd1, grd2$ and $grd3$) state that when Pacemaker's actuator is in OFF state, Pacemaker's Sensor is in ON and clock counter sp is higher than refractory period RF then actions state that it store the value of clock counter (sp) to variable $last\_ss$ and start the Pacemaker's sensor for sensing the intrinsic activity from the heart chambers (atria or ventricular). This event triggers pacing stimulus in atria or ventricular when sense the pulse from atria or ventricular chambers in alert period (LRI-VRP or LRI-ARP). The alert period follows and is the interval during which sensing can occur, triggers pacing, and initiate a new LRI.

EVENT Pace_ON_with_Sensor
WHERE
grd1 : $Pacemaker\_Actuator = OFF$
grd2 : $Pacemaker\_Sensor = ON$
grd3 : $sp \geq RF$
THEN
act1 : $Pacemaker\_Sensor := ON$
act2 : $last\_ss := sp$
END

4.2 First refinement

In the abstract model, we have presented that single electrode pacemaker pacing and sensing in atomic step in natural pacemaker or heart under real time constraints. So our goal is to model pacing and sensing of pacemaker in correct manner. In the first refinement step, we introduce more invariants in different operating modes of Pacemaker system to apply the strong action-reaction pattern under real-time constraints and to achieve more reliable system. In AOO and VOO modes, we have already added the strong invariants in abstarct model so we have no need to add any extra invariants but we are adding new invariants in other opearating modes as follows:-

$$
\begin{array}{l}
inv1 : sp > 0 \wedge sp < RF \Rightarrow Pacemaker\_Sensor = OFF \\
inv2 : sp > 0 \wedge sp < RF \Rightarrow Pacemaker\_Actuator = OFF \\
inv3 : sp > RF \wedge sp \leq Pace\_Int \Rightarrow Pacemaker\_Sensor = ON
\end{array}
$$

The modes of VVI and VVT is same as AAI and AAT modes respectively, there are difference between only in refractory period RF. The first and second invariants state that the Pacemaker's sensor and actuator are always in OFF state during the refractory period RF. These are the essential safety properties for the refractory period during which pacemaker timing is not be affected by any events that occur within it. The last invariant represents that the Pacemaker's sensor is in ON state and continuously sensing the intrinsic activities from the heart chamber within an alert period $(LRI - RF)$. We have added more real time constraints as the guards of event *tic* in all operating modes that controls the progressive increment of the current clock counter *sp*.

$$
\begin{array}{l}
grd1 : (sp > 0 \wedge sp < RF \wedge sp < Pace\_Int \wedge \\
\qquad Pacemaker\_Sensor = OFF \wedge Pacemaker\_Actuator = OFF) \\
\qquad \vee \\
\qquad (sp \geq RF \wedge sp < Pace\_Int \wedge \\
\qquad Pacemaker\_Sensor = ON \wedge Pacemaker\_Actuator = OFF)
\end{array}
$$

4.3 Second refinement:Threshold

The basic requirements of the single electrode pacemaker system are pacing and sensing into the natural pacemaker or heart in any particular chamber (atria or ventricular). In the stepwise refinement of abstract model we introduce the concept of sensing threshold value of the single electrode pacemaker. A pacemaker has a stimulation threshold measuring unit which measures a stimulation threshold voltage value of heart and a pulse generator for deliverying stimulation pulses to the heart. The pulse generator is controlled by a control unit to deliver the stimulation pulses with respective amplitudes related to the measured threshold value under safety margin. The constant THR holds the constant value of atria chamber as follows:-

$$axm1 : THR \in \mathbb{N}1 \land THR = 75$$

The constant THR holds the value of ventricular chamber as follows:-

$$axm1 : THR \in \mathbb{N}1 \land THR = 250$$

Pacemaker's sensor begins for sensing after the refractory period but Pacemaker's actuator delivers a pacing stimulus when sensing value is higher than the standard threshold value THR. In this refinement each time the pacemaker sensor sense the pulse signal either from atria or ventricular. The first invariant introduces in operating modes (AAI,VVI) and it states that Pacemaker's actuator is in OFF state when Pacemaker's sensor is in ON state, obtained sensor value is higher than standard threshold value, the current clock counter sp is within the alert period and state of threshold value is in TRUE state. Similarly, the second invariant is added in operating modes (AAT,VVT) and it states that Pacemaker's actuator is in ON state when Pacemaker's sensor is in ON state, obtained sensor value is higher than standard threshold value, the current clock counter sp within the alert period and state of threshold value is in TRUE state. The threshold value of different chambers(atria and ventricular) in different modes(AAI,VVI,AAT and VVT) are specified by the doctor after diagnose the patient requirements.

