Technical Report on Formal Development of Two-Electrode Cardiac Pacing System

Dominique Méry, Neeraj Kumar Singh

Technical Report on Formal Development of Two-Electrode Cardiac Pacing System

Dominique Méry
E-mail: mery@loria.fr

Neeraj Kumar Singh
E-mail: neerajkumar.singh@loria.fr

LORIA
Université Henri Poincaré Nancy 1
BP 239
54506 Vandœuvre-lès-Nancy
April 13, 2010

Abstract

To build a high quality and zero defects medical devices and softwares is a crucial task. Formal modeling techniques help to achieve this target at certain level. Formal modeling of High-Confidence Medical devices those are too much error prone in operating, are an International Grand Challenge in the area of Verified Software. Formal modeling of an artificial pacemaker is also one of the proposed challenge. The architecture and functional behaviour of the double electrode pacemaker is more complex than the single electrode pacemaker. Proof-based an incremental approach, we use to develop the formal model of functional behaviour of the double electrode pacemaker. 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, which adds parametric based functional properties to the abstract system-level specifications using some intermediate models. The properties express system architecture and action-reaction under real-time constraints. This technical report focuses on the formal development of the double electrode operating modes and finds the common architecture of operating modes in tree form that helps to make the consistent system. The Event B modeling language is used to express the double electrode pacemaker and generated proof obligations are proved by RODIN platform. Finally, the pacemaker model has been validated by an Event B animator; ProB tool.

1 Introduction

The high confidence medical devices are highly dependent on the performance, the need for absolute precision can be a life or death issue. So, after a long time due to many failure cases and untrustworthiness of the medical devices, the equipment manufacturers have turned towards formalism in the engineering of medical device. For decades, software failures have costed billions of dollars a year [37]. Software failures and lack of warranties of products have emerged the software crisis. Due to software crisis, various formalism and rigourous techniques (VDM, Z, Event-B, Alloy etc.) have been used in the development process of safety-critical systems. These approaches provide the certain level of reliability and confidence to develop the error free systems. Formal methods and their tools have achieved a certain level of usability that could be applied even in industrial scale applications allowing software developers to provide more meaningful guarantees to their projects.

The high confidence medical devices are too complex in operating and several concurrent process are running together. To validate such kind of system, only simulation and testing can be usual techniques. 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 of mathematical proofs provide correctness of a given system that can be realized the actual system in early stage of development.

Tony Hoare suggested a Grand Challenges for Computing Research [24] to integrate the research community to work together towards a common goal, agreed to be valuable and achievable by a team effort within a predicted timescale. Verification Grand Challenges is one of them. From the Verification Grand Challenges, many application areas were proposed by the Verified Software Initiative [23]. The pacemaker specification [13, 20, 2] has been proposed by the software quality research laboratory at McMaster University as a pilot project for the Verified Software Initiative [38, 30]. 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 [11],

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 the pacemaker system

$\mathcal{D}$ is the context of the problem to solve and it is defined in EVENT B (parameters, constants etc.). $\mathcal{S}$ is the system made up of the pacemaker and the heart. $\mathcal{R}$ is requirements for the heart system such that sensing and demand pacing under time constraints. The operating modes of Bradycardia therapy and formal model of pacemaker system are based on informal requirements, which are given by Boston Scientific [13].

H.D. Macedo, et al. [30] have developed a distributed real-time model of a cardiac pacing system but the development was not based on proof and refinement techniques. Similarly, in other case study V.P. Manna, et al. [32] have developed a simple pacemaker implementation. Recently, Gomes et al [21] wrote a formal specification of the pacemaker system using the Z modelling language. According to the paper, they have modelled the sequential model similar to H.D. Macedo et al. work [30]. In this report, we have specially covered the bradycardia operating modes of the double electrode pacemaker. We have developed the parametric and functional based incremental development of bradicardia operating modes. Moreover, we have added the *threshold*, and *rate adaptive* bradicardia operating modes. Incremental development is based on refinement approach and at every level of the development, we have proved the all required safety properties (*refinement* and *consistency checking*). Other specifications [30, 21] of the pacemaker developed as a one shot model, means those are not based on the refinement.

Our approach is based on the Event B modelling language which is supported by the RODIN platform integrating tools for proving models and refinements of models; moreover we use the ProB tool [34, 28] for analyzing the models and for validating these models. Here we present a stepwise development to model and verify such interdisciplinary requirements in Event B [14, 4]. 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 the requirements of the pacemaker. Hence, validation must be carried out by both formal modeling and domain experts. The abstract model includes event modeling of bradycardia operating modes of a double electrode pacemaker system.

The refinement is supported by the RODIN [35] platform guarantees the preservation of safety properties. Thus, the behavior of the final system is preserved by an abstract model as well as in the correctly refined models. Proof-based development methods [4] integrate formal proof techniques in the development of software systems. The main idea is to start with an abstract model of the given system. Details are gradually added to this first abstract model by building a sequence of more concrete events. The relationship between two successive models in this sequence is *refinement* [4, 6]. The current work intends to explore those problems related to the modeling of bradycardia operating modes using a double electrode pacemaker system under real time constraints and to evaluate the refinement process.

The outline of the remaining report is as follows: Basic outline of a pacemaker and a heart system are given in Section 2. The modelling framework is presented in Section 3. Section 4 explores the refinement based formal development of the double electrode pacemaker. The pacemaker models are validated by the ProB tool [34, 28] and correctness of the system are analyzed by generated proof obligations (see Table-3) in Section 5. Finally, in Section 6, we conclude the report with some lessons learned from this experience and some prospective along with direction for future work.

2 Basic Overview of Pacemaker system

In Fig. 1 a suitable interface block diagram of the pacemaker and the heart is given. The conventional pacemakers serve two major functions, namely pacing and sensing. The pacemaker actuator is pacing by the delivery of a short, intense electrical pulse into the heart. However the pacemaker sensor uses the same electrode to detect the intrinsic activity of the heart. So, the pacemaker function of pacing and sensing activities are dependent on the behavior of the heart. The sensing and pacing functions regulates the heart rhythm. In this report, we present only the formal models of the double electrode pacemaker.

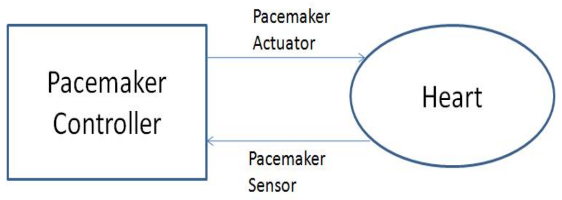

Fig. 1 Pacemaker and Heart Interface

The pacemaker system is a small electronic device that helps the heart to maintain the regular heart beat. 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. In this study, the pacemaker is treated as an embedded system operating in an environment containing the heart. We first review the heart system that interact with the pacemaker (Section 2.1) and then consider elements of the pacemaker system itself (Section 2.2).

2.1 The Heart System

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

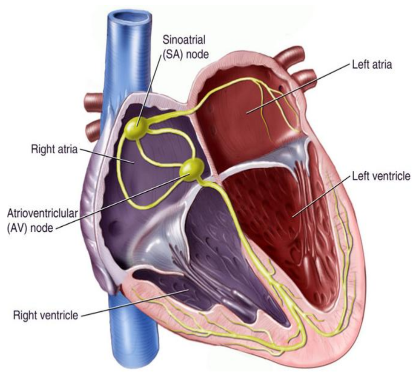

Fig. 2 Heart or Natural Pacemaker [1]

An artificial pacemaker is implanted to assist the natural pacemaker or heart in case of a arrhythmias condition to control the heart rate [31]. Arrhythmias are due to cardiac problems producing abnormal heart rhythms. In general arrhythmias reduce heamodynamic performance including situations where the heart's natural pacemaker develops an abnormal rate or rhythm or when normal conduction pathways are interrupted and a different part of the heart takes over control of the rhythm. An arrhythmia can involve an abnormal rhythm increase (tachycardia; $>$ 100 bpm) or decrease (bradycardia; $<$ 60 bpm), or may be characterized by an irregular cardiac rhythm, e.g. due to asynchrony of the cardiac chambers. The irregularity of the heartbeat, called bradycardia and techycardia. The bradycardia indicates that the heart rate falls below the expected level while in techycardia indicates that the heart rate go above the expected level of the heart rate. An artificial pacemaker can restore synchrony between the atria and ventricles. In an artificial pacemaker system, the firmware controls the hardware such that an adequate heart rate is maintained, which is necessary either because the heart's natural pacemaker is insufficiently fast or slow or there is a block in the heart's electrical conduction system [7, 18, 22, 27, 29, 31]. Beats per minute (bpm) is a basic unit to measure the rate of heart activity.

2.2 The Pacemaker system

The basic elements of the pacemaker system [7, 18] are:

1. **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 its configuration: either one ("unipolar") or two ("bipolar") separated points of electrical contact with the heart.

2. **The Pacemaker Generator:** The pacemaker is both the power source and the brain of the pacing and sensing systems. It contains an implanted battery and controller as an electronic circuitry.

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

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

4. **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.

In the double 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 [13] describes all possible operating modes that are controlled by the different programmable parameters of the pacemaker. All the programmable parameters are related to real-time and action-reaction constraints, that is used to regulate the heart rate.

In order to understand the "language" of pacing, it is necessary to comprehend the coding system that produced by a combined working party of the North American Society of Pacing and Electrophysiology (NASPE) and the British Pacing and Electrophysiology Group (BPEG) known as NBG(NASPE/BPEG generic) code [33]. This is a code of five letters of which the first three are most often used. The code provides a description of the pacemaker pacing and sensing functions. The sequence is referred to as "bradycardia operating modes"(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 indicates which chambers are being paced, the second letter indicates which chambers are being sensed, the third letter of the code indicates the response to sensing and the final letter, which is optional indicates the presence of rate modulation in response to the physical activity measured by the accelerometer. Accelerometer is an additional sensor in the pacemaker system that detects a physiological result of exercise or emotion and increase the pacemaker rate on the basis of a programmable algorithms. "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 an inhibition from further pacing after sensing of an intrinsic activity from the heart chamber.The NBG code also uses a fifth letter relating to antitachycardia function which is not discussed in this report.

3 The modelling framework

Here, we will summarize the concepts of the EVENT B modelling language developed by Abrial [14, 4] and will indicate the links with the tool called RODIN [35]. The modelling process deals with various languages, as seen by considering the triptych of Bjoerner [8, 9, 10, 12]: $\mathcal{D}, \mathcal{S} \longrightarrow \mathcal{R}$. Here, the domain $\mathcal{D}$ deals with properties, axioms, sets, constants, functions, relations, and theories. The system model $\mathcal{S}$ expresses a model or a refinement-based chain of models of the system. Finally, $\mathcal{R}$ expresses requirements for the system to be designed. Considering the EVENT B modelling language, we notice that the language can express *safety* properties, which are either *invariants* or *theorems* in a machine corresponding to the system. Recall that two main structures are available in EVENT B :

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

A EVENT B model is defining either a context or a machine. The triptych of Bjoerner [8, 9, 10, 12] $\mathcal{D}, \mathcal{S} \longrightarrow \mathcal{R}$ is translated as follows: $\mathcal{C}, \mathcal{M} \longrightarrow \mathcal{R}$, where $\mathcal{C}$ is a context, $\mathcal{M}$ is a machine and $\mathcal{R}$ are the requirements. The relation $\longrightarrow$ is defined to be a satisfaction relation with respect to an underlying logico-mathematical theory. This satisfaction relation is supported by the RODIN platform. A machine is organizing events modifying state variables and it uses static informations defined in a context. These basic structure mechanisms are extended by the refinement mechanism which provides a mechanism for relating an abstract model and a concrete model by adding new events or by adding new variables. This mechanism allows us to develop gradually EVENT B models and to validate each decision step using the proof tool. The refinement relationship should be expressed as follows: a model M is refined by a model P, when P is simulating M. The final concrete model is close to the behaviour of real system that is executing events using real source code. We give details now on the definition of events, refinement and guidelines for developing complex system models.

3.1 Modelling actions over states

The event-driven approach [3, 14, 4] is based on the B notation. It extends the methodological scope of basic concepts to take into account the idea of *formal models*. Briefly, a formal model is characterized by a (finite) list x of *state variables* possibly modified by a (finite) list of *events*, where an invariant $I(x)$ states properties that must always be satisfied by the variables x and *maintained* by the activation of the events. In the following, we summarize the definitions and principles of formal models and explain how they can be managed by tools [5, 16, 35].

Generalized substitutions are borrowed from the B notation. They provide a means to express changes to state variable values. In its simple form $x := E(x)$, a generalized substitution looks like an assignment statement. In this construct, x denotes a vector built on the set of state variables of the model, and $E(x)$ denotes a vector of expressions. Here, however, the interpretation we shall give to this statement is not that of an assignment statement. We interpret it as a *logical simultaneous substitution* of each variable of the vector x by the corresponding expression of the vector $E(x)$. There exists a more general normal form of this, denoted by the construct $x : |(P(x, x'))$. This should be read as *x is modified in such a way that the value of x afterwards, denoted by x', satisfies the predicate* $P(x, x')$, where x' denotes the *new value* of the vector and x denotes its *old value*. This is clearly nondeterministic in general.

An event has two main parts, namely, a *guard*, which is a predicate built on the state variables, and an *action*, which is a generalized substitution. An event can take one of three normal forms. The first form (BEGIN $x : |(P(x, x')$ END) shows an event that is not guarded, being therefore always enabled and semantically defined by $P(x, x')$. The second form (WHEN $G(x)$ THEN $x : |(Q(x, x'))$ END) and third form (ANY t WHERE $G(t, x)$ THEN $x : |(R(x, x', t))$ END) are guarded by a guard that states the necessary condition for these events to occur. The guard is represented by WHEN $G(x)$ in the second form, and by ANY t WHERE $G(t, x)$ (for $\exists t \cdot G(t, x)$) in the third form. We note that the third form defines a possibly nondeterministic event where t represents a vector of distinct local variables. The *before–after* predicate $BA(x, x')$, associated with each of the three event types, describes the event as a logical predicate expressing the relationship linking the values of the state variables just before (x) and just after (x') the *execution* of event EVENT evt. The second and the third forms are semantically equivalent to $G(x) \wedge Q(x, x')$ resp. $\exists t \cdot (G(t, x) \wedge R(x, x', t)$. Table-2 summarizes the three possible forms for writing a B event:

Proof obligations (INV 1 and INV 2) are produced by the RODIN tool [35] from events to state that an invariant condition $I(x)$ is preserved. Their general form follows immediately from the definition of the before–after predicate $BA(e)(x, x')$ of each event e (see Table-2). Note that it follows from the two guarded forms of the events that this obligation is trivially discharged when the guard of the event is false. Whenever this is the case, the event is said to be *disabled*. The proof obligation FIS expresses the feasibility of the event e with respect to the invariant I.

3.2 Model refinement

The refinement of a formal model allows us to enrich the model via a *step-by-step* approach and is the foundation of our correct-by-construction approach [26]. Refinement provides a way to strengthen invariants and to add details to a model. It is also used to transform an abstract model to a more concrete version by modifying the state description. This is done by extending the list of state variables (possibly suppressing some of them), by refining each abstract event to a corresponding concrete version, and by adding new events. The abstract

Event e	Before-after Predicate BA(e)(x,x')

BEGIN
$x :\|(P(x,x'))$
END
	$P(x,x')$

WHEN $G(x)$ THEN $x :\|(Q(x,x'))$ END	$G(x) \wedge Q(x,x')$
ANY t WHERE $G(t,x)$ THEN $x :\|(R(x,x',t))$ END	$\exists t \cdot (\, G(t,x) \wedge R(x,x',t)\,)$

Proof obligations

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

Table-2 Event B events and proof obligations

(x) and concrete (y) state variables are linked by means of a *gluing invariant* $J(x, y)$. A number of proof obligations ensure that (1) each abstract event is correctly refined by its corresponding concrete version, (2) each new event refines *skip*, (3) no new event takes control for ever, and (4) relative deadlock freedom is preserved. Details of the formulation of these proofs follows.

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

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

Now, proof obligation (2) states that $BA(f)(y, y')$ must refine *skip* $(x' = x)$, generating the following simple statement to prove (2).

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

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

The Event B modelling language is supported by the RODIN platform [35] and has been introduced in publications [4, 14], where there are many case studies and discussions about the language itself and the foundations of the Event B approach. The language of *generalized substitutions* is very rich, enabling the expression of any relation between states in a set-theoretical context. The expressive power of the language leads to a requirement for help in writing relational specifications, which is why we should provide guidelines for assisting the development of Event B models.

3.3 Guidelines for Event B Modelling

Considering design patterns [19], the purpose is to capture structures and to make decisions within a design that are common to similar modeling and analysis tasks. They can be re-applied when undertaking similar tasks in order to reduce the duplication of effort. The design pattern approach 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 producing 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.

The 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.

