Modeling Cardiac Pacemakers With Timed Coloured Petri Nets And Related Tools

MODELING CARDIAC PACEMAKERS WITH TIMED COLOURED PETRI NETS AND RELATED TOOLS

BY

MOHAMMED ASSIRI, M.A.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2021)	McMaster University
(Computing and Software)	Hamilton, Ontario, Canada

TITLE:	Modeling Cardiac Pacemakers With Timed Coloured Petri Nets And Related Tools
AUTHOR:	Mohammed Assiri M.A.Sc., McMaster University
SUPERVISOR:	Dr. Ryszard Janicki
NUMBER OF PAGES:	xviii, 259

To the memory of my late father (1956-2018)

ABSTRACT

Verification Grand Challenge is one of the Grand Challenges for Computing Research. Verification, which is the strict proof of the correctness of software according to its specifications, results in reliable software and potential cost reductions. In addition, the capabilities of other software engineering techniques, such as requirements analysis and testing, can be complemented and extended by verification. Medical devices are well-known examples of safety-critical systems which require significant advances in verification (among other areas). The failure of safety-critical systems not only damages property or the environment but could also lead to human fatality. Therefore, software verification is one approach to prevent such negative consequences.

The cardiac pacemaker is an electronic device that monitors and controls the heart rhythm via sensing and pacing operations. The pacemaker treats cardiac arrhythmia, defined as abnormal patterns of the heartbeat. The Software Quality Research

Laboratory at McMaster University proposed the pacemaker system specification as a pilot problem for the Verified Software Initiative.

This research utilizes formal methods to model and verify the interdisciplinary requirements of pacemaker systems. It additionally provides customizable data to assess and optimize various algorithms and parameters.

Keywords:

Cardiac Pacemakers · Verification Grand Challenge · Electrocardiogram · Timed Coloured Petri Nets · Biomodelling.

NOTATION AND ABBREVIATIONS

AP	Atrial Pace
ARP	Atrial Refractory Period
AS	Atrial Sense
ATR	Atrial Tachycardia Response
AV	Atrial-to-Ventricular
BOM	Bradycardia Operation Mode
BPM	Beats Per Minute
CCS	Cardiac Conduction System
CPN	Coloured Petri Net
CPS	Cardiac Pacemaker System
DCM	Device Controller-monitor
ECG	Electrocardiogram, external heart signals

EGM	Electrogram, internal heart signals
EP	Electrophysiology, electrophysiologist
FDA	Food and Drug Adminstration
HRL	Hysteresis Rate Limit
LRL	Lower Rate Limit
MSR	Maximum Sensor Rate
PG	Pulse Generator
ppm	Pulses Per Minute
PN	Petri Net
PVARP	Post-Ventricular Atrial Refractory Period
PVC	Premature Ventricular Contraction
SA	Senatorial Node
TCPN	Timed Coloured Petri Net
URL	Upper Rate Limit
VA	Atrioventricular Node
VP	Ventricular Pace
VRP	Ventricular Refractory Period

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

CHAPTER

1

INTRODUCTION

In this chapter, section 1.1 presents a background for this research. Section 1.2 defines the research aims, while section 1.3 addresses the involved scope of this work. Finally, section 1.4 illustrates the structure of the thesis.

1.1 Background

Verification Grand Challenge is one of the Grand Challenges for Computing Research that has been proposed in [Barbosa et al. [2013]]. Verification is the strict proof of the correctness of software according to its specification, resulting in reliable software and usually cost reductions [Hoare et al. [2009]]. In addition, the capabilities of other software engineering techniques, such as requirements analysis and testing, can be complemented and extended by verification.

Medical devices are well-known examples of safety-critical systems [Knight [2002]], which required significant advances in verification (among other areas). The failure of safety-critical systems not only damages property or the environment but also could lead to the death of people. Between 2006 and 2011, the US Food and Drug Administration (FDA) reported 2,294 fault cases and 1,154,451 side effects cases due to the faults of medical devices. Among those cases, there were 92.600 cases of patient injury and 4.590 cases of patient death. Almost 23 percent of these failures are related to software applications of medical devices [Alemzadeh et al. [2013]]. Therefore, software verification is one approach to prevent such faults.

The cardiac pacemaker (pacemaker thereafter) is an electronic device that monitors and controls the heart rhythm via sensing and pacing operations. The pacemaker treats abnormal patterns of the heartbeat, which is called cardiac arrhythmia. The Software Quality Research Laboratory at McMaster University proposed the pacemaker system specification [Boston Scientific [2007]] as a pilot problem for the Verified Software Initiative [Woodcock [2006]].