$$\begin{array}{l} inv1 : sp > RF \land Pace\_Sensor = ON \land thr \geq THR \land \\ \qquad sp < Pace\_Int \land thr\_val\_state = TRUE \Rightarrow \\ \qquad Pace\_Actu = OFF \\ inv2 : sp > RF \land Pace\_Sensor = ON \land thr \geq THR \land \\ \qquad sp < Pace\_Int \land thr\_val\_state = TRUE \Rightarrow \\ \qquad Pace\_Actu = ON \end{array}$$

The new variable (thr) introduce in this refinement and we add this variable in different events of last refinement. We add an extra guard ($grd4 : thr \geq THR$ in events ($Pace\_OFF\_with\_Sensor, Pace\_ON\_with\_Sensor$) and modify the guard of (tic) event as follows:-

EVENT tic
WHERE
grd1 : $(sp > 0 \land sp < RF \land sp < Pace\_Int \land$
$Pacemaker\_Sensor = OFF \land Pacemaker\_Actuator = OFF)$
$\lor$
$(sp \geq RF \land sp < Pace\_Int \land Pacemaker\_Sensor = ON \land$
$Pacemaker\_Actuator = OFF \land thr < THR \land thr\_val\_state = FALSE)$
THEN
...
END

In the refinement of event ($Sense\_ON$), we have added the new action as ($thr\_val\_state := TRUE$). This action used to change the state of threshold variable as TRUE. The new event Thr_value is introduced in all modes (AAI,AAT,VVI and VVT), which obtains the measured value from the Pacemaker's sensor. The guards of this event states that when the Pacemaker's sensor is in ON state, the current clock counter sp is within the alert period and state of threshold value thr_val_state is in TRUE state then the sensed value th is assigned to the threshold variable thr and state of threshold variable thr_val_state set to the FALSE state.

EVENT Thr_value
ANY
th
WHERE
grd1 : $Pacemaker\_Sensor = ON$
grd2 : $th \in \mathbb{N}$
grd3 : $sp \geq RF$
grd4 : $thr\_val\_state = TRUE$
grd5 : $sp < Pace\_Int$
THEN
act1 : $thr := th$
act2 : $thr\_val\_state := FALSE$
END

In this refinement, we have added a new guard $thr \geq THR$ in events ($Pace\_OFF\_with\_Sensor$, $Pace\_ON\_with\_Sensor$) and modified the guard of event tic to synchronize the pacing-sensing events with new threshold functional behavior under real-time constraints.

4.4 Third refinement:Hysteresis

In the third refinement, we introduce the concept of "$Hysteresis''$ in pacing and sensing mode of single electrode pacemaker. "Hysteresis", from the Greek for "to lag behind," means a delay of effect behind the cause. The principal purpose of hysteresis is to allow the patient to have his or her own underlying rhythm as much as possible. Hysteresis, in a pacing terminology which refers to the response pattern, in which the Pacemaker does not begin firing until the heart rate drops (e.g., 60 beats/min) that is well below the standard pacing rate (e.g., 72 beats/min). It will continue firing until one of the heart's intrinsic beats is sensed and it will not fire again until the heart rate drops below 60 bpm [3, 4]. The application of a hysteresis interval to provide consistent pacing of the atrial or ventricle, or to prevent constant pacing of the atrial or ventricle. An implantable pacemaker system is provided with a conditional hysteresis feature, whereby a hysteresis value is added to the pacing escape interval (Hyt_Pace_Int) only when the prior spontaneous rate corresponded to a rate below the top of a