Under action-reaction chapter [4] 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 because the action moves too quickly (the continuance of an action is too short, or the interval between actions is too short). This is known as pattern of action and weak reaction.

Action and Strong Reaction: For every action, there is a corresponding reaction. To keep proper synchronization between action and corresponding reaction, known as pattern of action and strong reaction.

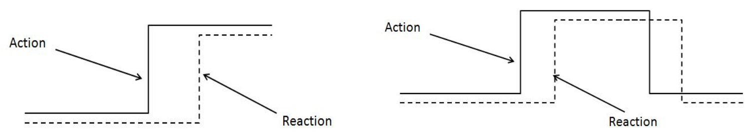

Fig. 3 Action-Reaction Patterns

The action-reaction events of a pacemaker system are based on the time constraint pattern in IEEE 1394 proposed by Cansell et. al and on the 2-Slots Simpson Algorithm case studies [15, 36]. 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, 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 EVENT B language. It is only a modeling technique instead of a specialized formal system. The timed variable 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.

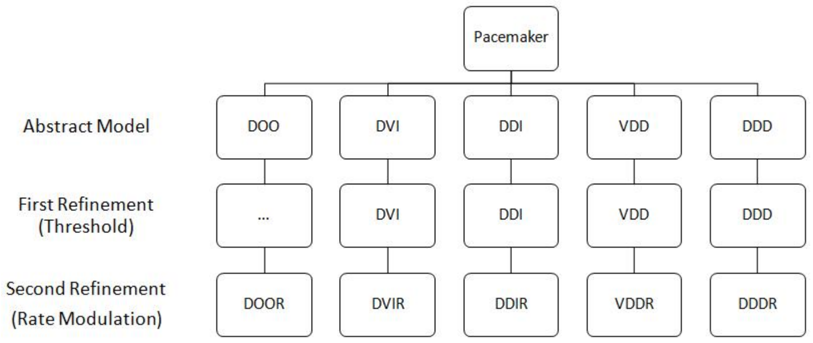

Fig. 4 Refinement structure of the double electrode pacemaker operating modes

4 Formal Development

The two electrode pacemaker system is more complex than one electrode pacemaker system. We have designed the block diagram (see Fig. 4) of hierarchical tree structure of the possible operating modes of the double electrode pacemaker. The hierarchical tree structure shows the stepwise refinement from abstract to concrete model. Each level of refinement introduces the new features of pacemaker as functional and parametric requirements. The root of this tree indicates the double electrode pacemaker. In a double electrode pacemaker, two electrodes are placed in both chambers; atrium and ventricular. These atrium and ventricular are the right atrium and ventricular. The next five branches of tree show the five opearting modes; DOO, DVI, DDI, VDD, and DDD (see Table-1). In thses operating modes, the pacemaker uses the both electrodes to pace in both chambers synchronously. It is an abstract level of the model. In the abstract model, we introduce all the operating modes abstractly with required properties of the pacemaker. From first refinement to last refinement, there is only one branch in every operating modes of the atrium and the ventricular chambers. The subsequent refinement models introduce all detailed information for the resulting system. Every refinement level shows an extension of previous operating modes as an introduction of a new feature or functional requirement. the triple dots (...) represents that there is no refinement at that level in particular operating mode (DOO). In abstract level and first refinement level, we have similar operating modes. But in the second refinement level, we have achieved the additional rate adaptive operating modes(i.e.DOOR, DVIR, DDIR, VDDR and DDDR). These operating modes are different from the previous levels operating modes. These refinement structure is very helpful to model the functional requirements of the double electrode cardiac pacemaker. The following outline is given about every refinement level to understand the basic notion of the cardiac pacemaker model:-

- **Abstract Model :** Specifies the pacing, sensing and timing components with the help of action-reaction and real-time pattern using some initial events ($Actuator\_ON\_A$, $Actuator\_OFF\_A$, $Actuator\_ON\_V$, $Actuator\_OFF\_V$, $Sensor\_ON\_A$, $Sensor\_OFF\_A$, $Sensor\_ON\_V$, $Sensor\_OFF\_V$, tic, tic_AV).

- **Refinement 1 :** Introduces the *threshold* parameter to filter the exact sensing value within a sensing period to control the sensing and the pacing event and introduces more invariants to satisfy the pacing and sensing requirements of the system in both chambers.
- **Refinement 2 :** Introduces the *accelerometer sensor* component and *rate modulation* function to achieve the new rate adaptive operating modes of the pacemaker.

We have presented here only selected parts of our formalization and omit proof details.

4.1 The Context and Initial Model

In this section we describe the formal development of initial modes of double electrode cardiac pacemaker system. In the abstract model, we introduce the basic notions of action-reaction and real-time constraints using actuator and sensor in different heart chambers. In this abstraction, we begin with an abstract model of a double electrode cardiac 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 related to rate modulation are not described in this level. Instead they will emerged into final refinement. Thus, in this level, for every modes of pacemaker are treated in the same way as common basic modes, which are essential for the double electrode cardiac pacemaker. The model consists of several modules, each corresponding to an operating mode of the pacemaker. Here, we define the required context and then abstraction of the double electrode cardiac pacemaker system.

We begin by defining the Event-B context. The context uses the sets and constants to define the axioms and theorems. The axioms and theorems represent the logical formulation of the system. The logical formulation is the constant behaviour and the set of properties of the system. In the context, we define the constants LRL and URL that relate to the lower rate limit (LRL) (minimum number of pace pulses delivered per minute by pacemaker) and upper rate limit (URL) (how fast the pacemaker will allow the heart to be paced). These constants are extracted from the pacemaker specification document [13]. The lower rate limit (LRL) must be between 30 and 175 pulse per minute (ppm) and upper rate limit (URL) must be between 50 and 175 pulse per minute (ppm). To test the model by ProB model checker [34], we have taken a nominal value of lower rate limit (LRL) as 60 ppm and upper rate limit (URL) as 120 ppm according the pacemaker specification document [13].

The two new constants URI (upper rate interval) and LRI (lower rate interval) are defined by axioms ($axm3$ and $axm4$). 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. Additionaly, we define an enumerated set *status* of an electrode as ON and OFF states and new constant atrioventricular interval $FixedAV$ in $axm5$ and $axm6$, respectively. Refractory period constants Atria Refractory Period ARP, Ventricular Refractory Period VRP and Post Ventricular Atria Refractory Period $PVARP$ are defined in $axm7$,$axm8$ and $axm9$, respectively. Another new constant V_Blank is defined as blanking period as initial period of VRP. Finally, we have introduced some basic initial properties between defined constants of the system by axioms($axm11$, $axm12$, , $axm13$, $axm14$ and $axm15$).

$$\boxed{\begin{array}{l}
axm1 : LRL \in 30 \mathinner{..} 175 \\
axm2 : URL \in 50 \mathinner{..} 175 \\
axm3 : URI \in \mathbb{N}_1 \wedge URI = 60000/URL \\
axm4 : LRI \in \mathbb{N}_1 \wedge LRI = 60000/LRL \\
axm5 : status = \{ON, OFF\} \\
axm6 : FixedAV \in 70 \mathinner{..} 300 \\
axm7 : ARP \in 150 \mathinner{..} 500 \\
axm8 : VRP \in 150 \mathinner{..} 500 \\
axm9 : PVARP \in 150 \mathinner{..} 500 \\
axm10 : V\_Blank \in 30 \mathinner{..} 60 \\
axm11 : LRL < URL \\
axm12 : URI < LRI \\
axm13 : URI > PVARP \\
axm14 : URI > VRP \\
axm15 : VRP \geq PVARP
\end{array}}$$

4.1.1 Abstraction of DOO mode:

In the double electrode pacemaker system, the pacemaker delivers a pacing stimulus in the atrial and ventricular chambers. In DOO operating mode of double electrode cardiac pacemaker system, the first letter 'D' represents that the pacemaker paces both atrial and ventricle, second letter 'O' represents that the pacemaker does not sense the atrial and the ventricle chambers and final letter 'O' represents that there is no any inhibits or triggers in both chambers (atrail and ventricular). In the block diagram (Fig-5) of heart pacing in DOO operating mode pacemaker will pace both chambers (atrial and ventricular) asynchronously at the constant rate regardless of the underlying rhythm. It does not sense, If the native rhythm is slower than the constant rate then atrial and ventricular capture will most likely be seen at the constant rate.

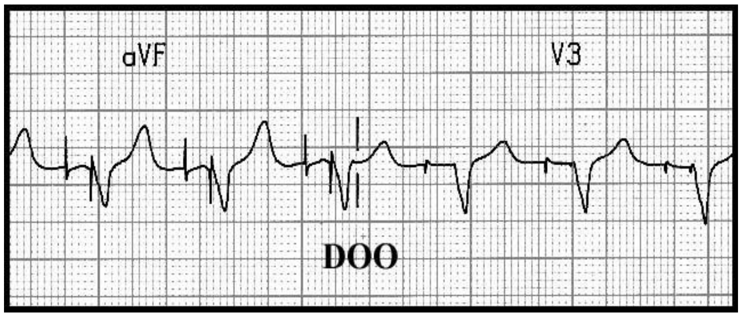

Fig. 5 Basic block diagram of ECG rhythm strip in DOO Operating Mode

In our initial model, we have formalized the functional behaviors of the cardiac pacemaker system , where two new variables $PM\_Actuator\_A$ and $PM\_Actuator\_V$ are represented ON or OFF states of the pacemaker's actuators for pacing in the atrial and ventricular chambers. An interval between two paces is defined by a new variable *Pace_Int* that must be between upper rate interval (URI) and lower rate interval (LRI), is represented by an invariant ($inv3$). The variable *Pace_Int* is an interval between two paces of ventricular chamber that is initialized by the system before start the pacing. This interval is equal to atrioventricular(AV) interval plus ventriculoatrial(VA) interval. A variable *sp* (since pace) represents a current *clock counter*. A variable *last_sp* represents the last interval (in ms.) between two paces and a safety property in invariant ($inv5$) states that last interval must be between URI and LRI. In invariant ($inv6$) a new variable *Atria_state* is used as boolean type to control the state of the atrial chamber. The invariant ($inv7$) states that the pacemaker's actuator of atrial and ventricular chambers are OFF when clock counter *sp* is less than ventriculoatrial (VA) interval and atrial state (*Atria_state*) is $FALSE$. The next invariant ($inv8$) represents that the pacemaker's actuator of both chambers are OFF when clock counter *sp* is greater than atrioventricular (AV) interval and atrial state (*Atria_state*) is $TRUE$. The last invariants ($inv9$ and $inv10$) state that pacemaker's actuator of atrial is ON when clock counter *sp* is equal to ventriculoatrial (VA) interval $Pace\_Int - FixedAV$ and pacemaker's actuator of ventricular is ON when clock counter *sp* is equal to the pace interval *Pace_Int*, respectively.

$inv1 : PM\_Actuator\_A \in status$
$inv2 : PM\_Actuator\_V \in status$
$inv3 : Pace\_Int \in URI \,..\, LRI$
$inv4 : sp \in 1 \,..\, Pace\_Int$
$inv5 : last\_sp \geq URI \land last\_sp \leq LRI$
$inv6 : Atria\_state \in BOOL$
$inv7 : sp < (Pace\_Int - FixedAV) \land Atria\_state = FALSE$
$\quad \Rightarrow$
$\quad PM\_Actuator\_V = OFF \land PM\_Actuator\_A = OFF$
$inv8 : sp > (Pace\_Int - FixedAV) \land sp < Pace\_Int \land$
$\quad Atria\_state = TRUE \Rightarrow$
$\quad PM\_Actuator\_A = OFF \land PM\_Actuator\_V = OFF$
$inv9 : PM\_Actuator\_A = ON$
$\quad \Rightarrow$
$\quad sp = Pace\_Int - FixedAV$
$inv10 : PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int$

In the abstract specification of *DOO* operating mode, there are five events $Pace\_ON\_A$ to start pacing in atrial, $Pace\_OFF\_A$ to stop pacing in atrial, $Pace\_ON\_V$ to start pacing in ventricular, $Pace\_OFF\_V$ to stop pacing in ventricular and *tic* to increment the current clock counter *sp* under real time constraints.

```
EVENT Pace_ON_A
  WHEN
    grd1 : $PM\_Actuator\_A = OFF$
    grd2 : $Atria\_state = FALSE$
    grd3 : $sp = Pace\_Int - FixedAV$
  THEN
    act1 : $PM\_Actuator\_A := ON$
  END
```

The events $Pace\_ON\_A$ and $Pace\_OFF\_A$ start and stop the pulse discharging into the atrial chamber. The guards and an action of event ($Pace\_ON\_A$) state that pacemaker's actuator ($PM\_Actuator\_A$) of atrial is ON when pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF, atrial state ($Atria\_state$) is FALSE and clock counter *sp* is equal to ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$).

```
EVENT Pace_OFF_A
  WHEN
    grd1 : $PM\_Actuator\_A = ON$
    grd2 : $PM\_Actuator\_V = OFF$
    grd3 : $sp = Pace\_Int - FixedAV$
  THEN
    act1 : $PM\_Actuator\_A := OFF$
    act2 : $Atria\_state := TRUE$
  END
```

The guards and actions of event ($Pace\_OFF\_A$) state that pacemaker's actuator ($PM\_Actuator\_A$) of atrial chamber is OFF and atrial state ($Atria\_state$) is TRUE, when pacemaker's actuator ($PM\_Actuator\_A$) of atrial is ON, pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is OFF and clock counter sp is equal to ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$).

```
EVENT Pace_ON_V
  WHEN
    grd1 : $PM\_Actuator\_V = OFF$
    grd2 : $PM\_Actuator\_A = OFF$
    grd3 : $sp = Pace\_Int$
  THEN
    act1 : $PM\_Actuator\_V := ON$
    act2 : $last\_sp := sp$
  END
```

The events $Pace\_ON\_V$ and $Pace\_OFF\_V$ also synchronize start and stop the pulse discharging into the ventricular chamber. The guards and actions of event ($Pace\_ON\_V$) state that pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON and clock counter sp assigns to a variable ($last\_sp$) when pacemaker's actuator of both chambers ($PM\_Actuator\_A$, $PM\_Actuator\_V$) is OFF and and clock counter sp is equal to the pace interval ($Pace\_Int$).

```
EVENT Pace_OFF_V
  WHEN
    grd1 : $PM\_Actuator\_V = ON$
    grd2 : $PM\_Actuator\_A = OFF$
    grd3 : $Atria\_state = TRUE$
    grd4 : $sp = Pace\_Int$
  THEN
    act1 : $PM\_Actuator\_V := OFF$
    act2 : $sp := 1$
    act3 : $Atria\_state := FALSE$
  END
```

The guards and actions of event ($Pace\_OFF\_V$) state that pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is OFF, clock counter sp resets to 1 and atrial state ($Atria\_state$) sets into TRUE state when pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON, pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF, atrial state ($Atria\_state$) is TRUE and clock counter sp is equal to the pace interval ($Pace\_Int$).

```
EVENT tic
  WHEN
    grd1 : $(sp < (Pace\_Int - FixedAV))$
           $\vee$
           $(sp \geq (Pace\_Int - FixedAV) \wedge sp < Pace\_Int \wedge$
           $Atria\_state = TRUE \wedge PM\_Actuator\_V = OFF$
           $\wedge PM\_Actuator\_A = OFF)$
  THEN
    act1 : $sp := sp + 1$
  END
```

The last event *tic* of this abstraction progressively increases the current clock counter sp under pre-defined pace interval ($Pace\_Int$). The guard of this event controls the pacing stimulus into the heart chambers (atrial and ventricular) and synchronizes ON and OFF states of pacemaker's actuator of each chamber (atrial and ventricular) under real time constraints. The guards of this event provides the required conditions to increase the current clock counter sp by 1 (ms.).

4.1.2 Abstraction of DVI mode:

In DVI operating mode of double electrode cardiac pacemaker system, the first letter 'D' represents that the pacemaker paces both atrial and ventricle, second letter 'V' represents that the pacemaker senses the ventricle only and final letter 'I' represents that the ventricular sensing inhibits atrial and ventricular pacing. In the block diagram (Fig-6) of heart pacing in DVI operating mode a ventriculoatrial (VA) interval follows the timing for each atrioventricular (AV) interval and an atrioventricular (AV) interval follows the timing for each ventriculoatrial (VA) interval, except with an R wave sensed during the ventriculoatrial (VA) interval that starts timing of a new ventriculoatrial (VA) interval.