1.2 Research Aims

This thesis aims to verify the pacemaker system specification that represents its functions and operating properties. This can be translated into three objectives. Formal methods based on Timed Coloured Petri Nets (TCPNs) to model and verify the interdisciplinary requirements of pacemaker systems. Formal methods to precisely present and produce realistic synthetic events of the cardiac electrical activities to validate the pacemaker's various parameters. Finally, formulating a system to build proper and related datasets from the above models' processed data for further analysis, optimization and employment.

1.3 Scope

The work of this thesis involves studying the applicability of Timed Coloured Petri Nets (TCPNs) for the modelling and analysis of safety-critical systems. Timed Coloured Petri Nets were utilized to verify and validate the pacemaker specification and parameters. In addition, it assessed the feasibility of applying an evolutionary approach to develop a model for an artificial cardiac pacemaker.

1.4 Structure of the Thesis

The structure of this research is as follows:

- Chapter 2 provides backgrounds for the pacemaker system and environment. It also discusses related work and system requirements.

- Chapter 3 introduces the Timed Coloured Petri Nets (TCPN), which is the used modelling language of theproposed models.
- Chapter 4 presents in detail the TCPN-based modelling of the Cardiac Conduction System (CCS).
- Chapter 5 extends the model from chapter 4 to include the TCPN-based modelling of the Cardiac Pacemaker System (CPS).
- Chapter 6 concludes with the contributions of the thesis, related publications, and future work.

CHAPTER

2

THE PACEMAKER SYSTEM AND ENVIRONMENT

This chapter discusses briefly the environment where the pacemaker is intended to operate in section 2.1 and the pacemaker systems as in section 2.2.

2.1 Environment

Section 2.1.1 introduces the environment. Section 2.1.2 demonstrates the natural pacemaker of the heart. Section 2.1.3 shows the typical ECG waveform. Finally, section 2.1.4 briefly discusses related studies.

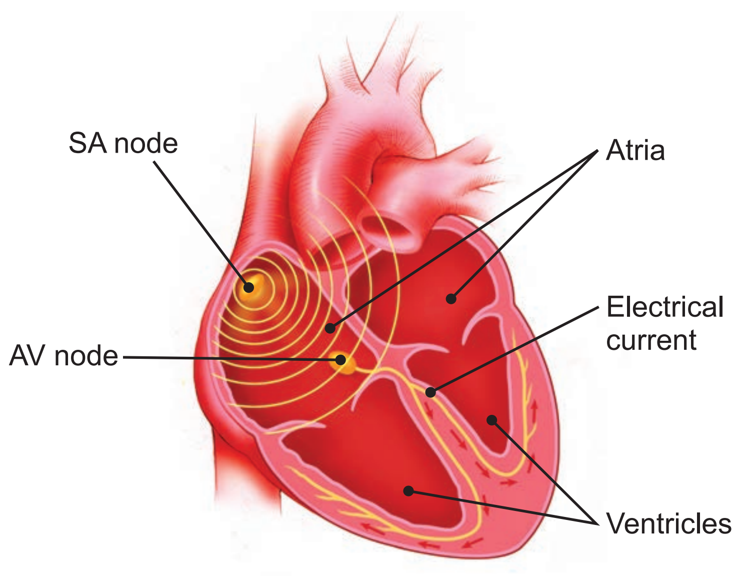
2.1.1 Introduction

As the pacemaker is an embedded system, it is intended to operate in an environment including the heart. The heart spontaneously produces electrical impulses that cause its muscles to contract and circulate the blood in each heartbeat. Such cardiac electrical activities can be recorded, as events of signals representing voltage over time, when a series of electrodes are appropriately placed onto the human body's surface. This recording is called the electrocardiogram (ECG). These ECG events assist in providing meaningful information regarding cardiac arrhythmias, cardiovascular disease, and the pacing functionality of an implanted artificial pacemaker.