predetermined hysteresis band. This feature limits the lengthening of the escape interval (Hyt_Pace_Int) when there are sudden drops in the natural rate thereby avoiding excessive changes in rate. In a preferred embodiment, the pacemaker defines a hysteresis band around a given pacing rate, lower rate limit, the band having an upper hysteresis limit (URL_Time_Int) and a lower hysteresis limit (HRL_Time_Int). No hysteresis lengthening of the escape interval is utilized for spontaneous heartbeats having rates above the upper hysteresis limit; for spontaneous heartbeats having rates between the lower rate limit and the upper hysteresis limit, an escape interval is set to have a value corresponding to a rate between the pacing limit and the lower rate limit of the hysteresis band which is below the lower rate limit; and for a sensed spontaneous rate below the lower rate limit, a hysteresis escape interval corresponding to the lower hysteresis limit is established. In the preferred embodiment, sensed heartbeats having a prior rate between the lower rate limit and the upper hysteresis limit cause an escape interval which is lengthened beyond the LRL escape interval by an amount which varies linearly with the differential between the upper hysteresis rate limit and the spontaneous rate. We introduce some new constants for modeling the $Hysteresis$ concepts in the modes of pacemaker as follows:-

$$
\begin{array}{l}
axm1 : HRL = LRL \\
axm2 : HRL\_Time\_Int = LRL\_Time\_Int \\
axm3 : Hyt\_Pace\_Int = HRL\_Time\_Int \\
axm4 : HYT\_State \in BOOL
\end{array}
$$

Here, new constant Hysteresis Rate Limit (HRL) that is equal to the Lower Rate Limit (LRL). Next constant Hysteresis Rate Ineterval (HRL_Time_Int) is also equal to the Lower Rate Ineterval (LRI). Similarly third constant escape interval (Hyt_Pace_Int) is equal to Hysteresis Rate Interval and last constant hysteresis state (HYT_State) is introduced as boolean type. To design the reliable system, we introduce three more invariants and one theorem.

$$
\begin{array}{l}
inv1 : HYT\_State = TRUE \\
\qquad \Rightarrow \\
\qquad last\_sp \geq URL\_Time\_Int \wedge last\_sp \leq HRL\_Time\_Int \\
inv2 : HYT\_State = TRUE \\
\qquad \Rightarrow \\
\qquad last\_ss \geq RF \wedge last\_ss \leq HRL\_Time\_Int \\
inv3 : HYT\_State = TRUE \\
\qquad \Rightarrow \\
\qquad Pace\_Int = HRL\_Time\_Int \\
thm1 : HYT\_State = FALSE \\
\qquad \Rightarrow \\
\qquad Pace\_Int \geq URL\_Time\_Int \wedge Pace\_Int \leq HRL\_Time\_Int
\end{array}
$$

The invariant ($inv1$) states that if hysteresis state is TRUE then interval between two pace should be in hysteresis band (upper rate limit to lower rate

limit). The next invariant ($inv2$) states that if hysteresis state is TRUE then the interval between two sensed pulse should be greater than refractory period RF and less then lower hysteresis rate limit (HRL_Time_Int). The third invariant ($inv3$) states that if hysteresis state is TRUE then pacing interval ($Pace\_Int$) and lower hysteresis rate limit (HRL_Time_Int) should be equal. The theorem ($thm1$) states that if hysteresis state is FALSE then pacing inetrval should be greater than upper rate limit time interval (URL_Time_Int) and less than hysteresis rate limit time interval (HRL_Time_Int). In this refinement the invariants and theorem is same for all the modes(AAI and VVI). Many VVI and AAI modes of pacemakers have a rate function called *hysteresis*. *Positive* hysteresis can add an additional period of time for the pacemaker to wait and see if a native R wave will occur before pacing. In this application it can occur only after an R wave is sensed and does not occur after a paced event. The hysteresis rate is less than the lower rate. In this manner the principal purpose of hysteresis is to allow the patient to have his or her own underlying rhythm as much as possible. This can help conserve the pacemakers battery life. $Hysteresis$ concept is not available in AAT and VVT modes of the pacemaker. But in the refinement we modeled the hysteresis concept for AAT and VVT modes and it satisfy all the proof obligations which occured in this refinement. We have checked it that there in no any effect in AAT and VVT modes of pacemaker when applied the $Hysteresis$. So $Hysteresis$ is only aplicable with AAI and VVI modes.