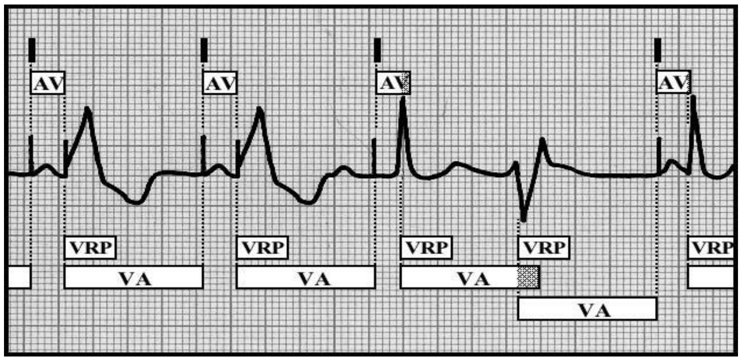

Fig. 6 Basic block diagram of ECG rhythm strip in DVI Operating Mode

In the abstract model of DVI mode, two new variables ($PM\_Actuator\_A$) and ($PM\_Actuator\_V$) represent the presence (ON) or absence (OFF) of pulse in atrial and ventricular chambers, respectively. The variable (PM_Sensor_V) also represents the presence (ON) or absence (OFF) of pacemaker's sensor in ventricular chamber. An interval between two paces is defined by a new variable $Pace\_Int$ that must be between upper rate interval (URI) and lower rate interval (LRI), is represented by an invariant ($inv3$). The variable $Pace\_Int$ is an interval between two paces of ventricular chamber that is initialized by the system before start the pacing. This interval is equal to atrioventricular(AV) interval plus ventriculoatrial(VA) interval. A variable *sp* (since pace) represents the current *clock counter*. A variable *last_sp* represents the last interval (in ms.) between two paces and a safety property in invariant ($inv6$) states that last interval must be greater than or equal to VRP and less than and equal to LRI. In invariant ($inv7$) a new variable AV_Count_STATE is used as boolean type to control the counting interval of atrioventricular (AV) interval. The variable (AV_Count) define as natural number to count the atrioventricular (AV) interval. A invariant ($inv9$) represents that the safety property and states that during the ventricular refractory period (VRP), pacemaker's actuator ($PM\_Actuator\_A$, $PM\_Actuator\_V$) of atrial and ventricular chambers are OFF and pacemaker's sensor (PM_Sensor_V) of ventricular chamber is also OFF. The last invariants ($inv10$ and $inv11$) state that pacemaker's actuator of atria is ON when clock counter *sp* is greater than or equal to ventriculoatrial (VA) interval $Pace\_Int - FixedAV$ and pacemaker's actuator of ventricular is ON when clock counter *sp* is equal to the pace interval *Pace_Int*, respectively.

$inv1 : PM\_Actuator\_A \in status$
$inv2 : PM\_Actuator\_V \in status$
$inv3 : PM\_Sensor\_V \in status$
$inv4 : Pace\_Int \in URI \,..\, LRI$
$inv5 : sp \in 1 \,..\, Pace\_Int$
$inv6 : last\_sp \geq VRP \wedge last\_sp \leq LRI$
$inv7 : AV\_Count\_STATE \in BOOL$
$inv8 : AV\_Count \in \mathbb{N}$
$inv9 : sp < VRP \Rightarrow PM\_Actuator\_A = OFF \wedge$
$\quad PM\_Actuator\_V = OFF \wedge PM\_Sensor\_V = OFF$
$inv10 : PM\_Actuator\_A = ON \Rightarrow$
$\quad sp \geq Pace\_Int - FixedAV$
$inv11 : PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int$

In the abstract specification of DVI operating mode, there are eight events $Actuator\_ON\_A$ to start pacing in atria, $Actuator\_OFF\_A$ to stop pacing in atria, $Actuator\_ON\_V$ to start pacing in ventricular, $Actuator\_OFF\_V$ to stop pacing in ventricular, $Sensor\_ON\_V$ to star sensing in ventricular, $Sensor\_OFF\_V$ to stop sensing in ventricular, tic to increment the current clock counter sp under real time constraints and tic_AV to count the atrioventricular (AV) interval.

```
EVENT Actuator_ON_V
  WHEN
    grd1 : $PM\_Actuator\_V = OFF$
    grd2 : $(sp = Pace\_Int)$
    grd3 : $sp \geq VRP$
  THEN
    act1 : $PM\_Actuator\_V := ON$
    act2 : $last\_sp := sp$
  END
```

The events $Actuator\_ON\_V$ and $Actuator\_OFF\_V$ start and stop the pacemaker's actuator in ventricular chamber and synchronizes ON and OFF states. The guards of event $Actuator\_ON\_V$ represent that when pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is OFF, current clock counter sp is equal to pace interval ($Pace\_Int$) and greater than or equal to VRP. The actions represent that if all guards are satisfy then the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON and clock counter sp assigns to a new variable ($last\_sp$).

```
EVENT Actuator_OFF_V
  WHEN
    grd1 : $PM\_Actuator\_V = ON$
    grd2 : $(sp = Pace\_Int)$
    grd3 : $AV\_Count \geq FixedAV$
  THEN
    act1 : $PM\_Actuator\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $PM\_Sensor\_V := OFF$
    act5 : $sp := 1$
    act6 : $PM\_Actuator\_A := OFF$
  END
```

The guards of event *Actuator_OFF_V* states that pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON and clock counter *sp* is equal to pace interval (*Pace_Int*) and atrioventricular (AV) counter (*AV_Count*) is greater than or equal to atrioventricular (AV) interval ($FixedAV$). The actions of this event reset all the required parameters of cardiac pacemaker system. The actions ($act1-act6$) of this event state that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular sets in OFF state, assigns the value of variable (*AV_count*) as 0, atrioventricular (AV) counter state (AV_Count_STATE) sets in FALSE state, pacemaker's sensor (PM_Sensor_V) of ventricular sets in OFF state, assigns the value of clock counter *sp* as 1 and sets in OFF state of pacemaker's actuator ($PM\_Actuator\_A$) of atrial.

```
EVENT Actuator_ON_A
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq Pace\_Int - FixedAV \wedge sp \geq VRP \wedge sp < Pace\_Int$
    grd3 : $PM\_Actuator\_A = OFF$
    grd4 : $AV\_Count\_STATE = FALSE$
  THEN
    act1 : $PM\_Actuator\_A := ON$
    act2 : $PM\_Sensor\_V := OFF$
  END
```

The actions ($act1, act2$) of event (*Actuator_ON_A*) state that the pacemaker's actuator ($PM\_Actuator\_A$) of atria sets in ON state and pacemaker's sensor (PM_Sensor_V) of ventricular sets in OFF state when all guards satisfy. The first guard of this event states that pacemaker's sensor (PM_Sensor_V) of ventricular is ON, the next guard ($grd2$) states that clock counter *sp* is greater than or equal to ventriculoatrial (VA) interval, VRP and less than pace interval (*Pace_Int*)), the third guard shows that the pacemaker's actuator

($PM\_Actuator\_A$) of atrial is OFF and and in last guard states that atrioventricular (AV) counter state (AV_Count_STATE) is FALSE.

```
EVENT Actuator_OFF_A
  WHEN
    grd1 : $PM\_Actuator\_A = ON$
    grd2 : $sp \geq Pace\_Int - FixedAV \wedge sp \geq VRP \wedge sp < Pace\_Int$
    grd3 : $AV\_Count\_STATE = FALSE$
  THEN
    act1 : $PM\_Actuator\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END
```

The actions ($act1, act2$) of event ($Actuator\_OFF\_A$) state that pacemaker's actuator ($PM\_Actuator\_A$) of atria is OFF and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE. The guards ($grd1 - grd2$) of this event state that pacemaker's actuator ($PM\_Actuator\_A$) of atria is ON, clock counter sp is greater than or equal to ventriculoatrial (VA) interval, VRP and less than pace interval ($Pace\_Int$). The last guard shows that atrioventricular (AV) counter state (AV_Count_STATE) is FALSE.

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

The events ($Sensor\_ON\_V$ and $Sensor\_OFF\_V$) is used to control the sensing activities from the ventricular chamber. The pacemaker's sensor (PM_Sensor_V) of ventricular chamber synchronizes the ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_V$) represents that if pacemaker's sensor (PM_Sensor_V) of ventricular is OFF and a guard ($grd2$) represents that current clock counter sp is less than ventriculoatrial (VA) interval or greater than or equal to ventriculoatrial (VA) interval and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE. The last guard ($grd3$) represents that current clock counter sp is greater than or equal to VRP. If all guards are true then in action part of this event pacemaker's sensor (PM_Sensor_V) of ventricular is ON.

```
EVENT Sensor_OFF_V
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq VRP \wedge sp < Pace\_Int$
  THEN
    act1 : $PM\_Sensor\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $last\_sp := sp$
    act5 : $sp := 1$
    act6 : $PM\_Actuator\_V := OFF$
    act7 : $PM\_Actuator\_A := OFF$
  END
```

The event ($Sensor\_OFF\_V$) is used to set the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF. The guards of this event represent that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON and current clock counter sp is greater than or equal to VRP and less than pace interval ($Pace\_Int$). All actions of this event is same as event ($Actuator\_OFF\_V$) of DOO operating mode, which are already described. Here only extra action ($act1$) is added to set the pacemaker's sensor (PM_Sensor_V) of ventricular is in OFF state.

```
EVENT tic
  WHEN
    grd1 : $(sp < VRP \wedge PM\_Sensor\_V = OFF)$
           $\vee$
           $(sp \geq VRP \wedge sp < Pace\_Int \wedge PM\_Sensor\_V = ON \wedge$
           $AV\_Count\_STATE = FALSE)$
  THEN
    act1 : $sp := sp + 1$
  END
```

The event (tic) of this abstraction progressively increases the current clock counter sp under pre-defined pace interval ($Pace\_Int$). The guard ($grd1$) of this event control the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_A, PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus of ventricular chamber and synchronizes the ON and OFF states of ventricular pacemaker's sensor (PM_Sensor_V) under real time constraints.

```
EVENT tic_AV
  WHEN
    grd1 : $AV\_Count \leq FixedAV$
    grd2 : $AV\_Count\_STATE = TRUE$
    grd3 : $sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int$
  THEN
    act1 : $AV\_Count := AV\_Count + 1$
    act2 : $sp := sp + 1$
  END
```

The last event (*tic_AV*) of this abstraction progressively counts the atrioventricular (AV) interval and also increases the current clock counter *sp* is represented in actions ($act1$ and $act2$) . The guards ($grd1 - grd3$) of this event states that atrioventricular (AV) counter (*AV_Count*) is less than and equal to atrioventricular (AV) interval (*FixedAV*), atrioventricular (AV) counter state (*AV_count_STATE*) is in TRUE state and current clock counter *sp* is within the atrioventricular (AV) interval.

4.1.3 Abstraction of DDI mode:

In DDI operating mode of double electrode pacemaker system, the first letter 'D' represents that the pacemaker paces both atrial and ventricle, second letter 'D' represents that the pacemaker senses both atrial and ventricle and final letter 'I' represents two conditional meaning that depends on atrial and ventricular sensing; first is that atrial sensing inhibits atrial pacing and does not trigger ventricular pacing and second is that ventricular sensing inhibits ventricular and atrial pacing. In the block diagram (Fig-7) of heart pacing in DDI operating mode a new LRI follows the timing of each preceding LRI. The timing of an atrioventricular (AV) interval occurs within this period only following a completed ventriculoatrial (VA) interval (i.e., when no atrial sensing occurs).

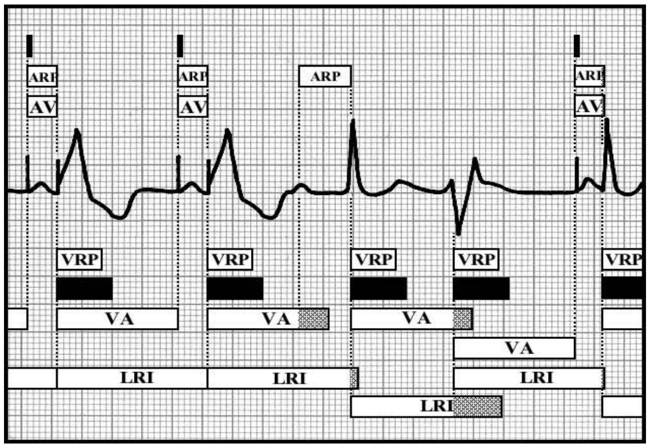

Fig. 7 Basic block diagram of ECG rhythm strip in DDI Operating Mode

In this abstract model, we have formalized a bradycardia operating mode *DDI* of the double electrode pacemaker. In this operating mode, the pacemaker uses actuators and sensors in both chambers. We have defined two new variables $PM\_Actuator\_A$ and $PM\_Actuator\_V$ that represent ON or OFF states of pacemaker's actuators for pacing in the atrial and ventricular chambers. Similarly next two variables PM_Sensor_A and PM_Sensor_V represent ON or OFF states of pacemaker's sensor for sensing an intrinsic pulse from both the atrial and ventricular chambers. An interval between two paces is defined by a new variable *Pace_Int* that must be between upper rate interval (URI) and lower rate interval (LRI), is represented by an invariant ($inv5$). A variable *sp* (since pace) represents the current *clock counter*. A variable *last_sp* represents the last interval (in ms.) between two paces and a safety property in invariant ($inv7$) states that last interval must be between PVARP and pace interval *Pace_Int*. Another new variable AV_Count_STATE is defined as boolean type to control the atrioventricular (AV) interval state and next variable AV_Count is defined as natural number to count the atrioventricular (AV) interval. Extra two new invariants ($inv11$ and $inv12$) represent the safety properties. The invariant ($inv11$) states that when clock counter *sp* is less than ventricular refractory period (VRP) and atrioventricular (AV) counter state is FALSE then the pacemaker's actuators and sensors of both chambers are OFF. The next invariant ($inv12$) represents that the pacemaker's actuator of ventricular is ON when clock counter *sp* is equal to the pace interval *Pace_Int*.

$$
\begin{array}{l}
inv1 : PM\_Actuator\_A \in status \\
inv2 : PM\_Actuator\_V \in status \\
inv3 : PM\_Sensor\_A \in status \\
inv4 : PM\_Sensor\_V \in status \\
inv5 : Pace\_Int \in URI \mathrel{..} LRI \\
inv6 : sp \in 1 \mathrel{..} Pace\_Int \\
inv7 : last\_sp \geq PVARP \land last\_sp \leq Pace\_Int \\
inv8 : AV\_Count\_STATE \in BOOL \\
inv9 : AV\_Count \in \mathbb{N} \\
inv10 : Pace\_Int - FixedAV < Pace\_Int \\
inv11 : sp < VRP \land AV\_Count\_STATE = FALSE \\
\qquad\quad \Rightarrow \\
\qquad\quad PM\_Actuator\_A = OFF \land \\
\qquad\quad PM\_Actuator\_V = OFF \land \\
\qquad\quad PM\_Sensor\_A = OFF \land \\
\qquad\quad PM\_Sensor\_V = OFF \\
inv12 : PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int
\end{array}
$$

In the abstract specification of *DDI* operating mode, there are ten events $Actuator\_ON\_A$ to start pacing in atria, $Actuator\_OFF\_A$ to stop pacing in atria, $Actuator\_ON\_V$ to start pacing in ventricular, $Actuator\_OFF\_V$ to stop pacing in ventricular, $Sensor\_ON\_V$ to start sensing in ventricular, $Sensor\_OFF\_V$

to stop sensing in ventricular, $Sensor\_ON\_A$ to star sensing in atrial, $Sensor\_OFF\_A$ to stop sensing in atrial, tic to increment the current clock counter sp under real time constraints and tic_AV to count the atrioventricular (AV) interval.

```
EVENT Actuator_ON_V
  WHEN
    grd1 : $PM\_Actuator\_V = OFF$
    grd2 : $(sp = Pace\_Int)$
    grd3 : $sp \geq VRP \wedge sp \geq PVARP$
  THEN
    act1 : $PM\_Actuator\_V := ON$
    act2 : $last\_sp := sp$
  END
```

The events $Actuator\_ON\_V$ and $Actuator\_OFF\_V$ start and stop the pacemaker's actuator $(PM\_Actuator\_V)$ in ventricular chamber and synchronizes ON and OFF states. The guards of event $(Actuator\_ON\_V)$ represent that when pacemaker's actuator $(PM\_Actuator\_V)$ of ventricular is OFF, current clock counter sp is equal to pace interval $(Pace\_Int)$, in next guard current clock counter sp is greater then or equal to ventricular refractory period (VRP) and post ventricular refractory period (PVARP). The actions of this event represent that if all guards are satisfy then the pacemaker's actuator $(PM\_Actuator\_V)$ of ventricular is ON and current clock counter sp assigns to a variable $(last\_sp)$.

```
EVENT Actuator_OFF_V
  WHEN
    grd1 : $PM\_Actuator\_V = ON$
    grd2 : $(sp \geq Pace\_Int) \wedge sp \geq VRP \wedge sp \geq PVARP$
    grd3 : $AV\_Count\_STATE = TRUE$
    grd4 : $PM\_Sensor\_A = OFF$
    grd5 : $PM\_Actuator\_A = OFF$
  THEN
    act1 : $PM\_Actuator\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $PM\_Sensor\_V := OFF$
    act5 : $sp := 1$
  END
```