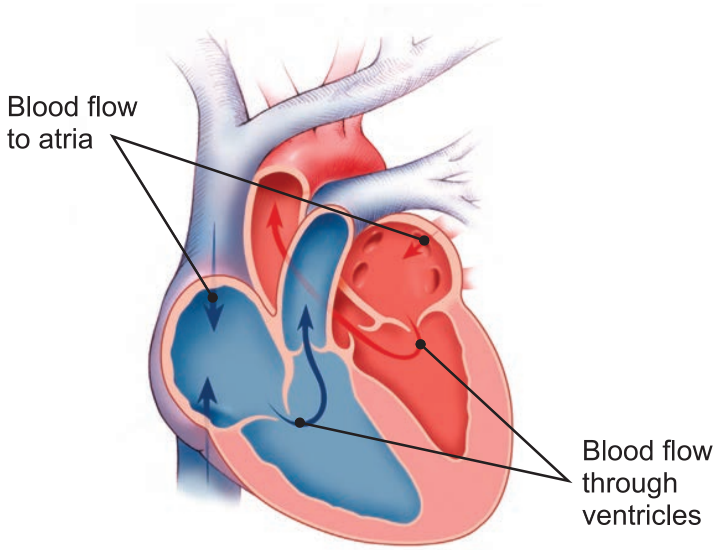
The ECG is the recording of cumulative signals generated by the cell population at a given time eliciting changes in their membrane potentials. The ECG does not directly measure the cardiac depolarization and repolarization but rather the changes of the cardiac electrical activities over time citeiaizzo2009handbook. As the ECG is a standard method to exam the heart's rate and to draw some conclusions, ECG, therefore, is utilized in this work as an instrument reflecting some information concerning cardiac activities.

2.1.2 The Natural Pacemaker of the Heart

A brief illustration of the functionality of the heart's electricity will assist in understanding the functionality of the pacemaker system. The heart is not only a mechanical pump but also an electrical organ that produces electrical signals via the sino-atrial (SA) node (specialized cells in the right atrium) through the atrioventricular (AV) node. As shown in Figure 2.2 [Boston Scientific [2014]], the two atria concurrently contract and fill the two lower ventricles with blood once the signals of the sino-atrial node reach them. The electrical signal of the sino-atrial also causes the contraction of ventricles, which pumps blood out to the body. This cycle periodically and continuously occurs [Boston Scientific [2014]].



(a) The heart and its electrical pathways



(b) The blood flow

Figure 2.2: The heart and its blood flow

2.1.3 The ECG Waveform

The Standard ECG consists of waves, complexes, intervals, and segments in accordance with the phases of the cardiac conduction. Some signals of ECG patterns indicate the occurrences of specific electrocardiographic activities. Under normal conditions, the most typical events include *P wave*, *QRS complex*, *T wave*, and sometimes *U wave* as shown in Figure 2.3.