We have't introduced any extra events in this refinement. We have added the hysteresis related constants and variables in already defined events. We have added the following new guard ($grd4$) in event $Pace\_ON$ and ($grd5$) in events $Sense\_ON$ and Thr_value. These guards represent that hysteresis states (ON and OFF),hysteresis pacing inetrval and normal pace inetrval of the pacemaker parameters should be valid at different operating modes of the pacemaker in pacemaker events.

$$
\begin{array}{l}
grd4 : (HYT\_State = FALSE \wedge sp \geq Pace\_Int) \\
\qquad \vee \\
\qquad (HYT\_State = TRUE \wedge sp \geq Hyt\_Pace\_Int) \\
\\
grd5 : (HYT\_State = FALSE \wedge sp < Pace\_Int) \\
\qquad \vee \\
\qquad (HYT\_State = TRUE \wedge sp < Hyt\_Pace\_Int)
\end{array}
$$

We modify the old guard of event tic with new guard as shown in below box. This guard states that the current clock counter sp is incremented under the real-time constraints and hysteresis functional behavior for all operating modes of the Pacemaker system. This guard is necessary for satisfy the time constraints for every operation of the pacemaker. The modified guard controls the time counter in different operating modes of pacemaker.

$$
\begin{aligned}
grd1 : ((&HYT\_State = FALSE \wedge sp > 0 \wedge sp < RF \wedge \\
&sp < Pace\_Int \wedge Pace\_Sensor = OFF \wedge \\
&Pace\_Actu = OFF) \\
&\vee \\
&(HYT\_State = FALSE \wedge sp \geq RF \wedge sp < Pace\_Int \wedge \\
&Pace\_Sensor = ON \wedge Pace\_Actu = OFF \wedge \\
&thr < THR \wedge thr\_val\_state = FALSE)) \\
&\vee \\
&((HYT\_State = TRUE \wedge sp > 0 \wedge sp < RF \wedge \\
&sp < Hyt\_Pace\_Int \wedge Pace\_Sensor = OFF \wedge \\
&Pace\_Actu = OFF) \\
&\vee \\
&(HYT\_State = TRUE \wedge sp \geq RF \wedge sp < Hyt\_Pace\_Int \wedge \\
&Pace\_Sensor = ON \wedge Pace\_Actu = OFF \wedge \\
&thr < THR \wedge thr\_val\_state = FALSE))
\end{aligned}
$$

4.5 Fourth refinement:Rate Modulation

This refinement is the last and important refinement in the single electrode pacemaker system. In this refinements we introduce the rate responsive technique to the pacemaker. Rate responsive term has led to the more acceptable use of the terms rate adaptive and rate modulating. All these terms are used to describe the capacity of a pacing system to respond to physiologic need by increasing and decreasing pacing rate. The capability of a pacing system depends on the presence of one of a variety of physiologic sensors that monitor need or indication for rate variability. The predominant need for rate modulation derives from physical activity or exertion. There are other physiologic situations in which normally there are modulations of heart rate for example, with fever and emotional stress. These, however, are substantially less important, especially in the context of pacing systems. This refinement of pacemaker also give some new pacing and sensing mode (AAIR,VVIR,AATR and VVTR) of the pacemaker. The rate adapting mode of pacemaker can progressively pace faster than the lower rate, but no more than the upper sensor rate limit, when it determines that heart rate needs to increase. This typically occurs with exercise in patients that cannot increase their own heart rate. The amount of rate increase is determined by how much exertion the pacemaker thinks the patient is performing. This increased pacing rate is sometimes referred to as the "sensor indicated rate". When exertion has stopped the pacemaker will progressively decrease the paced rate down to the lower rate. For modeling the rate modulation technique in single electrode pacemaker, we introduce the some axioms as follows:-

$$axm1 : MSR \in 50 \mathinner{..} 175 \land MSR = 120$$
$$axm2 : threshold \in \mathbb{N}_1 \land threshold = 4$$
$$axm3 : reactionTime \in 10 \mathinner{..} 50 \land reactionTime = 10$$
$$axm4 : recoveryTime \in 2 \mathinner{..} 16 \land recoveryTime = 2$$
$$axm5 : responseFactor \in 1 \mathinner{..} 16 \land responseFactor = 8$$