The guards $(grd1, grd2)$ of event $Actuator\_OFF\_V$ state that pacemaker's actuator $(PM\_Actuator\_V)$ of ventricular is ON and current clock counter sp is greater than or equal to pace interval $(Pace\_Int)$, VRP and PVARP. The third guard $(grd3)$ states that atrioventricular (AV) counter state (AV_Count_STATE) is TRUE and last two guards state that the pacemaker's sensor and actuator $(PM\_Sensor\_A, PM\_Actuator\_A)$ of atrial are OFF. The actions of this event reset all the parameters of operating mode of the pacemaker system. The actions ($act1-act5$) of this event state that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular sets OFF, assigns the value of variable (AV_count) as 0, atrioventricular (AV) counter state (AV_Count_STATE) sets FALSE, the pacemaker's sensor (PM_Sensor_V) of ventricular sets OFF and assigns the value of the current clock counter sp as 1.

```
EVENT Actuator_ON_A
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq Pace\_Int - FixedAV \land sp \geq VRP \land sp \geq PVARP$
    grd3 : $PM\_Actuator\_A = OFF$
    grd4 : $PM\_Sensor\_A = ON$
  THEN
    act1 : $PM\_Actuator\_A := ON$
    act2 : $PM\_Sensor\_V := OFF$
    act3 : $PM\_Sensor\_A := OFF$
  END
```

The actions ($act1 - act3$) of event ($Actuator\_ON\_A$) state that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial sets ON and the pacemaker's sensor (PM_Sensor_V, PM_Sensor_A) of ventricular and atrial set OFF when all guards satisfy. The first guard of this event states that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON, the next guard ($grd2$) states that current clock counter sp is greater than or equal to ventriculoatrial (VA) interval, VRP and PVARP, the third guard shows that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF and and in last guard states that the pacemaker's sensor (PM_Sensor_A) of atrial is ON.

```
EVENT Actuator_OFF_A
  WHEN
    grd1 : $PM\_Actuator\_A = ON$
    grd2 : $sp = Pace\_Int - FixedAV \land sp \geq VRP \land sp \geq PVARP$
    grd3 : $AV\_Count\_STATE = FALSE$
  THEN
    act1 : $PM\_Actuator\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END
```

The actions ($act1, act2$) of event ($Actuator\_OFF\_A$) state that the pacemaker's actuator ($PM\_Actuator\_A$) of atria sets in OFF state and atrioventricular (AV) counter state (AV_Count_STATE) sets in TRUE state. The guards ($grd1, grd2$) of this event state that the pacemaker's actuator ($PM\_Actuator\_A$) of atria is ON, current clock counter sp is greater than or equal to ventriculoatrial (VA) interval, VRP and PVARP. In last guard states that atrioventricular (AV) counter states (AV_Count_STATE) is FALSE.

```
EVENT Sensor_ON_A
  WHEN
    grd1 : $PM\_Sensor\_A = OFF$
    grd2 : $sp < Pace\_Int - FixedAV \wedge sp \geq VRP \wedge sp \geq PVARP$
    grd3 : $PM\_Sensor\_V = OFF$
  THEN
    act1 : $PM\_Sensor\_A := ON$
  END
```

The events ($Sensor\_ON\_A$ and $Sensor\_OFF\_A$) is used to control the sensing activities from the atrial chamber. The pacemaker's sensor (PM_Sensor_A) of atrial chamber synchronizes ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_A$) represents that if the pacemaker's sensor (PM_Sensor_A) of atrial is OFF and next guard ($grd2$) represents that the current clock counter sp is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP. The last guard ($grd3$) represents that the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF. If all guards are true then in action part of this event the pacemaker's sensor (PM_Sensor_A) of atrial sets ON state.

```
EVENT Sensor_OFF_A
  WHEN
    grd1 : $PM\_Sensor\_A = ON$
    grd2 : $sp < Pace\_Int - FixedAV \wedge sp \geq VRP \wedge sp \geq PVARP$
  THEN
    act1 : $PM\_Sensor\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END
```

The event ($Sensor\_OFF\_A$) is used to set the pacemaker's sensor (PM_Sensor_A) of atrial in OFF state. The guards of this event represent that the pacemaker's sensor (PM_Sensor_A) of atrial is ON and current clock counter sp is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP. In actions of this event state that the pacemaker's sensor (PM_Sensor_A) of atrial sets OFF and atrioventricular (AV) counter state sets TRUE.

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

The events ($Sensor\_ON\_V$ and $Sensor\_OFF\_V$) is used to control the sensing activities from the ventricular chamber. The pacemaker's sensor (PM_Sensor_V) of ventricular chamber synchronizes ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_V$) represents that if the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF and in next guard ($grd2$) shows that current clock counter *sp* is greater than or equal to VRP, less than ventriculoatrial (VA) interval and the pacemaker's sensor (PM_Sensor_A) of atrial is ON, or greater than or equal to ventriculoatrial (VA) interval and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE. The last guard ($grd3$) states that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF. If all guards are true then in action part of this event, the pacemaker's sensor (PM_Sensor_V) of ventricular sets in ON state.

```
EVENT Sensor_OFF_V
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq VRP \wedge sp \geq PVARP$
    grd3 : $(sp < Pace\_Int - FixedAV)$
           $\vee$
           $(sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int)$
    grd4: $PM\_Actuator\_V = OFF$
    grd5: $PM\_Actuator\_A = OFF$
  THEN
    act1 : $PM\_Sensor\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $last\_sp := sp$
    act5 : $sp := 1$
    act6 : $PM\_Sensor\_A := OFF$
  END
```

The event ($Sensor\_OFF\_V$) is used to set the pacemaker's sensor (PM_Sensor_V) of ventricular in OFF state. The guards ($grd1, grd2$) of this event represent that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON and current clock counter *sp* is greater than or equal to VRP and PVARP. The next guard ($grd3$) represents that the current clock counter *sp* is less than ventriculoatrial (VA) interval or greater than or equal to ventriculoatrial (VA) interval and less than pace interval ($Pace\_Int$). The last two guards ($grd4, grd5$) state that the pacemaker's actuator ($PM\_Actuator\_V, PM\_Actuator\_A$) of ventricular and atrial are OFF. The actions ($act1 - act6$) of this event state that the pacemaker's sensor (PM_Sensor_V) of ventricular sets OFF, assigns the value of variable (AV_count) as 0, atrioventricular (AV) counter state (AV_Count_STATE) sets FALSE, assigns the value of the current clock counter *sp* to new variable ($last\_sp$), assigns the value of clock counter *sp* as 1 and sets OFF state of the pacemaker's actuator ($PM\_Actuator\_A$) of the atrial chamber.

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

The event *tic* of this abstraction progressively increases the current clock counter *sp* under pre-defined pace interval ($Pace\_Int$). The guards of this event control the pacing stimulus into the heart chambers (atria and ventricular), synchronize ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_A$, $PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus from atrial and ventricular chambers and synchronize ON and OFF states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) in both chambers under real time constraints.

```
EVENT tic_AV
  WHEN
    grd1 : $AV\_Count \leq FixedAV$
    grd2 : $AV\_Count\_STATE = TRUE$
    grd3 : $sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int$
  THEN
    act1 : $AV\_Count := AV\_Count + 1$
    act2 : $sp := sp + 1$
  END
```

The last event (tic_AV) of this abstraction progressively counts the atrioventricular (AV) interval and also increases the current clock counter *sp* is represented in actions ($act1$ and $act2$) . The guards of this event state that atrioventricular (AV) counter (AV_Count) is less than and equal to atrioventricular (AV) interval ($FixedAV$), atrioventricular (AV) state (AV_count_STATE) is

TRUE and the current clock counter *sp* is within the atrioventricular (AV) interval.

4.1.4 Abstraction of VDD mode:

In VDD operating mode of the double electrode pacemaker system, the first letter 'V' represents that the pacemaker paces ventricle only, second letter 'D' represents that the pacemaker senses both atrial and ventricle and final letter 'D' represents two conditional meaning that depends on atrial and ventricular sensing; first is that atrial sensing triggers ventricular pacing and second is that ventricular sensing inhibits ventricular pacing. In the block diagram (Fig-8) of heart pacing in VDD operating mode a new LRI follows the timing of each preceding LRI. When a sensed P wave occurs an atrioventricular (AV) interval is triggered within.

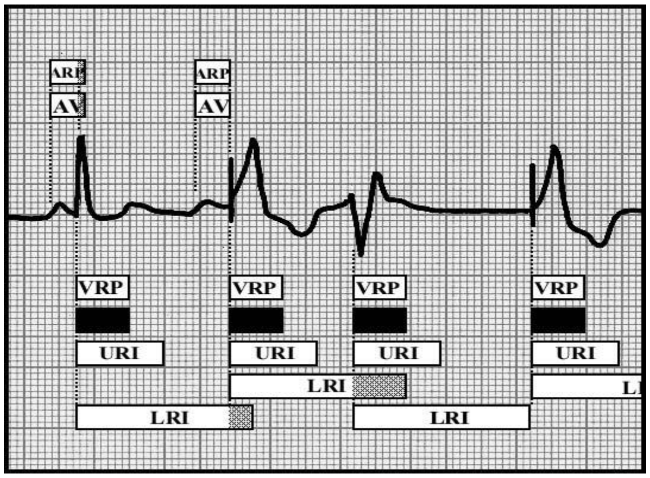

Fig. 8 Basic block diagram of ECG rhythm strip in VDD Operating Mode

In this abstract model, we have formalized a bradycardia operating mode *VDD* of the double electrode pacemaker. In this operating mode, the pacemaker uses actuators and sensors in both chambers. We have defined a new variable *PM_Actuator_V* that represents ON or OFF states of pacemaker's actuators for pacing in the ventricular chamber. Next two variables *PM_Sensor_A* and *PM_Sensor_V* represent the ON or OFF states of pacemaker's sensor for sensing an intrinsic pulse from both the atrial and ventricular chambers. An interval between two paces is defined by a new variable *Pace_Int* that must be between upper rate interval (URI) and lower rate interval (LRI), is represented by an invariant (*inv*4). A variable *sp* (since pace) represents the current *clock counter*. A variable *last_sp* represents the last interval (in ms.) between two paces and a safety property in invariant (*inv*6) states that last interval must be between PVARP and pace interval *Pace_Int*. Another new variable *AV_Count_STATE* in invariant (*inv*7) is defined as boolean type to control the atrioventricular (AV) interval state and next variable *AV_Count* is defined as natural number to count the atrioventricular (AV) interval by invariant ($inv8$). Here new invariant ($inv10$) represents the safety property and states that when clock counter *sp* is less then ventricular refractory period (VRP) and atrioventricular (AV) counter state (AV_Count_STATE) is $TRUE$, the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is OFF and pacemaker's sensors of both chambers are OFF. The next invariant ($inv11$) represents that pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON when clock counter *sp* is either equal to the pace interval $Pace\_Int$ or clock counter *sp* less than pace interval $Pace\_Int$ and atrioventricular (AV) counter AV_Count is greater then blanking period V_Blank and greater than or equal to the atrioventricular (AV) period $FixedAV$.

$$
\boxed{
\begin{array}{ll}
inv1: & PM\_Actuator\_V \in status \\
inv2: & PM\_Sensor\_A \in status \\
inv3: & PM\_Sensor\_V \in status \\
inv4: & Pace\_Int \in URI \,..\, LRI \\
inv5: & sp \in 1 \,..\, Pace\_Int \\
inv6: & last\_sp \geq PVARP \land last\_sp \leq Pace\_Int \\
inv7: & AV\_Count\_STATE \in BOOL \\
inv8: & AV\_Count \in \mathbb{N} \\
inv9: & Pace\_Int - FixedAV < Pace\_Int \\
inv10: & sp < VRP \land AV\_Count\_STATE = FALSE \\
 & \Rightarrow \\
 & PM\_Actuator\_V = OFF \land \\
 & PM\_Sensor\_A = OFF \land \\
 & PM\_Sensor\_V = OFF \\
inv11: & PM\_Actuator\_V = ON \\
 & \Rightarrow \\
 & (sp = Pace\_Int \\
 & \lor \\
 & (sp < Pace\_Int \land \\
 & AV\_Count > V\_Blank \land \\
 & AV\_Count \geq FixedAV))
\end{array}
}
$$

In the abstract specification of VDD operating mode, there are eight events $Actuator\_ON\_V$ to start pacing in ventricular, $Actuator\_OFF\_V$ to stop pacing in ventricular, $Sensor\_ON\_V$ to star sensing in ventricular, $Sensor\_OFF\_V$ to stop sensing in ventricular, $Sensor\_ON\_A$ to star sensing in atrial, $Sensor\_OFF\_A$ to stop sensing in atrial, *tic* to increment the current clock counter under real time constraints and tic_AV to count the atrioventricular (AV) interval.

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

The events *Actuator_ON_V* and *Actuator_OFF_V* start and stop the pacemaker's actuator (*PM_Actuator_V*) in ventricular chamber and synchronizes ON and OFF states. The guard ($grd1$) of event *Actuator_ON_V* represents that when the pacemaker's actuator (*PM_Actuator_V*) of ventricular is OFF and in guard ($grd2$) the current clock counter *sp* is equal to pace interval (*Pace_Int*) or less than pace interval *Pace_Int* and atrioventricular (AV) counter (*AV_Count*) is greater than blanking period (*V_Blank*) and greater than or equal to atrioventricular (AV) interval (*FixedAV*). In the last guard, the current clock counter *sp* is greater than or equal to ventricular refractory period (VRP) and post ventricular refractory period (PVARP). The actions of this event represent that if all guards are satisfy then the pacemaker's actuator (*PM_Actuator_V*) of ventricular is ON and clock counter *sp* assigns to a new variable (*last_sp*).

```
EVENT Actuator_OFF_V
  WHEN
    grd1 : $PM\_Actuator\_V = ON$
    grd2 : $(sp = Pace\_Int)$
           $\vee$
           $(sp \geq VRP \wedge sp < Pace\_Int \wedge AV\_Count > V\_Blank \wedge$
           $AV\_Count \geq FixedAV \wedge AV\_Count\_STATE = TRUE)$
    grd2 : $(sp \geq PVARP)$
  THEN
    act1 : $PM\_Actuator\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $PM\_Sensor\_V := OFF$
    act5 : $PM\_Sensor\_A := OFF$
    act6 : $last\_sp := sp$
    act7 : $sp := 1$
  END
```

The guards of event *Actuator_OFF_V* states that the pacemaker's actuator (*PM_Actuator_V*) of ventricular is ON and current clock counter *sp* is equal to the pace interval *Pace_Int* or greater than or equal to VRP, less than pace interval (*Pace_Int*), atrioventricular (AV) counter (*AV_Count*) is greater than blanking period (*V_Blank*), atrioventricular (AV) counter (*AV_Count*) is greater than or equal to the atrioventricular (AV)interval (*FixedAV*) and atrioventricular (AV) counter state (*AV_Count_State*) is TRUE. The last guard states that current clock counter *sp* is greater than or equal to PVARP. The actions of this event reset the all parameters of the pacemaker for beginning the pacing cycle. In action (*act*1) sets OFF state of the pacemaker's actuator (*PM_Actuator_V*) of ventricular, in the second action reassign the value of variable (*AV_count*) as 0, in next action (*act*3) sets FALSE state to the AV counter state *AV_Count_STATE*, sets OFF state to the pacemaker's sensor (*PM_Sensor_V*, *PM_Sensor_A*) of both chambers in actions (*act*4, *act*5), in action (*act*6) assigns the value of the current clock counter *sp* to new variable *last_sp* and finally in action (*act*7) assigns the value of the current clock counter *sp* as 1.

EVENT Sensor_ON_A
  WHEN
    grd1 : $PM\_Sensor\_A = OFF$
    grd2 : $sp < Pace\_Int - FixedAV \land sp \geq VRP \land sp \geq PVARP$
    grd3 : $PM\_Sensor\_V = OFF$
  THEN
    act1 : $PM\_Sensor\_A := ON$
  END

The events (*Sensor_ON_A* and *Sensor_OFF_A*) is used to control the sensing activities from the atrial chamber. The pacemaker's sensor (*PM_Sensor_A*) of atrial chamber synchronizes ON and OFF states under real time constraints. The guard (*grd*1) of event (*Sensor_ON_A*) represents that if the pacemaker's sensor (*PM_Sensor_A*) of atrial is OFF and guard (*grd*2) represents that the current clock counter *sp* is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP. The last guard (*grd*3) represents that the pacemaker's sensor (*PM_Sensor_V*) of ventricular is OFF. If all guards are true then in action part of this event then the pacemaker's sensor (*PM_Sensor_A*) of atrial sets in ON state.