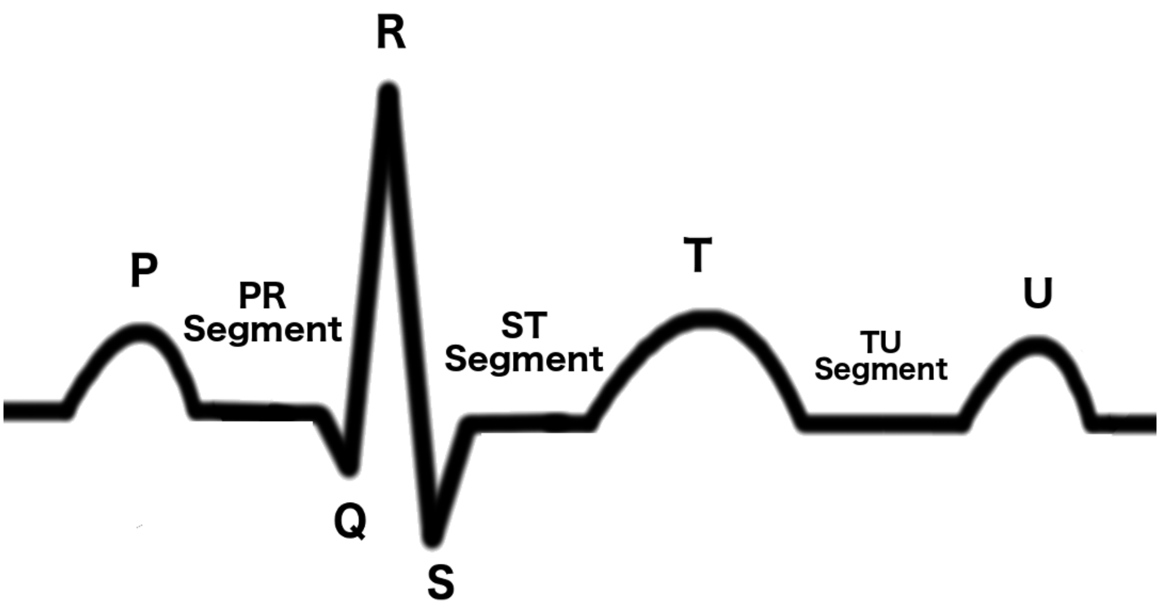


Figure 2.3: The typical ECG waveform measured from Lead II position

The normal cardiac cycle starts with the stimulation of the senatorial node (SA), which is located within the right atrium. This initial stimulation is not detected by the typical ECG as the SA is not composed of an adequately large quantity of cells to produce a detectable electrical potential. The depolarization of SA is conducted then throughout the atria, which can be observed as **P wave** in the ECG. The action potentials, which then spread throughout the atrioventricular node (VA) and the His bundle, are not large enough to be detectable by the ECG. The **QRS complex** indicates the depolarization result of the ventricles. Simultaneously, the effects of the atrial repolarization are masked by this QRS complex due to the larger amount of tissues causing the ventricular depolarization. Thus, atrial repolarization usually is undetectable in the ECG. The potential of ventricular repolarization is represented in the ECG as **T wave**. Finally, in some individuals, **U wave** represents the late repolarization of papillary muscles.

As each electrode represents a different view of the cardiac electrical activities in a particular direction, the morphology of the ECG waves can consequently vary. For the heart anatomy and the resting remembrance potential, the reader is referred to [MacLeod [2013]]

2.1.4 Related Work

The accuracy of new biomedical signal processing algorithms is commonly analyzed with available real databases such as the PhysioNet database [Goldberger et al. [2000]]. Nevertheless, in different clinical settings with a range of noise levels and sampling frequencies, assessing the overall performance and validity can be challenging [McSharry et al. [2003]]. Therefore, further analysis with formal synthetic ECG events may efficiently and comprehensively improve the overall outcomes. The formal mathematical representation of ECG events must be inclusive and comprehensive to present a wide variety of rhythms. However, it must also be uncomplicated to facilitate the formulation of different algorithms.

The number of Petri Net-based models regarding the ECG is limited despite the expressive power, readability, and convenience of Petri Nets as one of the formal specification techniques [Ghezzi et al. [2003]]. In [Chin and Willsky [1989]], a framework for modelling the electrical events using stochastic Petri Nets was presented. Nevertheless, this study focused solely on P and R *waves*. Petri Nets have been utilized in modelling areas related to the ECG, such as in [Sobrinho et al. [2014]], which discussed the verification and validation of biomedical signal devices; in [Chiang [2015]], in which a rule-based reasoning model is created for mental stress assessment by combining fuzzy and associative Petri Net methodologies; and in [Shih et al. [2013]], which proposed ECG arrhythmias pattern identification by using associative Petri Net. Furthermore, approaches and formalisms other than Petri Nets have been used to model topics related to ECG such as in [Javed and Ahmad [2018], McSharry et al. [2003], Boulakia et al. [2010]] and the references therein.

2.2 The Pacemaker System

Section 2.2.1 introduces the pacemaker system. Section 2.2.2 describes the system components. Section 2.3 discusses relevant studies. Lastly, section 2.4 reviews the pacemaker system specification.

2.2.1 Introduction

The pacemaker system is a typical example of safety-critical and real-time systems where not only their failure or malfunction can lead to fatal damages, but also their correct behaviours depend rigorously on timing restrictions. A healthy heart is capable of beating sixty to one hundred times per minute when the body is relaxed. However, for many causes, when the heartbeat is abnormally slow, the body is insufficiently supplied by blood and oxygen and may dysfunctional. A pacemaker, therefore, may be required. The pacemaker is an implanted device that restores and maintains a normal level of the heartbeat, and consequently, regular daily activities can be resumed.

2.2.2 System Components

The pacemaker system consists of a pulse generator, device controller-monitor and one or more leads. The pulse generator (also called a device) is implanted under the skin in the pectoral area (below the collarbone). It holds a small computer with several electronic circuits and a safely sealed battery within its case. The pulse generator functions include monitoring the heart rhythm, delivering electrical energy (i.e., impulse), and storing information about the heart. Microcomputer-based equipment called a ”programmer” communicates with the pulse generator from outside the body via a wand over the skin. The programmer is used to adjust the settings of the pulse generator and retrieve the stored information. The lead is an insulated wire implanted in the heart through veins and then connects to the pulse generator. The lead transfers the heart signal to the pulse generator and returns energy from the pulse generator to the heart to maintain a healthy level of the heart rhythms. The pacemaker system is recommended to treat and reduce the risks associated with heart rhythm problems such as bradycardia (a slow heartbeat fewer than sixty beats per minute) [Boston Scientific [2014]].