In above axioms, ($axm1$) represents the maximum sensor rate (MSR) is maximum pacing rate allowed as a result of sensor control and it should be in between 50 to 175 pulse per minute (ppm). We have taken the nominal value of MSR in this model as 120 ppm. The next axiom ($axm2$) represents the activity threshold is the value the accelerometer sensor output shall exceed before the pacemaker's rate is affected by activity data. The nominal value of activity threshold is 4 in this model. The accelerometer shall determine the rate of increase of the pacing rate. The reaction time is the time required for an activity to drive the rate from LRL to MSR, which is defined as axiom ($axm3$). Similarly axioms ($axm4, axm5$) represent the recovery time and response factor in rate adapting pacing respectively. The recovery time shall be the time required for the rate to fall from MSR to LRL when activity falls below the activity threshold and the response factor in rate adapting pacing, the accelerometer shall determine the pacing rate that occurs at various levels of steady state patient activity. The highest response factor setting (16) shall allow the greatest incremental change in rate and the lowest response factor setting (1) shall allow a smaller change in rate. We have taken the nominal value of reaction time, recovery time and response factor in our model.

A new variable $acler\_sensed$ define as $acler\_sensed \in \mathbb{N}$, to store the sensor value of the accelerometer from the heart chamber. We have introduced the following invariants as follows in this refinement:-

$$inv1 : acler\_sensed < threshold \Rightarrow Pace\_Int = 60000/LRL$$
$$inv2 : acler\_sensed > threshold \Rightarrow Pace\_Int = 60000/MSR$$

The invariant ($inv1$) states that when the sensed value of the accelerometer sensor is less than constant activity threshold value then the pacing inetrval should be equal to $60000/LRL$ so that the heart rate never fall below the lower rate limit (LRL) and similarly the invariant ($inv2$) states that when sensed value of the accelerometer sensor is greater than constant activity threshold value then the pacing interval should be equal to $60000/MSR$, so that the heart rate never exceed the maximum sensor rate or upper rate limit of the heart pacing. These two invariants always check the safety margin in rate adapting pacing. Finally the simulation of the rate controller follows as a relation between the reach of the MSR with a exceeding input value of the treshold, and the LRL as a decrease after the reacovery time form the MSR or the normal functioning of the system.

In this final refinement we introduce only two extra events ($Increase\_Interval$, $Decrease\_Interval$) to control the pacing rate of the single electrode pacemaker

in rate adapting pacing modes. The new event ($Increase\_Interval$) controls the value of pace interval whenever the sensed value of accelerometer sensor is greater then activity threshold value. The other new event ($Decrease\_Interval$) controls the value of pace interval whenever the sensed value of accelerometer sensor less then activity threshold value. After introducing these two new events in this refinement, we have found the new modes(AOOR, VOOR, AAIR, VVIR, AATR and VVTR) in the single electrode pacemaker, which are using in the pacemaker as a rate adaptive pacing features. All these modes apply to control the pacing activity of pacemaker. So here we have found the relationship between different modes of pacemaker in stepwise refinements.

EVENT Increase_Interval
ANY
WHERE
grd1 : $acler\_sensed > threshold$
THEN
act1 : $Pace\_Int := 60000/MSR$
END

EVENT Decrease_Interval
ANY
WHERE
grd1 : $acler\_sensed < threshold$
THEN
act1 : $Pace\_Int := 60000/LRL$
END

We have introduced the similar refinements in all other modes of the single electrode Pacemaker for atria as well as ventricular chambers. Rate modulated Pacemakers mimic physiological response by increasing heart rate and, subsequently, cardiac output in response to exercise. Rate modulated Pacemaker use metabolic or motion-derived sensors to adjust pacing rate based on physiologic requirements by translating indices of increased metabolic need into signals that can be used to restore chronotropic competence. The most common and versatile sensors include activity or acceleration, minute ventilation, or combinations; these sensors remain functional with standard pacing leads. All sensor systems have some limitations. Although, any rate response is better than none,the degree of rate response depends on programming. For a single activity level, any sensor can be programmed to provide virtually any desired rate. Different sensors respond differently to the same stimuli. Hence, combinations of complementary sensors may better simulate normal sinus node response. Some devices automatically reprogram rate response parameters based on average activity levels. Finally we have modeled the all possible modes of single electrode Pacemaker with all required functions of different modes of Pacemaker system using step-

wise refinements. We have also discovered relationship among operating modes (see Figure-5).