EVENT Sensor_OFF_A
  WHEN
    grd1 : $PM\_Sensor\_A = ON$
    grd2 : $sp < Pace\_Int - FixedAV \land sp \geq VRP \land sp \geq PVARP$
  THEN
    act1 : $PM\_Sensor\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END

The event ($Sensor\_OFF\_A$) is used to set the pacemaker's sensor (PM_Sensor_A) of atrial in OFF state. The guards of this event show that the the pacemaker's sensor (PM_Sensor_A) of atrial is ON and the current clock counter sp is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP. In actions of this event state that the pacemaker's sensor (PM_Sensor_A) of atrial sets in OFF state and atrioventricular (AV) counter state (AV_Count_STATE) sets in TRUE state.

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

The events ($Sensor\_ON\_V$ and $Sensor\_OFF\_V$) is used to control the sensing activities from the ventricular chamber. The pacemaker's sensor (PM_Sensor_V) of ventricular chamber synchronizes ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_V$) represents that if the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF and in next guard ($grd2$) represents that the current clock counter sp is greater than or equal to VRP, less than ventriculoatrial (VA) interval and the pacemaker's sensors (PM_Sensor_A) of atrial is ON or current clock counter sp is greater than or equal to ventriculoatrial (VA) interval and atrioventricular (AV) counter state (AV_Count_STATE) is in TRUE state. If all guards are true then in action part of this event, the pacemaker's sensor (PM_Sensor_V) of ventricular sets in ON state.

```
EVENT Sensor_OFF_V
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq VRP \wedge sp \geq PVARP$
    grd3 : $(sp < Pace\_Int - FixedAV)$
           $\vee$
           $(sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int)$
    grd4: $PM\_Actuator\_V = OFF$
  THEN
    act1 : $PM\_Sensor\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $last\_sp := sp$
    act5 : $sp := 1$
    act6 : $PM\_Sensor\_A := OFF$
  END
```

The event ($Sensor\_OFF\_V$) is used to set the pacemaker's sensor (PM_Sensor_V) of ventricular in OFF state. The guard ($grd1$) of this event represents that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON and the current clock counter *sp* is greater than or equal to VRP and PVARP interval. The next guard ($grd3$) represents that the current clock counter *sp* is less than ventriculoatrial (VA) interval or greater than or equal to ventriculoatrial (VA) interval and less than automatic pace interval ($Pace\_Int$). The last guard ($grd4$) state that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is OFF. All actions of this event is same as event ($Sensor\_OFF\_V$) of DDI operating mode which are already described in actions part of this event ($Sensor\_OFF\_V$) .

```
EVENT tic
  WHEN
    grd1 : $(sp < VRP)$
           $\vee$
           $(sp \geq VRP \wedge sp < Pace\_Int \wedge PM\_Sensor\_V = ON \wedge$
           $PM\_Sensor\_A = ON$
  THEN
    act1 : $sp := sp + 1$
  END
```

The event (tic) of this abstraction progressively increases the current clock counter *sp* under pre-defined pace interval ($Pace\_Int$). The guard of this event control the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular chamber and also control the sensing intrinsic stimulus of the atrial and ventricular chamber and synchronize ON and OFF states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) in both chambers under real time constraints.

```
EVENT tic_AV
  WHEN
    grd1 : $AV\_Count < FixedAV$
    grd2 : $AV\_Count\_STATE = TRUE$
    grd3 : $(sp \geq VRP \land sp \geq PVARP \land sp < Pace\_Int - FixedAV)$
           $\lor$
           $(sp \geq Pace\_Int - FixedAV \land sp < Pace\_Int)$
  THEN
    act1 : $AV\_Count := AV\_Count + 1$
    act2 : $sp := sp + 1$
  END
```

The last event (tic_AV) of this abstraction progressively counts the atrioventricular (AV) interval and also increases the current clock counter sp is represented in actions ($act1$ and $act2$) . The guards of this event states that atrioventricular (AV) counter (AV_Count) is less than atrioventricular (AV) interval ($FixedAV$), atrioventricular (AV) state (AV_count_STATE) is in TRUE state and the current clock counter sp is within the ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$) and greater than or equal to VRP and PVARP interval or clock counter sp is greater than or equal to atrioventricular (AV) interval and less than pace interval ($Pace\_Int$).

4.1.5 Abstraction of DDD mode:

In DDD operating mode of double electrode pacemaker system, the first letter 'D' represents that the pacemaker paces in both atrial and ventricle chambers, second letter 'D' represents that the pacemaker senses intrinsic activities from both atrial and ventricle chambers and final letter 'D' represents two conditional meaning that depends on atrial and ventricular sensing; first is that atrial sensing inhibits atrial pacing and triggers ventricular pacing and second is that ventricular sensing inhibits ventricular and atrial pacing. In the block diagram (Fig-9) of heart pacing in DDD operating mode a ventriculoatrial (VA) interval follows the timing for each atrioventricular (AV) interval and an atrioventricular (AV) interval follows the timing for each ventriculoatrial (VA) interval, except with a 'P' wave or an 'R' wave (PVC) that starts timing of a new ventriculoatrial (VA) interval.

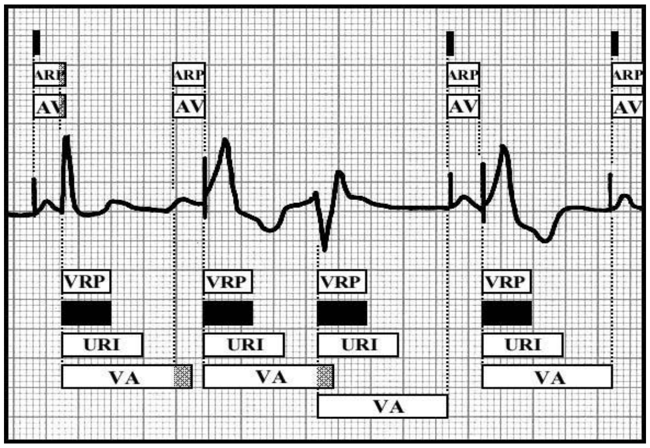

Fig. 9 Basic block diagram of ECG rhythm strip in DDD Operating Mode

In this abstract model, we have formalized a bradycardia operating mode *DDD* of the double electrode pacemaker. In this operating mode, the pacemaker uses actuators and sensors in both chambers. We have defined two new variables $PM\_Actuator\_A$ and $PM\_Actuator\_V$ that represent ON or OFF states of pacemaker's actuators for pacing in the atrial and ventricular chambers. Similarly next two variables PM_Sensor_A and PM_Sensor_V represent ON or OFF states of pacemaker's sensor for sensing an intrinsic pulse from both the atrial and ventricular chambers. An interval between two paces is defined by a new variable $Pace\_Int$ that must be between upper rate interval (URI) and lower rate interval (LRI), is represented by an invariant ($inv5$). A variable sp (since pace) represents the current *clock counter*. A variable $last\_sp$ represents the last interval (in ms.) between two paces and a safety property in invariant ($inv7$) states that last interval must be between PVARP and pace interval $Pace\_Int$. Another new variable AV_Count_STATE is defined as boolean type to control the atrioventricular (AV) interval state and next variable AV_Count is defined as natural number to count the atrioventricular (AV) interval. The invariants ($inv11$,$inv12$ and $inv13$) represent the safety properties. The invariant $inv11$ states that when clock counter sp is less than ventricular refractory period (VRP) and atrioventricular (AV) counter state AV_Count_State is $FALSE$, pacemaker's actuators and sensors of both chambers are OFF. Similarly,the next invariants ($inv12$ and $inv13$) represent the conditions of ON state of the pacemaker's actuators in both chambers

$inv1 : PM\_Actuator\_A \in status$
$inv2 : PM\_Actuator\_V \in status$
$inv3 : PM\_Sensor\_A \in status$
$inv4 : PM\_Sensor\_V \in status$
$inv5 : Pace\_Int \in URI \,..\, LRI$
$inv6 : sp \in 1 \,..\, Pace\_Int$
$inv7 : last\_sp \geq PVARP \land last\_sp \leq Pace\_Int$
$inv8 : AV\_Count\_STATE \in BOOL$
$inv9 : AV\_Count \in \mathbb{N}$
$inv10 : Pace\_Int - FixedAV < Pace\_Int$
$inv11 : sp < VRP \land AV\_Count\_STATE = FALSE \Rightarrow$
$\quad PM\_Actuator\_V = OFF \land PM\_Sensor\_A = OFF \land$
$\quad PM\_Sensor\_V = OFF \land PM\_Actuator\_A = OFF$
$inv12 : PM\_Actuator\_V = ON \Rightarrow$
$\quad sp = Pace\_Int \lor (sp < Pace\_Int \land$
$\quad AV\_Count > V\_Blank \land AV\_Count \geq FixedAV)$
$inv13 : PM\_Actuator\_A = ON \Rightarrow$
$\quad (sp \geq Pace\_Int - FixedAV)$

In the abstract specification of *DDD* operating mode, there are ten events *Actuator_ON_A* to start pacing in atrial, *Actuator_OFF_A* to stop pacing in atrial, *Actuator_ON_V* to start pacing in ventricular, *Actuator_OFF_V* to stop pacing in ventricular, *Sensor_ON_V* to start sensing in ventricular, *Sensor_OFF_V* to stop sensing in ventricular, *Sensor_ON_A* to star sensing in atrial, *Sensor_OFF_A* to stop sensing in atrial, *tic* to increment the current clock counter *sp* under real time constraints and *tic_AV* to count the atrioventricular (AV) interval.

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

The events *Actuator_ON_V* and *Actuator_OFF_V* start and stop the pacemaker's actuator in ventricular chamber and synchronize ON and OFF states. The guard (*grd*1) of event *Actuator_ON_V* represents that the pacemaker's actuator (*PM_Actuator_V*) of ventricular is OFF. In guard (grd2), the current clock counter sp is equal to pace interval ($Pace\_Int$) or less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter (AV_Count) is greater than blanking period (V_Blank) and greater than or equal to atrioventricular (AV) interval ($FixedAV$). In last guard ($grd3$), the current clock counter sp is greater than or equal to ventricular refractory period (VRP) and post ventricular refractory period (PVARP). The actions represent that if all guards are satisfy then the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON and the current clock counter sp assigns to new variable ($last\_sp$).

```
EVENT Actuator_OFF_V
 WHEN
  grd1 : $PM\_Actuator\_V = ON$
  grd2 : $(sp = Pace\_Int)$
         $\vee$
         $(sp < Pace\_Int \wedge AV\_Count > V\_Blank \wedge$
         $AV\_Count \geq FixedAV$
  grd3 : $AV\_Count\_STATE = TRUE$
  grd4 : $PM\_Actuator\_A = OFF$
  grd5 : $PM\_Sensor\_A = OFF$
 THEN
  act1 : $PM\_Actuator\_V := OFF$
  act2 : $AV\_Count := 0$
  act3 : $AV\_Count\_STATE := FALSE$
  act4 : $PM\_Sensor\_V := OFF$
  act5 : $sp := 1$
 END
```

The guards of event ($Actuator\_OFF\_V$) states that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON and current clock counter sp is equal to pace interval ($Pace\_Int$) or less than pace interval ($Pace\_Int$), atrioventricular (AV) counter (AV_Count) is greater than blanking period (V_Blank) and atrioventricular (AV) counter is greater than or equal to atrioventricular (AV)interval ($FixedAV$). The guard ($grd3$) states that atrioventricular (AV) counter state (AV_Count_STATE) is TRUE and last two guards represent that the pacemaker's actuator and sensor ($PM\_Actuator\_A, PM\_Sensor\_A$) of atrial chamber are OFF. The actions of this event reset the all parameters of the pacemaker. In actions part, sets OFF state of the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular, reassigns the value of variable (AV_count) as 0, sets FALSE state to the AV counter state (AV_Count_STATE), sets an OFF state to the pacemaker's sensor (PM_Sensor_V) of ventricular chamber and finally assigns the value of the current clock counter sp as 1.

```
EVENT Actuator_ON_A
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq Pace\_Int - FixedAV \land$
           $sp \geq VRP \land sp \geq PVARP$
    grd3 : $PM\_Actuator\_A = OFF$
    grd4 : $PM\_Sensor\_A = ON$
  THEN
    act1 : $PM\_Actuator\_A := ON$
    act2 : $PM\_Sensor\_V := OFF$
    act3 : $PM\_Sensor\_A := OFF$
  END
```

The actions ($act1 - act3$) of event ($Actuator\_ON\_A$) state that the pacemaker's actuator ($PM\_Actuator\_A$) of atria is ON and pacemaker's sensors (PM_Sensor_V, PM_Sensor_A) of ventricular and atrial is OFF when all guards are satisfied. The first guard states that pacemaker's sensor (PM_Sensor_V) of ventricular is ON, the next guard ($grd2$) states that current clock counter sp is greater than or equal to ventriculoatrial (VA) interval, VRP and PVARP. The last two guards show that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF and pacemaker's sensor (PM_Sensor_A) of atrial is ON.

```
EVENT Actuator_OFF_A
  WHEN
    grd1 : $PM\_Actuator\_A = ON$
    grd2 : $AV\_Count\_STATE = FALSE$
    grd3 : $sp \geq Pace\_Int - FixedAV \land$
           $sp \geq VRP \land sp \geq PVARP$
  THEN
    act1 : $PM\_Actuator\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END
```

In event $Actuator\_OFF\_A$, the actions ($act1, act2$) state that pacemaker's actuator ($PM\_Actuator\_A$) of atria is OFF and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE when the guards are satisfied. The first two guards ($grd1, grd2$) state that pacemaker's actuator ($PM\_Actuator\_A$) of atrial is ON and atrioventricular (AV) counter state (AV_Count_STATE) is $FALSE$. The last guard represents clock counter sp is greater than or equal to ventriculoatrial (VA) interval, VRP and PVARP.

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

The events ($Sensor\_ON\_V$ and $Sensor\_OFF\_V$) is used to control the sensing activities from the ventricular chamber. The pacemaker's sensor (PM_Sensor_V) of ventricular chamber synchronizes ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_V$) represents that if the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF and next guard ($grd2$) represents that the current clock counter *sp* is greater than or equal to VRP, less than ventriculoatrial (VA) interval and the pacemaker's sensor (PM_Sensor_A) of atrial is ON or greater than or equal to ventriculoatrial (VA) interval and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE. The last guard ($grd3$) states that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF. If all guards are true then in action part of this event, the pacemaker's sensor (PM_Sensor_V) of ventricular sets in ON state.

```
EVENT Sensor_OFF_V
  WHEN
    grd1 : $PM\_Sensor\_V = ON$
    grd2 : $sp \geq VRP \wedge sp \geq PVARP$
    grd3 : $(sp < Pace\_Int - FixedAV)$
           $\vee$
           $(sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int)$
    grd4: $PM\_Actuator\_V = OFF$
    grd5: $PM\_Actuator\_A = OFF$
  THEN
    act1 : $PM\_Sensor\_V := OFF$
    act2 : $AV\_Count := 0$
    act3 : $AV\_Count\_STATE := FALSE$
    act4 : $last\_sp := sp$
    act5 : $sp := 1$
    act6 : $PM\_Sensor\_A := OFF$
  END
```

The event ($Sensor\_OFF\_V$) is used to set the pacemaker's sensor (PM_Sensor_V) of ventricular in OFF state. The guards ($grd1, grd2$) of this event represent that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON and the current clock counter sp is greater than or equal to VRP and PVARP. The next guard ($grd3$) represents that the current clock counter sp is less than ventriculoatrial (VA) interval or greater than or equal to ventriculoatrial (VA) interval and less than pace interval ($Pace\_Int$). The last two guards ($grd4, grd5$) state that the pacemaker's actuator ($PM\_Actuator\_V, PM\_Actuator\_A$) of ventricular and atrial are OFF. The actions ($act1 - act6$) of this event state that the pacemaker's sensor (PM_Sensor_V) of ventricular sets in OFF state, assigns the value of variable (AV_count) as 0, atrioventricular (AV) counter state (AV_Count_STATE) sets in FALSE state, assigns the value of clock counter sp to new variable ($last\_sp$), assigns the value of the current clock counter sp as 1 and sets OFF state of the pacemaker's actuator ($PM\_Actuator\_A$) of atrial.

```
EVENT Sensor_ON_A
  WHEN
    grd1 : $PM\_Sensor\_A = OFF$
    grd2 : $PM\_Sensor\_V = OFF$
    grd3 : $sp < Pace\_Int - FixedAV \land$
           $sp \geq VRP \land sp \geq PVARP$
  THEN
    act1 : $PM\_Sensor\_A := ON$
  END
```

The events ($Sensor\_ON\_A$ and $Sensor\_OFF\_A$) is used to control the sensing activities from the atrial chamber. The pacemaker's sensor (PM_Sensor_A) of atrial chamber synchronizes ON and OFF states under real time constraints. The guard ($grd1$) of event ($Sensor\_ON\_A$) represents that if the pacemaker's sensor (PM_Sensor_A) of atrial is OFF and second guard ($grd2$) represents that the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF.In last guard ($grd3$) represents that the current clock counter sp is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP interval. If all guards are true then in action part of this event, pacemaker's sensor (PM_Sensor_A) of atrial sets in ON state.

```
EVENT Sensor_OFF_A
  WHEN
    grd1 : $PM\_Sensor\_A = ON$
    grd2 : $sp < Pace\_Int - FixedAV \land$
           $sp \geq VRP \land sp \geq PVARP$
  THEN
    act1 : $PM\_Sensor\_A := OFF$
    act2 : $AV\_Count\_STATE := TRUE$
  END
```

The event ($Sensor\_OFF\_A$) is used to set the pacemaker's sensor (PM_Sensor_A) of atrial in OFF state. The guards of this event represent that the pacemaker's sensor (PM_Sensor_A) of atrial is ON and the current clock counter sp is less than ventriculoatrial (VA) interval and greater than or equal to VRP and PVARP. In actions of this event state that the pacemaker's sensor (PM_Sensor_A) of atrial sets in OFF state and atrioventricular (AV) counter state (AV_Count_STATE) sets TRUE.

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

The event (tic) of this abstraction progressively increases the current clock counter sp under pre-defined pace interval ($Pace\_Int$). The guards of this event control the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_A, PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus of atrial and ventricular chamber and synchronizes ON and OFF states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) in atrial and ventricular under real time constraints.

```
EVENT tic_AV
  WHEN
    grd1 : $AV\_Count < FixedAV$
    grd2 : $AV\_Count\_STATE = TRUE$
    grd3 : $(sp \geq VRP \land sp \geq PVARP \land$
           $sp < Pace\_Int - FixedAV)$
           $\lor$
           $(sp \geq Pace\_Int - FixedAV \land$
           $sp < Pace\_Int)$
  THEN
    act1 : $AV\_Count := AV\_Count + 1$
    act2 : $sp := sp + 1$
  END
```

The last event (tic_AV) of this abstraction progressively counts the atrioventricular (AV) interval and also increases the current clock counter (sp) is represented in actions ($act1$ and $act2$) . The guards of this event states that atrioventricular (AV) counter (AV_Count) is less than and equal to atrioventricular (AV) interval ($FixedAV$), atrioventricular (AV) state (AV_count_STATE) is TRUE and the current clock counter sp is within the ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$) and greater than or equal to VRP and PVARP or the current clock counter sp is greater than or equal to atrioventricular (AV) interval and less than pace interval ($Pace\_Int$).

4.2 First refinement:Threshold

The pacemaker control unit delivers stimulation to the heart chambers, on the basis of measured threshold value under safety margin. We define two new constants STA_THR_A and STA_THR_V to hold the standard threshold value in axioms ($axm1$ and $axm2$). The threashold constants are different for the atria and the ventricular chambers.

$$\boxed{\begin{array}{l} axm1 : STA\_THR\_A \in nat_1 \land STA\_THR\_A = 75 \\ axm1 : STA\_THR\_V \in nat_1 \land STA\_THR\_V = 250 \end{array}}$$

The pacemaker's sensor starts sensing after the refractory period; Atria Refractory Period (ARP) , Ventricular Refractory Period (VRP). The pacemaker's actuator delivers a pacing stimulus when sensing value is greater than or equal to the standard threshold constants STA_THR_A or STA_THR_V. In DOO operating mode only pacemaker's actuators paces in atrial and ventricular chambers under automatic pace interval without using any pacemaker's sensors, so in this mode no any refinement related to the threashold.

4.2.1 First refinement of DVI mode:

In the first refinement of DVI operating mode, we formalize the concept of sensing threshold value in the double 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. We introduce the new variable (Thr_V) to hold the sensing threashold value of the pacemaker's sensor of ventricular chamber and next variable (Thr_V_State) represents the TRUE and FALSE states of the pacemaker's sensor to sense the intrinsic activity of the ventricular chamber.

$inv1: Thr\_V \in \mathbb{N}_1$
$inv2: Thr\_V\_State \in BOOL$
$inv3: sp > VRP \wedge sp < Pace\_Int - FixedAV \Rightarrow PM\_Sensor\_V = ON$
$inv4: PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int$
$inv5: sp > VRP \wedge sp < Pace\_Int \wedge Thr\_V \geq STA\_THR\_V \wedge$
$\quad Thr\_V\_State = TRUE \Rightarrow PM\_Sensor\_V = OFF$
$inv6: sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Actuator\_V = OFF$
$inv7: sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Actuator\_A = OFF$
$inv8: sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_V = ON$
$inv9: PM\_Actuator\_A = ON \Rightarrow sp \geq Pace\_IntFixedAVspVRPsp < Pace\_Int$

From invariants ($inv3 - inv9$) represent the safety properties of the pacemaker system under pacing and sensing activities of electrode in DVI operating mode. The third invariant ($inv3$) states that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON when current clock counter (sp) is greater than VRP and less than ventriculoatrial(VA) interval. The fourth invariant state that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON when current clock counter sp is equal to the pace interval $Pace\_Int$. The invariant ($inv5$) represents that the pacemaker's sensor (PM_Sensor_V) of ventricular is OFF when clock counter sp is greater then VRP, less then pace interval ($Pace\_Int$), sensed value (Thr_V) is greater than or equal to standard threshold (STA_THR_V) value of ventricular chamber. The next three invariants ($inv6$, $inv7$ and $inv8$) represent that the pacemaker's actuator ($PM\_Actuator\_A, PM\_Actuator\_V$) of atrial and ventricular are OFF and the pacemaker's sensor (PM_Sensor_V) of ventricular is ON when clock counter sp is greater than ventriculoatrial (VA) interval, less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter state (AV_Count_STATE) is $TRUE$. The last invariant ($inv9$) states that the pacemaker's actuator of atrial chambers can be ON when current clock counter sp is within the ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$) and greater than or equal to VRP and less than pace interval $Pace\_Int$.

In this refinement we introduce the new event (Thr_Value_V) for sensing the intrinsic activities of ventricular chamber. This event is synchronized with all other events of this operating mode under all safety properties and real time constraints. The guards ($grd2 - grd4$) of this event state that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON, threashold state (Thr_V_State) of ventricular is TRUE and sensed value (Thr_V) is less than standard threshold value (STA_THR_V) of ventricular chamber. The last guard states that either the current clock counter sp is greater than or equal to VRP and less than ventriculoatrial (VA) interval or the current clock counter sp is greater then and equal to atrioventricular (AV) interval and less then pace interval ($Pace\_Int$). The actions ($act1 - act2$) of this event state that actual sensed value (Thr_V_val) of ventricular chamber assigns to variable (Thr_V) and sets the FALSE state of threashold ventricular state (Thr_V_State).

```
EVENT Thr_Value_V
  WHEN
    grd1 : $Thr\_V\_val \in \mathbb{N}$
    grd2 : $PM\_Sensor\_V = ON$
    grd3 : $Thr\_V\_State = TRUE$
    grd4 : $Thr\_V < STA\_THR\_V$
    grd5 : $(sp \geq VRP \land sp < Pace\_Int - FixedAV)$
           $\lor$
           $(sp \geq Pace\_Int - FixedAV \land sp < Pace\_Int)$
  THEN
    act1 : $Thr\_V := Thr\_V\_val$
    act2 : $Thr\_V\_State := FALSE$
  END
```

We add some new actions in events ($Actuator\_OFF\_V$, $Sensor\_ON\_V$, and $Sensor\_OFF\_V$)[1] to synchronize the sensing activities using event (Thr_V_val) under real time constraints which are already defined in the abstract model of this operating mode.

```
EVENT Actuator_OFF_V
$\oplus$   act9 : $Thr\_V := 0$
$\oplus$   act10 : $Thr\_V\_State := FALSE$

EVENT Sensor_ON_V
$\oplus$   act2 : $Thr\_V\_State := FALSE$

EVENT Sensor_OFF_V
$\oplus$   act8 : $Thr\_V := 0$
```

The event (*tic*) of this refinement model progressively increases the current clock counter *sp* under pre-defined pace interval ($Pace\_Int$). The guard ($grd1$) of this event controls the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of pacemaker's actuators ($PM\_Actuator\_A, PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus of ventricular chamber and synchronizes ON and OFF states of the pacemaker's sensor (PM_Sensor_V) in ventricular under real time constraints. We modify the guard ($grd1$) of this event and add more properties to synchronize the pacing and sensing activities and we also add new action ($act2$). The additional guards and action handle the behavior of event (Thr_V_val) to sense the intrinsic activities from the ventricular chamber at actual required time.

[1]$\oplus$: To add a new guard and an action.

```
EVENT tic
 WHEN
   grd1 : $(sp < VRP \wedge PM\_Sensor\_V = OFF \wedge$
          $AV\_Count\_STATE = FALSE)$
          $\vee$
          $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV \wedge$
          $PM\_Sensor\_V = ON \wedge AV\_Count\_STATE = FALSE \wedge$
          $Thr\_V\_State = FALSE \wedge Thr\_V < STA\_THR\_V))$
 THEN
$\oplus$  act2 : $Thr\_V\_State := TRUE$
 END
```

We add some new guards ($grd4-grd8$) and an action ($act3$) in event (tic_AV) of this refinement. The new guards provide more specific and stronger guards to count the atrioventricular (AV) interval and action ($act3$) states that threashold state (Thr_V_State) of ventricular is TRUE when all guards are satisfied.

```
EVENT tic_AV
 WHEN
$\oplus$  grd4 $PM\_Sensor\_V = ON$
$\oplus$  grd5 $Thr\_V\_State = FALSE$
$\oplus$  grd6 $Thr\_V < STA\_THR\_V$
$\oplus$  grd7 $PM\_Actuator\_V = OFF$
$\oplus$  grd8 $PM\_Actuator\_A = OFF$
 THEN
$\oplus$  act3 : $Thr\_V\_State := TRUE$
 END
```

4.2.2 First refinement of DDI mode:

In the first refinement of DDI operating mode, we formalize the concept of sensing threshold value of the double 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. We introduce the new variables (Thr_A and Thr_V) to hold the sensing threashold value of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) of atrial and ventricular chambers. Similarly next variables (Thr_A_State and Thr_V_State) represent TRUE or FALSE states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) to sense the intrinsic activity of the atrial and ventricular chambers.

$inv1 : Thr\_A \in \mathbb{N}_1$
$inv2 : Thr\_V \in \mathbb{N}_1$
$inv3 : Thr\_A\_State \in BOOL$
$inv4 : Thr\_V\_State \in BOOL$
$inv5 : PM\_Actuator\_V = ON \Rightarrow sp = Pace\_Int$
$inv6 : sp > VRP \wedge sp < Pace\_Int - FixedAV \Rightarrow PM\_Actuator\_A = OFF$
$inv7 : sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_V = ON$
$inv8 : sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Actuator\_V = OFF$
$inv9 : sp > Pace\_Int - FixedAV \wedge sp < Pace\_Int \wedge$
$\quad AV\_Count\_STATE = TRUE \Rightarrow PM\_Actuator\_A = OFF$
$inv10 : PM\_Actuator\_A = ON \Rightarrow sp = Pace\_Int - FixedAV$

From invariants ($inv5 - inv10$) represent the safety properties of the pacemaker system under pacing and sensing activities of electrode in DDI operating mode. The fifth invariant ($inv5$) state that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON when current clock counter sp is equal to the pace interval $Pace\_Int$. The next invariant ($inv6$) states that the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is ON when current clock counter sp is greater than VRP and less than ventriculoatrial(VA) interval. The next three invariants ($inv7$, $inv8$, $inv9$) represent that the pacemaker's actuator ($PM\_Actuator\_A, PM\_Actuator\_V$) of atrial and ventricular chambers are OFF and the pacemaker's sensor (PM_Sensor_V) of ventricular is ON when current clock counter (sp) is greater than ventriculoatrial (VA) interval, less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE. The last invariant states that the pacemaker's actuator of atrial chamber is ON when current clock counter sp is equal to ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$).

```
EVENT Thr_Value_V
 WHEN
   grd1 : $Thr\_V\_val \in \mathbb{N}$
   grd2 : $PM\_Sensor\_V = ON$
   grd3 : $Thr\_V\_State = TRUE$
   grd4 : $Thr\_V < STA\_THR\_V$
   grd5 : $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV)$
          $\vee$
          $(sp \geq Pace\_Int - FixedAV \wedge sp < Pace\_Int)$
   grd6 $(Thr\_A\_State = FALSE \wedge Thr\_A < STA\_THR\_A)$
          $\vee$
          $PM\_Sensor\_A = OFF \wedge AV\_Count < FixedAV$
 THEN
   act1 : $Thr\_V := Thr\_V\_val$
   act2 : $Thr\_V\_State := FALSE$
 END
```

In this refinement we introduce the two new events (Thr_Value_V and Thr_Value_A) for sensing the intrinsic activities from ventricular and atrial chambers. These events are synchronized with all other events of this operating mode under all safety properties and real time constraints. The guards ($grd2 - grd4$) of event (Thr_V_val) state that pacemaker's sensor (PM_Sensor_V) of ventricular is ON, threashold state (Thr_V_State) of ventricular is TRUE and sensed value (Thr_V) is less than standard threshold value (STA_THR_V) of ventricular chamber. The next guard ($grd5$) represents that either clock counter (sp) is greater than or equal to VRP and less than ventriculoatrial (VA) interval or clock counter (sp) is greater than or equal to atrioventricular (AV) interval and less then pace interval ($Pace\_Int$). The last guard ($grd6$) states that either threshold state (Thr_A_State) of atrial chamber is FLASE and threashold value (Thr_A) of atrial is less then standard threashold value (STA_THR_A) of atrial chamber or the pacemaker's sensor (PM_Sensor_A) of atrial chamber is OFF and atrioventricular (AV) counter (AV_Count) is less than atrioventricular (AV) interval ($FixedAV$). The actions ($act1 - act2$) of this event state that actual sensed value (Thr_V_val) of ventricular chamber assigns to variable Thr_V and sets FALSE state of threashold ventricular state (Thr_V_State).

EVENT Thr_Value_A
WHEN
grd1 : $Thr\_A\_val \in \mathbb{N}$
grd2 : $PM\_Sensor\_A = ON$
grd3 : $Thr\_A\_State = TRUE$
grd4 : $Thr\_A < STA\_THR\_A$
grd5 : $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV)$
THEN
act1 : $Thr\_A := Thr\_A\_val$
act2 : $Thr\_A\_State := FALSE$
END

Other new event (Thr_Value_A) introduce to take the intrinsic activities of the atrial chamber. The guards ($grd2 - grd4$) state that the pacemaker's sensor (PM_Sensor_A) of the atrial chamber is ON, threashold state of atrial chamber is TRUE and sensed value (Thr_A) of atrial chamber is less than standard threashold (STA_THR_A) of atrial chamber. The last guard ($grd5$) of this event state that the current clock counter sp is greater than or equal to VRP and less than ventriculoatrial (VA) interval. The actions ($act1 - act2$) of this event state that actual sensed value (Thr_A_val) of atrial chamber assigns to variable (Thr_A) and sets FALSE state of threashold ventricular state (Thr_A_State).

EVENT Actuator_OFF_V
$\oplus$ act6 : $Thr\_A := 0$
$\oplus$ act7 : $Thr\_V := 0$
$\oplus$ act8 : $Thr\_A\_State := FALSE$
$\oplus$ act9 : $Thr\_V\_State := FALSE$

EVENT Sensor_ON_A
$\oplus$ act2 : $Thr\_A\_State := TRUE$

EVENT Sensor_OFF_V
$\oplus$ grd6 : $Thr\_V \geq STA\_THR\_V$
$\oplus$ act7 : $Thr\_A := 0$
$\oplus$ act8 : $Thr\_V := 0$
$\oplus$ act9 : $Thr\_A\_State := FALSE$
$\oplus$ act10 : $Thr\_V\_State := FALSE$

We add some new actions and guards in events ($Actuator\_OFF\_V$, $Sensor\_ON\_A$, and $Sensor\_OFF\_V$) to synchronize the sensing activities using events (Thr_A_val and Thr_V_val) under real time constraints, which are already defined in the abstract model of this operating mode.

```
EVENT tic
  WHEN
    grd1 : $(sp < VRP \wedge AV\_Count\_STATE = FALSE)$
           $\vee$
           $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV \wedge$
           $PM\_Sensor\_V = ON \wedge PM\_Actuator\_A = OFF \wedge$
           $Thr\_V\_State = FALSE \wedge Thr\_V < STA\_THR\_V))$
  THEN
⊕    act2 : $Thr\_A\_State := TRUE$
⊕    act3 : $Thr\_V\_State := TRUE$
  END
```

The event (*tic*) of this refinement model progressively increases the current clock counter *sp* under pre-defined pace interval ($Pace\_Int$). The guard of this event controls the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of pacemaker's actuator ($PM\_Actuator\_A$, $PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus of ventricular chamber and synchronizes ON and OFF states of pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) in atrial and ventricular chambers under real time constraints. We modify the guard ($grd1$) of this event and add more properties to synchronize the pacing and sensing activities and we also add new actions ($act2$ and $act3$). The additional guards and action handle the behavior of events (Thr_A_val and Thr_V_val) to sense the intrinsic activities from the atrial and ventricular chambers.

```
EVENT tic_AV
  WHEN
⊕    grd4 $PM\_Sensor\_V = ON$
⊕    grd5 $Thr\_V\_State = FALSE$
⊕    grd6 $Thr\_V < STA\_THR\_V$
⊕    grd7 $PM\_Actuator\_V = OFF$
⊕    grd8 $PM\_Actuator\_A = OFF$
  THEN
⊕    act3 : $Thr\_V\_State := TRUE$
  END
```

We add some new guards ($grd4-grd8$) and an action ($act3$) in event (tic_AV) of this refinement. The new guards provide more specific and stronger guards to count the atrioventricular (AV) interval and action ($act3$) states that threashold state (Thr_V_State) of ventricular sets TRUE.