5 Model Validation and Analysis

A systematic testing approach is used to validate the models derived during the staged development process. "Validation" in this context refers to the activity of gaining confidence that the formal models developed are consistent with the requirements expressed in the requirements document [5]. We have used the ProB [17] validation tool to test the all scenarios of all modes of single electrode pacemaker.The pacemaker specification is developed and formally proven by the Event-B. However, the development contains certain assumptions about the actual single electrode pacemaker system which have to be validated separately in order to ensure safe operation. We have used the ProB to dig all required information and missing safety properties in the model. We have used the ProB to test all modes scenarios of the single electrode pacemaker in different interesting situations such as the absence of input pulses, hysteresis, threshold and rate adapting pacing. The validation process involves to sense the chamber and paced into the chamber at the correct time in different situations such as hysteresis and rate adapting modes. We have successfully tested the all cases of pacemaker modes after modeling in Event-B, using ProB validation tool.

Model	Total number of POs	Automatic Proof	Interactive Proof
Abstract Model	118	114(95%)	4(5%)
First Refinement	60	44(73%)	16(27%)
Second Refinement	44	40(91%)	4(9%)
Third Refinement	36	24(66%)	12(34%)
Fourth Refinement	78	78(100%)	0(0%)
Total	336	300(89%)	36(11%)

Table-2 : Proof statistics

Through careful use of small refinement steps and appropriate intermediate abstractions, we are able to achieve an impressive degree of automatic proof. Here in this section we have also mentioned the table of proof obligations for single electrode pacemaker. The Table-2 is expressing the proof statistics of the development in the RODIN tool. These statistics measure the size of the model, the proof obligations are generated and discharged by the Rodin platform, and those are interactively proved. In the table, the total number of POs column represents the total number of proof obligations generated for each level. The Interactive Proof column represents the number of those proof obligations that have to be proved interactively. Those proof obligations that are not proved interactively are proved completely automatically by the prover. The complete development of single electrode Pacemaker system results in 336(100%) proof obligations, in which 300(88%) are proved completely automatically by the RODIN tool. The remaining 36(12%) proof obligations are proved interactively using RODIN tool.

This refinement approach together with the RODIN tool supports an incremental style of system development. We have presented the complete refinements in top down manner. We have started with the highest level specification and then produced a model approximating the lowest level. However in attempting to prove refinement between these models it is clear that the abstraction gap is too large would have required a complex gluing invariant. Instead we have decided that some intermediate abstraction are required. Any modifications to the refinement model has an impact on the existing proofs. For that we have need to give the proper gluing invariants. In the model, many proof obligations are generated due to introduction of new functional behaviors and their parameters (threashold,hysteresis and rate modulation) under real-time constraints. In order to guarantee the correctness of these functional behaviors, we have established various invariants in stepwise refinement. Most of the proofs are interactively discharged in the 1st and the 3rd refinements. Few proof obligations are also proved interactively in other refinements. The stepwise refinement of the single electrode Pacemaker system helps to achieve a high degree of automatic proof.

6 Conclusion and Future Works

We have presented a case study to formally develop a single electrode Pacemaker system in the Event-B and to discover the exact functional behavior of pacing and sensing events. Our approach for formalizing and reasoning about action-reaction is based on real-time system as a Pacemaker system. The Pacemaker case study suggests that such an approach can yield a viable model that can be subjected to useful validation against system-level properties at an early stage in the development process. We have applied the action-reaction [10] and time based patterns [22, 23] to develop the Pacemaker system. The proposed techniques based on development patterns intend to assist in the design process of system where correctness and safety are important issues.

More precisely, we have presented development of operating modes of single electrode Pacemaker system. For quick understanding, we have formalized several different developments, each highlighting a different aspect of problem, making different assumptions about the operating modes and establishing different properties. For example, we have considered a case of constant sensing and pacing, threshold parameter for electrode sensor, hysteresis mode pacing and rate modulation operating modes. We have also discovered the hierarchical relationship as optional parameter features between common and different operating modes of the Pacemaker in stepwise refinement (see Figure-5).