4.2.3 First refinement of VDD mode:

In the first refinement of VDD operating mode, we formalize the concept of sensing threshold value of the double 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. We introduce the new variables (Thr_A and Thr_V) to hold the sensing threashold value of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) of atrial and ventricular chambers. Similarly next variables (Thr_A_State and Thr_V_State) represent TRUE or FALSE state of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) to sense the intrinsic activity of the atrial and ventricular chambers.

$$
\begin{array}{l}
inv1 : Thr\_A \in \mathbb{N}_1 \\
inv2 : Thr\_V \in \mathbb{N}_1 \\
inv3 : Thr\_A\_State \in BOOL \\
inv4 : Thr\_V\_State \in BOOL \\
inv5 : sp > VRP \land sp < Pace\_Int \land AV\_Count\_STATE = FALSE \Rightarrow \\
\qquad PM\_Sensor\_A = ON \\
inv6 : PM\_Actuator\_V = ON \Rightarrow (sp = Pace\_Int) \lor \\
\qquad sp < Pace\_Int \land AV\_Count > V\_Blank \land AV\_Count \geq FixedAV \\
inv7 : sp > Pace\_Int - FixedAV \land sp < Pace\_Int \land \\
\qquad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_A = OFF \\
inv8 : sp > Pace\_Int - FixedAV \land sp < Pace\_Int \land \\
\qquad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_V = ON
\end{array}
$$

From invariants ($inv5 - inv8$) represent the safety properties of the pacemaker system under pacing and sensing activities of electrode in VDD operating mode. The fifth invariant ($inv5$) states that the pacemaker's sensor (PM_Sensor_A) of atrial is ON when current clock counter *sp* is greater than VRP and less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter state (AV_Count_STATE) is FALSE. The next invariant ($inv6$) represents that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON when either the current clock counter *sp* is equal to the pace interval ($Pace\_Int$) or clock counter *sp* is less then pace interval ($Pace\_Int$), atrioventricular counter (AV_Count) is greater than blanking period (V_Blank) and greater than or equal to the atrioventricular (AV) interval ($FixedAV$). The last two invariants ($inv7$, $inv8$) represent that the pacemaker's sensor (PM_Sensor_A) of atrial is OFF and the pacemaker's sensor (PM_Sensor_V) of ventricular is ON when the current clock counter *sp* is greater than ventriculoatrial (VA) interval, less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter state (AV_Count_STATE) is TRUE.

```
EVENT Thr_Value_V
 WHEN
   grd1 : $Thr\_V\_val \in \mathbb{N}$
   grd2 : $PM\_Sensor\_V = ON$
   grd3 : $Thr\_V\_State = TRUE$
   grd4 : $Thr\_V < STA\_THR\_V$
   grd5 : $(sp \geq VRP \land sp < Pace\_Int - FixedAV)$
          $\lor$
          $(sp \geq Pace\_Int - FixedAV \land sp < Pace\_Int)$
   grd6 $(Thr\_A\_State = FALSE \land Thr\_A < STA\_THR\_A)$
          $\lor$
          $(PM\_Sensor\_A = OFF \land AV\_Count < FixedAV)$
 THEN
   act1 : $Thr\_V := Thr\_V\_val$
   act2 : $Thr\_V\_State := FALSE$
 END
```

In this refinement we introduce the two new events (Thr_Value_V and Thr_Value_A) for sensing the intrinsic activities from ventricular and atrial chambers. These events are synchronized with all other events of this operating mode under all safety properties and real time constraints. The guards ($grd2 - grd4$) of event (Thr_Value_V) state that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON, threashold state (Thr_V_State) of ventricular is TRUE and sensed value (Thr_V) is less than standard threshold value (STA_THR_V) of ventricular chamber. The next guard ($grd5$) represents that either clock counter (sp) is greater then and equal to VRP and less than ventriculoatrial (VA) interval or clock counter (sp) is greater than or equal to atrioventricular (AV) interval and less then pace interval ($Pace\_Int$). The last guard ($grd6$) states that either threshold state (Thr_A_State) of atrial chamber is FLASE and threashold value (Thr_A) of atrial is less then standard threashold value (STA_THR_A) of atrial chamber or pacemaker's sensor PM_Sensor_A of atrial is OFF and atrioventricular (AV) counter (AV_Count) is less than atrioventricular (AV) interval ($FixedAV$). The actions ($act1 - act2$) of this event state that actual sensed value (Thr_V_val) of ventricular chamber assigns to variable (Thr_V) and sets FALSE state of threashold ventricular state (Thr_V_State).

EVENT Thr_Value_A
WHEN
grd1 : $Thr\_A\_val \in \mathbb{N}$
grd2 : $PM\_Sensor\_A = ON$
grd3 : $Thr\_A\_State = TRUE$
grd4 : $Thr\_A < STA\_THR\_A$
grd5 : $(sp \geq VRP \wedge sp < Pace\_Int)$
THEN
act1 : $Thr\_A := Thr\_A\_val$
act2 : $Thr\_A\_State := FALSE$
END

Other new event (Thr_Value_A) introduce to take the intrinsic activities of atrial chamber. The guards ($grd2 - grd4$) state that the pacemaker's sensor (PM_Sensor_A) of atrial chamber is ON, threashold state (Thr_A_State) of atrial chamber is TRUE and sensed value (Thr_A) of atrial chamber is less than standard threashold (STA_THR_A) of atrial chamber. The last guard of this event state that the current clock counter *sp* is greater than or equal to VRP and less than pace interval ($Pace\_Int$) . The actions ($act1 - act2$) of this event state that actual sensed value (Thr_A_val) of atrial chamber assigns to variable (Thr_A) and sets FALSE state of threashold atrial state (Thr_A_State).

EVENT Actuator_OFF_V
⊕ act6 : $Thr\_A := 0$
⊕ act7 : $Thr\_V := 0$
⊕ act8 : $Thr\_A\_State := FALSE$
⊕ act9 : $Thr\_V\_State := FALSE$

EVENT Sensor_ON_A
⊕ act2 : $Thr\_A\_State := TRUE$

EVENT Sensor_OFF_A
⊕ grd3 : $Thr\_A \geq STA\_THR\_A$
⊕ act2 : $Thr\_A\_State := TRUE$

EVENT Sensor_OFF_V
⊕ grd5 : $Thr\_V \geq STA\_THR\_V$
⊕ act7 : $Thr\_A := 0$
⊕ act8 : $Thr\_V := 0$
⊕ act9 : $Thr\_A\_State := FALSE$
⊕ act10 : $Thr\_V\_State := FALSE$

We add some new actions and guards in events ($Actuator\_OFF\_V$, $Sensor\_ON\_A$, $Sensor\_OFF\_A$, and $Sensor\_OFF\_V$), to synchronize the sensing activities using events (Thr_A_val and Thr_V_val) under real time constraints, which are already defined in the abstract model of this operating mode.

EVENT tic
  WHEN
    grd1 : $(sp < VRP$
          $\vee$
          $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV \wedge$
          $PM\_Sensor\_V = ON \wedge PM\_Sensor\_A = ON \wedge$
          $Thr\_V\_State = FALSE \wedge Thr\_V < STA\_THR\_V))$
    grd2 : $AV\_Count\_STATE = FALSE$
  THEN
$\oplus$ act2 : $Thr\_A\_State := TRUE$
$\oplus$ act3 : $Thr\_V\_State := TRUE$
  END

The event (tic) of this refinement model progressively increases the current clock counter sp under pre-defined pace interval ($Pace\_Int$). The guard of this event controls the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular chamber and also control the sensing intrinsic stimulus of atrail and ventricular chambers and synchronizes ON and OFF states of the pacemaker's sensors (PM_Sensor_A, PM_Sensor_V) in heart chambers under real time constraints. We modify the guard ($grd1$) and new guard ($grd2$) of this event and add more properties to synchronize the pacing and sensing activities and we also add new actions ($act2$ and $act3$). The additional guards and action handle the behavior of events (Thr_A_val and Thr_V_val) to sense the intrinsic activities from the atrial and ventricular chambers.

EVENT tic_AV
  WHEN
$\oplus$ grd4 $PM\_Sensor\_V = ON$
$\oplus$ grd5 $Thr\_V\_State = FALSE$
$\oplus$ grd6 $Thr\_V < STA\_THR\_V$
$\oplus$ grd7 $PM\_Actuator\_V = OFF$
$\oplus$ grd8 $PM\_Sensor\_A = OFF$
  THEN
$\oplus$ act3 : $Thr\_V\_State := TRUE$
  END

We add some new guards ($grd4 - grd8$) and an action ($act3$) of this event (tic_AV) in refinement. The new guards provide more specific and stronger guards to count the atrioventricular (AV) interval and action ($act3$) states that threashold state (Thr_V_State) of ventricular sets TRUE.

4.2.4 First refinement of DDD mode:

In the first refinement of DDD operating mode, we formalize the concept of sensing threshold value of the double 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. We introduce the new variables (Thr_A and Thr_V) to hold the sensing threashold value of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) of atrial and ventricular chambers. Similarly next variables (Thr_A_State and Thr_V_State) represent TRUE and FALSE states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) to sense the intrinsic activity of the atrial and ventricular chambers.

$$
\begin{array}{l}
inv1 : Thr\_A \in \mathbb{N}_1 \\
inv2 : Thr\_V \in \mathbb{N}_1 \\
inv3 : Thr\_A\_State \in BOOL \\
inv4 : Thr\_V\_State \in BOOL \\
inv5 : sp > VRP \land sp < Pace\_Int - FixedAV \Rightarrow PM\_Sensor\_V = ON \\
inv6 : PM\_Actuator\_V = ON \Rightarrow (sp = Pace\_Int) \lor \\
\qquad\qquad sp < Pace\_Int \land AV\_Count > V\_Blank \land AV\_Count \geq FixedAV \\
inv7 : sp > Pace\_Int - FixedAV \land sp < Pace\_Int \land \\
\qquad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_A = OFF \\
inv8 : sp > Pace\_Int - FixedAV \land sp < Pace\_Int \land \\
\qquad AV\_Count\_STATE = TRUE \Rightarrow PM\_Sensor\_V = ON \\
inv9 : sp > Pace\_Int - FixedAV \land sp < Pace\_Int \land \\
\qquad AV\_Count\_STATE = TRUE \Rightarrow PM\_Actuator\_A = OFF \\
inv10 : PM\_Actuator\_A = ON \Rightarrow sp \geq Pace\_Int - FixedAV
\end{array}
$$

From invariants ($inv5 - inv10$) represent the safety properties of the pacemaker system under pacing and sensing activities of electrode in DDD operating mode. The fifth invariant ($inv5$) states that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON when current clock counter sp is greater than VRP and less than ventriculoatrial (VA) interval. The next invariant ($inv6$) represents that the pacemaker's actuator ($PM\_Actuator\_V$) of ventricular is ON when either the current clock counter sp is equal to the pace interval ($Pace\_Int$) or clock counter sp is less then pace interval ($Pace\_Int$), atrioventricular counter (AV_Count) is greater than blanking period (V_Blank) and greater than or equal to the atrioventricular (AV) interval ($FixedAV$). The next three invariants ($inv7$, $inv8$, $inv9$) represent that the pacemaker's sensor (PM_Sensor_A) of atrial is OFF, the pacemaker's sensor (PM_Sensor_V) of ventricular is ON and the pacemaker's actuator ($PM\_Actuator\_A$) of atrial is OFF, when current clock counter sp is greater than ventriculoatrial (VA) interval, less than pace interval ($Pace\_Int$) and atrioventricular (AV) counter state

(AV_Count_STATE) is TRUE. The last invariant states that the pacemaker's actuator of atrial chamber is ON when current clock counter *sp* is greater than or equal to the ventriculoatrial (VA) interval ($Pace\_Int - FixedAV$).

```
EVENT Thr_Value_V
  WHEN
    grd1 : $Thr\_V\_val \in \mathbb{N}$
    grd2 : $PM\_Sensor\_V = ON$
    grd3 : $Thr\_V\_State = TRUE$
    grd4 : $Thr\_V < STA\_THR\_V$
    grd5 : $(sp \geq VRP \land sp < Pace\_Int - FixedAV)$
           $\lor$
           $(sp \geq Pace\_Int - FixedAV \land sp < Pace\_Int)$
    grd6 $(Thr\_A\_State = FALSE \land Thr\_A < STA\_THR\_A)$
           $\lor$
           $(PM\_Sensor\_A = OFF \land AV\_Count < FixedAV)$
  THEN
    act1 : $Thr\_V := Thr\_V\_val$
    act2 : $Thr\_V\_State := FALSE$
  END
```

In this refinement, we introduce the two new events (Thr_Value_V and Thr_Value_A) for sensing the intrinsic activities from ventricular and atrial chambers. These events are synchronized with all other events of this operating mode under all safety properties and real time constraints. The guards ($grd2 - grd4$) of event (Thr_Value_V) state that the pacemaker's sensor (PM_Sensor_V) of ventricular is ON, threashold state (Thr_V_State) of ventricular is TRUE and sensed value (Thr_V) is less than standard threshold value (STA_THR_V) of the ventricular chamber. The next guard ($grd5$) represents that either clock counter *sp* is greater then and equal to VRP and less than ventriculoatrial (VA) interval or clock counter *sp* is greater than or equal to atrioventricular (AV) interval and less then pace interval ($Pace\_Int$). The last guard ($grd6$) states that either threshold state (Thr_A_State) of atrial is FLASE and threashold value (Thr_A) of atrial is less then standard threashold value (STA_THR_A) of atrial chamber or the pacemaker's sensor (PM_Sensor_A) of atrial chamber is OFF and atrioventricular (AV) counter is less than atrioventricular (AV) interval ($FixedAV$). The actions ($act1 - act2$) of this event state that actual sensed value (Thr_V_val) of ventricular chamber assigns to variable (Thr_V) and sets the FALSE state of threashold ventricular state (Thr_V_State).

EVENT Thr_Value_A
WHEN
grd1 : $Thr\_A\_val \in \mathbb{N}$
grd2 : $PM\_Sensor\_A = ON$
grd3 : $Thr\_A\_State = TRUE$
grd4 : $Thr\_A < STA\_THR\_A$
grd5 : $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV)$
THEN
act1 : $Thr\_A := Thr\_A\_val$
act2 : $Thr\_A\_State := FALSE$
END

Other new event (Thr_Value_A) introduce to take the intrinsic activities of atrial chamber. The guards ($grd2 - grd4$) state that pacemaker's sensor (PM_Sensor_A) of atrial chamber is ON, threashold state (Thr_A_State) of atrial chamber is TRUE and sensed value (Thr_A) of atrial chamber is less than standard threashold (STA_THR_A) of atrial chamber. The last guard of this event state that clock counter sp is greater than or equal to VRP and less than ventriculoatrial (VA) interval. The actions ($act1 - act2$) of this event state that actual sensed value (Thr_A_val) of atrial chamber assigns to variable (Thr_A) and sets FALSE state of threashold atrial state (Thr_A_State).

EVENT Actuator_OFF_V
$\oplus$ act6 : $Thr\_A := 0$
$\oplus$ act7 : $Thr\_V := 0$
$\oplus$ act8 : $Thr\_A\_State := FALSE$
$\oplus$ act9 : $Thr\_V\_State := FALSE$

EVENT Sensor_ON_A
$\oplus$ act2 : $Thr\_A\_State := TRUE$

EVENT Sensor_OFF_A
$\oplus$ grd3 : $Thr\_A \geq STA\_THR\_A$
EVENT Sensor_OFF_V
$\oplus$ grd6 : $Thr\_V \geq STA\_THR\_V$
$\oplus$ act7 : $Thr\_A := 0$
$\oplus$ act8 : $Thr\_V := 0$
$\oplus$ act9 : $Thr\_A\_State := FALSE$
$\oplus$ act10 : $Thr\_V\_State := FALSE$

We add some new actions and guards in events ($Actuator\_OFF\_V$, $Sensor\_ON\_A$, $Sensor\_OFF\_A$, and $Sensor\_OFF\_V$) to synchronize the sensing activities using events (Thr_A_val and Thr_V_val) under real time constraints, which are already defined in the abstract model of this operating mode.

```
EVENT tic
 WHEN
   grd1 : $(sp < VRP \wedge AV\_Count\_STATE = FALSE$
          $\vee$
          $(sp \geq VRP \wedge sp < Pace\_Int - FixedAV \wedge$
          $PM\_Sensor\_V = ON \wedge PM\_Sensor\_A = ON \wedge$
          $Thr\_V\_State = FALSE \wedge Thr\_V < STA\_THR\_V))$
   grd2 : $AV\_Count\_STATE = FALSE$
 THEN
⊕    act2 : $Thr\_A\_State := TRUE$
⊕    act3 : $Thr\_V\_State := TRUE$
 END
```

The event (tic) of this refinement model progressively increases the current clock counter sp under pre-defined pace interval ($Pace\_Int$). The guard ($grd1$) of this event control the pacing stimulus into the heart chambers (atria and ventricular), synchronizes ON and OFF states of the pacemaker's actuator ($PM\_Actuator\_A, PM\_Actuator\_V$) of each chamber (atria and ventricular) and also control the sensing intrinsic stimulus of atrial and ventricular chambers and synchronizes ON and OFF states of the pacemaker's sensor (PM_Sensor_A, PM_Sensor_V) of atrial and ventricular under real time constraints. We modify the guard ($grd1$) and new guard ($grd2$) of this event and add more properties to synchronize the pacing and sensing activities and we also add new actions ($act2$ and $act3$). The additional guards and action handle the behavior of events (Thr_A_val and Thr_V_val) to sense the intrinsic activities from the atrial and ventricular chambers.

```
EVENT tic_AV
 WHEN
⊕    grd4 $PM\_Sensor\_V = ON$
⊕    grd5 $Thr\_V\_State = FALSE$
⊕    grd6 $Thr\_V < STA\_THR\_V$
⊕    grd7 $PM\_Actuator\_V = OFF$
⊕    grd8 $PM\_Sensor\_A = OFF$
⊕    grd9 $PM\_Actuator\_A = OFF$
 THEN
⊕    act3 : $Thr\_V\_State := TRUE$
 END
```

We add some new guards ($grd4-grd9$) and an action ($act3$) of event (tic_AV) in this refinement. The new guards provide more specific and stronger guards to count the atrioventricular (AV) interval and action ($act3$) states that threashold state (Thr_V_State) of ventricular sets TRUE.

4.3 Second refinement:Rate Modulation

Rate modulation term is used to describe the capacity of a pacing system to respond to physiologic need by increasing and decreasing pacing rate. The rate modulation mode of the 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 the pacemaker on the basis of maximum exertion is performed by the patient. 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.

In this final refinement, we introduce the rate modulation function and found some new operating modes (AOOR,VOOR,AAIR,VVIR,AATR and VVTR) of the pacemaker system. For modeling the rate modulation, we introduce the new constants maximum sensor rate MSR as $MSR \in 50 \,..\, 175$ and acc_thr as $acc\_thr \in \mathbb{N}$. The maximum sensor rate (MSR) is the maximum pacing rate allowed as a result of sensor control and it must be between 50 and 175 pulse per minute (ppm). The constant acc_thr represents the activity threshold. A new variable $acler\_sensed$ is defined as $acler\_sensed \in \mathbb{N}$, to store the measured value from the accelerometer. The accelerometer is used to measure the physical activities of the body in a pacemaker system.

The two invariants ($inv1, inv2$) provide the safety margin and state that the heart rate never falls below the lower rate limit (LRL) and never exceed the maximum sensor rate (MSR) limit.

$$\boxed{\begin{array}{ll} inv1: & acler\_sensed < acc\_thr \\ & \Rightarrow \\ & Pace\_Int = 60000/LRL \\ \\ inv2: & acler\_sensed > acc\_thr \\ & \Rightarrow \\ & Pace\_Int = 60000/MSR \end{array}}$$

In this final refinement, we introduce only two new events $Increase\_Interval$ and $Decrease\_Interval$, to control the pacing rate of the double electrode pacemaker in the rate modulating operating modes. The new events $Increase\_Interval$ and $Decrease\_Interval$ control the value of pace interval variable $Pace\_Int$, whenever a measured value ($acler\_sensed$) from the accelerometer sensor goes higher or lower than the activity threshold acc_thr.

```
EVENT Increase_Interval
  WHEN
    grd1 : $acler\_sensed > acc\_thr$
  THEN
    act1 : $Pace\_Int := 60000/MSR$
  END
```

```
EVENT Decrease_Interval
  WHEN
    grd1 : $acler\_sensed < acc\_thr$
  THEN
    act1 : $Pace\_Int := 60000/LRL$
  END
```

Finally, we have modeled all the functional and parametric requirements of different operating modes of the double electrode pacemaker system using stepwise refinements. We have also discovered the hierarchical development and relationship among all operating modes (see Figure-4) of double electrode cardiac pacemaker.

5 Model Validation and Analysis

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

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

This section conveys the validity of the model by using ProB tool [34, 28] and Proof Statistics. "Validation" refers to the activity of gaining confidence that the developed formal models are consistent with the requirements, which expressed in the requirements document [13]. We have used the ProB tool [34] that supports *automated consistency checking* of Event B machines via model checking [17] and constraint-based checking [25]. Animation using ProB worked very well and we have then used ProB to validate the Event B machine. This tool assists us to find potential problems, to improve invariants expressions in our EVENT B models, for instance by generating counter-examples when it discovers an invariant violation. ProB may help in improving invariant expression by suggesting hints for strengthening the invariant and each time an invariant is modified, new proof obligtions are generated by the RODIN platform. It is the complementary use of both techniques to develop formal models of critical systems, where high safety and security are required. More errors are corrected during the elaboration of the specifications while discharging the proof obligations and careful cross-reading than during the animations. We have validated all operating modes of the pacemaker in each refinement of models. The pacemaker specification is developed and formally proved by the RODIN tool.

ProB was very useful in the development of the pacemaker specification, and was able to animate all of our models and able to prove the absence of error (no counter example exist). The ProB model checker also discovered several invariant violations, e.g., related to incorrect responses or unordered pacing and sensing activities. It was also able to discover a deadlock in two of the models, which was due to the fact that "clock counter" were not properly recycled, meaning that after a while no pacing or sensing activities occur into the system. Such kind of errors would have been more difficult to uncover with the prover of RODIN tool.

Model	Total number of POs	Automatic Proof	Interactive Proof
Abstract Model	166	125(76%)	41(24%)
First Refinement	211	190(90%)	21(10%)
Second Refinement	67	66(99%)	1(1%)
Total	444	381(86%)	63(14%)

Table-3 : Proof statistics

The Table-3 is expressing the proof statistics of the development in the RODIN tool. These statistics measure the size of the model, the proof obligations generated and discharged by the RODIN prover, and those are interactively proved. The complete development of double electrode pacemaker system results in 444(100%) proof obligations, in which 381(86%) are proved automatically by the RODIN tool. The remaining 63(14%) proof obligations are proved interactively using RODIN tool. In the model, many proof obligations are generated due to the introduction of new functional behaviors and their parameters (threshold, 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 abstract level and the 1st refinement. These proof are quite simple, and achieve with the help of *"do case"* and instantiation of the constants and variables. The guards of some events are very complex, so for proving the invariants and the theorems, we simplify the guards using *"do case"*. The first abstract level is in detailed due to introduction of pacemaker's actuators and sensors of two electrodes for both heart chambers with proper synchronized properties under real time constraints.

6 Conclusion and Future Works

In this report, we have presented the formal specification of the double electrode pacemaker, is a grand challenge that is proposed by the Verified Software Initiative. We have used the EVENT B formal language on RODIN platform to develop the formal model of the operating modes of the double electrode pacemaker. Our approach for formalizing and reasoning about functional behaviour of pacing and sensing activities of the pacemaker based on action-reaction under real-time constraints.

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 [4] and time based patterns [15, 36] 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 double electrode pacemaker system. For quick understanding, we have formalized several different developments, each highlighting a different aspect of the problem, making different assumptions about the operating modes and establishing different properties. For example, we have considered a case of constant pacing in both chanbers, sensing and pacing synchronously, threshold parameter for an electrode sensor and rate modulation operating modes. In a stepwise refinement, we have also discovered the hierarchical development and relationship among double electrodes operating modes of the pacemaker (see Fig. 4).

Our 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 double electrode pacemaker system at abstract level secondly it introduces the peculiarity of the specific properties. 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).

The consistency of our specification has been checked through mathematical reasoning and validation experiments are performed by ProB model checker regarding safety conditions. As part of our reasoning, we have proved that the initialisation of the system is a valid one and we have calculated the preconditions of the operations. The latter has been executed to guarantee that our intention to have total operations has been fulfilled. At every stage of refinement we introduced the new parametric functionality of the system and proved the *consistency* and *refinement checking*. We introduced the invariants at refinement level that the initialisation of the whole system is valid. Proofs were quite simple, and achieved with the help of *instantiation* and *"do case"*. The guards of some events are very complex, so for proving the invariants, we simplify the guards using *"do case"*. In total, we have proved 444(100%) proof obligations, in which 381(86%) are proved automatically and remaining 63(14%) proof obligations are proved interactively using RODIN tool (see Table-3). Finally, we have validated the double electrode pacemaker system using the ProB model checker as validation tool and verify the correctness of our proved double electrode pacemaker system under the real-time constraints.

In the future, we have planned to create a formal proof based simulator for the single and double electrode pacemaker. It can be used by the doctor to analyze the real-time heart signal and predict the operating modes. As far as, it can be also used as a diagnostic tool to diagnose the patient and help to take the better decision for implanting a pacemaker [33]. For our on going research, we have contacted with physician and cardiologist experts to generalized the operating modes of single and double electrode pacemaker and derive the common parametric functional properties through refinement tree structures that help to design the automatic mode switching from one operating mode to another operating mode according to the heart pacing requirement.

References

[1] http://media.summitmedicalgroup.com/media/db/ relayhealth-images/nodes.jpg.

[2] A Reseach and Development Needs Report by NITRD. High-Confidence Medical Devices : Cyber-Physical Systems for 21st Century Health Care. http://www.nitrd.gov/About/MedDevice-FINAL1-web.pdf.

[3] J.-R. Abrial. $B^{\#}$: Toward a Synthesis Between Z and B. In D. Bert and M. Walden, editors, *3nd International Conference of B and Z Users - ZB 2003, Turku, Finland*, Lectures Notes in Computer Science. Springer, June 2003.

[4] J.-R. Abrial. *Modeling in Event-B: System and Software Engineering*. Cambridge University Press, 2009. Forthcoming book.

[5] J.-R. Abrial and D. Cansell. Click'n prove: Interactive proofs within set theory. In *TPHOL 2003*, pages 1–24, 2003.

[6] R. Back. On correct refinement of programs. *Journal of Computer and System Sciences*, 23(1):49–68, 1979.

[7] S. Serge Barold, Roland X. Stroobandt, and Alfons F. Sinnaeve. *Cardiac Pacemakers Step by Step*. Futura Publishing, 2004. ISBN 1-4051-1647-1.

[8] Dines Bjorner. *Software Engineering 1 Abstraction and Modelling*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. ISBN: 978-3-540-21149-5.

[9] Dines Bjorner. *Software Engineering 2 Specification of Systems and Languages*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. ISBN: 978-3-540-21150-1.

[10] Dines Bjorner. *Software Engineering 3 Domains, Requirements, and Software Design.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. ISBN: 978-3-540-21151-8.

[11] Dines Bjorner. *DOMAIN ENGINEERING Technology Management, Reserach and Engineering*, volume 4 of *COE Research Monograph Series.* JAIST, 2009.

[12] Dines Bjørner and Martin C. Henson, editors. *Logics of Specification Languages.* EATCS Textbook in Computer Science. Springer, 2007.

[13] Boston Scientific Boston Scientific: Pacemaker system specification, Technical report. 2007.

[14] Dominique Cansell and Dominique Méry. *The event-B Modelling Method: Concepts and Case Studies*, pages 33–140. Springer, 2007. See [12].

[15] Dominique Cansell, Dominique Méry, and Joris Rehm. *Formal Specification and Development in B*, chapter Time Constraint Patterns for Event B Development, pages 140–154. Lecture Notes in Computer Science. Springer US, 2006. ISSN 0302-9743.

[16] ClearSy, Aix-en-Provence (F). *B4FREE*, 2004. http://www.b4free.com.

[17] O. Grumberg E. M. Clarke and D. Peled. *Model Checking.* MIT Press, 1999. ISBN 978-0262032704.

[18] Kenneth A. Ellenbogen and Mark A. Wood. *Cardiac Pacing and ICDs.* 4th Edition, Blackwell, 2005. ISBN-10 1-4051-0447-3.

[19] E. Gamma, R. Helm, R. Johnson, R. Vlissides, and P. Gamma. *Design Patterns : Elements of Reusable Object-Oriented Software design Patterns.* Addison-Wesley Professional Computing, 1994.

[20] B. S. Goldman, E. J. Noble, J. G. Heller, and D. Covvey. The pacemaker challenge. *CMAJ*, 110(1):28–31, 1974.

[21] Artur Oliveira Gomes and Marcel Vinicius Medeiros Oliveira. Formal specification of a cardiac pacing system. In *FM 2009*, pages 692–707, 2009.

[22] Aaron Hesselson. *Simplified Interpretations of Pacemaker ECGs.* Blackwell Publishers, 2003. ISBN 978-1-4051-0372-5.

[23] C.A.R. Hoare, Jayadev Misra, Gary T. Leavens, and Natarajan Shankar. The verified software initiative: A manifesto. *ACM Comput. Surv.*, 41(4):1–8, 2009.

[24] Tony Hoare. The verifying compiler: A grand challenge for computing research. *J. ACM*, 50(1):63–69, 2003.

[25] Daniel Jackson. Alloy: a lightweight object modelling notation. *ACM Trans. Softw. Eng. Methodol.*, 11(2):256–290, 2002.

[26] Gary T. Leavens, Jean-Raymond Abrial, Don Batory, Michael Butler, Alessandro Coglio, Kathi Fisler, Eric Hehner, Cliff Jones, Dale Miller, Simon Peyton-Jones, Murali Sitaraman, Douglas R. Smith, and Aaron Stump. Roadmap for enhanced languages and methods to aid verification. In *Fifth Intl. Conf. Generative Programming and Component Engineering (GPCE 2006)*, pages 221–235. ACM, October 2006.

[27] Insup Lee, George J. Pappas, Rance Cleaveland, John Hatcliff, Bruce H. Krogh, Peter Lee, Harvey Rubin, and Lui Sha. High-confidence medical device software and systems. *Computer*, 39(4):33–38, 2006.

[28] Michael Leuschel and Michael Butler. *ProB: A Model Checker for B*, pages 855–874. LNCS. Springer, 2003.

[29] Charles J. Love. *Cardiac Pacemakers and Defibrillators.* Landes Bioscience Publishers, 2006. ISBN 1-57059-691-3.

[30] Hugo Daniel Macedo, Peter Gorm Larsen, and John Fitzgerald. *Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System Using VDM*, pages 181–197. LNCS. Springer, Los Alamitos, CA, USA, 2008.

[31] Jaakko Malmivuo. *Bioelectromagnetism.* Oxford University Press, 1995. ISBN 0-19-505823-2.

[32] Valerio Panzica La Manna, Andrea Tommaso Bonanno, and Alfredo Motta. Poster on a simple pacemaker implementation. ACM, May 2009.

[33] Writing Committee Members, Andrew E. Epstein, John P. DiMarco, Kenneth A. Ellenbogen, III Estes, N.A. Mark, Roger A. Freedman, Leonard S. Gettes, A. Marc Gillinov, Gabriel Gregoratos, Stephen C. Hammill, David L. Hayes, Mark A. Hlatky, L. Kristin Newby, Richard L. Page, Mark H. Schoenfeld, Michael J. Silka, Lynne Warner Stevenson, and Michael O. Sweeney. ACC/AHA/HRS 2008 Guidelines for Device-Based Therapy of Cardiac Rhythm Abnormalities: Executive Summary: A Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines (Writing Committee to Revise the ACC/AHA/NASPE 2002 Guideline Update for Implantation of Cardiac Pacemakers and Antiarrhythmia Devices): Developed in Collaboration With the American Association for Thoracic Surgery and Society of Thoracic Surgeons. *Circulation*, 117(21):2820–2840, 2008.

[34] ProB. The ProB animator and model checker for the B method. http://www.stups.uni-duesseldorf.de/ProB/overview.php/.

[35] Project RODIN. Rigorous open development environment for complex systems. http://rodin-b-sharp.sourceforge.net/, 2004. 2004–2007.

[36] Joris Rehm. 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.

[37] J. Woodcock and R. Banach. The verification grand challenge. 13(5):661–668, 2007.

[38] Jim Woodcock. First steps in the verified software grand challenge. *IEEE Computer*, 39(10):57–64, 2006.