We have outlined how an incremental refinement approach to the single electrode Pacemaker system allow to achieve a high degree of automatic proof using RODIN tool. Our different developments reflect not only the many facets of the problem, but also that there is a learning process involved in understanding the problem and its ultimate possible solutions. The approach is concerned with separation : firstly, it proves the basic behavior of single electrode Pacemaker system at abstract level secondly it introduces the peculiarity of the specific prop-

erties. We have proved the fundamental properties in the beginning, namely the action-reaction with real-time constraints and the uniqueness of a solution, are kept through the refinement process (provided, of course, the required proofs are done). This is the superiority of proposed approach. Finally, we have validated the single electrode Pacemaker system using the ProB validation tool and found the correctness of our proved single electrode Pacemaker system under the real-time constraints.

In the future, we have plan to meet with physician and cardiologist experts to improve the models as per current requirements of patients and introduce new behaviors of operating modes. We have also plan to work on double electrode pacemaker system specification.

Acknowledgement. This work is supported by grant No. ANR-06-SETI-015-03 awarded by the Agence Nationale de la Recherche.

References

1. Woodcock, J.: First steps in the verified software grand challenge. IEEE Computer **39**(10) (2006) 57–64
2. 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
3. Malmivuo, J. In: Bioelectromagnetism. Oxford University Press (1995) ISBN 0-19-505823-2.
4. Hesselson, A. In: Simplified Interpretations of Pacemaker ECGs. Blackwell Publishers (2003) ISBN 978-1-4051-0372-5.
5. Boston Scientific: Pacemaker system specification, Technical report, B.S. (2007)
6. Macedo, H.D., Larsen, P.G., Fitzgerald, J. LNCS. In: Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System Using VDM. Springer, Los Alamitos, CA, USA (2008) 181–197
7. Bjorner, D.: DOMAIN ENGINEERING Technology Managemnt, Reserach and Engineering. Volume 4 of COE Research Monograph Series. JAIST (2009)
8. Manna, V.P.L., Bonanno, A.T., Motta, A.: Poster on a simple pacemaker implementation, ACM (May 2009)
9. Cansell, D., Méry, D. In: The event-B Modelling Method: Concepts and Case Studies. Springer (2007) 33–140 See [25].
10. Abrial, J.R.: Modeling in Event-B: System and Software Engineering. Cambridge University Press (2009) Forthcoming book.
11. Project RODIN: Rigorous open development environment for complex systems. http://rodin-b-sharp.sourceforge.net/ (2004) 2004–2007.
12. Back, R.: On correct refinement of programs. Journal of Computer and System Sciences **23**(1) (1979) 49–68
13. Abrial, J.R.: The B book - Assigning Programs to Meanings. Cambridge University Press (1996)
14. Morgan, C.: Programming from Specifications. Prentice Hall International Series in Computer Science. Prentice Hall (1990)
15. Abrial, J.R., Cansell, D.: Click'n prove: Interactive proofs within set theory. In: TPHOL 2003. (2003) 1–24

16. 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
17. ProB: The prob animator and model checker for the b method. http://www.stups.uni-duesseldorf.de/ProB/overview.php/
18. Barold, S.S., Stroobandt, R.X., Sinnaeve, A.F. In: Cardiac Pacemakers Step by Step. Futura Publishing (2004) ISBN 1-4051-1647-1.
19. Ellenbogen, K.A., Wood, M.A. In: Cardiac Pacing and ICDs. 4th Edition, Blackwell (2005) ISBN-10 1-4051-0447-3.
20. 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)
21. Abrial, J.R.: Using design patterns in formal devlopments example: A mechanical press controler. journe scientifique du ppf iaem transversal - dveloppement incrmental et prouv de systmes, april 2006
22. Cansell, D., Méry, D., Rehm, J.: Time Constraint Patterns for Event B Development. Lecture Notes in Computer Science. In: Formal Specification and Development in B. Springer US (2006) 140–154 ISSN 0302-9743.
23. 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)
24. Love, C.J. In: Cardiac Pacemakers and Defibrillators. Landes Bioscience Publishers (2006) ISBN 1-57059-691-3.
25. Bjørner, D., Henson, M.C., eds.: Logics of Specification Languages. EATCS Textbook in Computer Science. Springer (2007)