The pacemaker can be a single-chamber, dual-chamber, or triple-chamber [Boston Scientific [2014] and Stanford Medical Center]. As names suggest, the single-chamber pacemaker has one lead placed into one heart chamber (either the right atrium or right ventricle). A common cause that a single-chamber is selected is in case of too slow signals produced by the sino-atrial node. The dual-chamber pacemaker has two leads placed into the right atrium and the right ventricle, respectively. The dual-chamber pacemaker helps treat too slow signals from the sino-atrial node, partial or complete block over the electrical pathway to the ventricles, and/or asynchronous timing sequence of the atrial and ventricular. The triple-chamber pacemaker has three leads placed into the right atrium and both ventricles (also referred to as a biventricular pacemaker). The triple-chamber pacemaker is used in advanced heart failure, such as weakened heart muscle, and it also resynchronizes the contractions of ventricles more efficiently.

Pacemaker devices are intended to be highly reliable. However, malfunctions may happen and cause a loss in the delivered therapy. Premature battery depletion, sensing or pacing issues, error codes, and/or loss of telemetry are some malfunctions of the pacemaker system. Product performance Reports, such as in [Scientific [2017]], include historical types and rates of malfunctions that have been occurred. Historical data may assist in understanding the reliability of products rather than predicting the future performance [Boston Scientific [2011]].

2.3 Related Work

The pacemaker system is one of the benchmark systems where numerous studies have been published. However, in this section, the determination criterion is as follows:

$$(\textit{Pacemaker System})\ \textit{and}\ (\textit{Petri Nets})$$

$$\textit{or}$$

$$(\textit{Pacemaker System})\ \textit{and}\ (\textit{Boston's Specification})\ \textit{and}\ (\textit{several operation modes})$$

This means that the presented studies covered the pacemaker system modelled by Petri Nets as shown in section 2.3.1, or the studies are not only based on the pacemaker system's specification published by Boston Scientific but also cover various operation modes and types as presented in section 2.3.2.

2.3.1 Petri Nets Formalism

Petri Nets have been adopted to model some aspects of pacemakers. In [Yang [2004]], the study's primary purpose was the remote diagnosis for failures of cardiac pacemakers. The study presented a model describing the operation of a combined synchronous pacemaker, which is a combination of an atrial-synchronous pacemaker and a demand pacemaker. When the atrial stimulus triggers the reset circuit, the oscillator is disabled, and the model is an atrial-synchronous pacemaker; otherwise, the model is a demand pacemaker (i.e., the output circuit is controlled by the oscillator.) The proposed model used twelve places as checkpoints that mentor the pacemaker's operational status (i.e., failure diagnosis). The result of pacemaker's conditions is then derived from the values of the checkpoints. The checked conditions include the power supply, atrial electrode, amplifiers, reset and output circuits, oscillator, vibrators, and ventricular electrode. Furthermore, the study presented a remote diagnosis model with the assumption of possible wireless communication that indicates when and why a pacemaker should be extracted. The model was integrated with the previous model presented by the study, and it was explained as four main parts, namely remote control function of the transmitter, common parts for the demand pacemaker and the atrial-synchronous pacemaker, the demand pacemaker, and the atrial-synchronous pacemaker.

Majma, N. et al. conducted several studies regarding pacemaker utilizing Petri Nets and other tools as in [Majma and Babamir [2014], Majma et al. [2015], Majma et al. [2017], and Majma et al. [2016]]. However, all the mentioned studies focused essentially on runtime verification and pacemaker behaviour (i.e., irrelevant studies).

2.3.2 Other Formalisms

Based on the specification from [Boston Scientific [2007]], the work of [Macedo et al. [2008]] used VDM to propose a pragmatic, incremental approach modelling the cardiac pacing system. The study introduced an abstract model, sequential and concurrent models, and distributed real-time model. These models are derived sequentially (i.e., incremental development). The initial model (i.e., the abstract model) represented the operating mode of the pacemaker. The sequential and concurrent models described the structure and computing of data; however, the concurrent model also helped analyze concurrency issues. The last model (i.e., the distributed real-time model) illustrated the distribution over CPUs in a topology determined by bus configuration. The models were validated through systematic test scenarios where the requirements from [Boston Scientific [2007]] are confirmed to be achieved. This study is claimed to be the first attempt at the pacemaker challenge organized by the Software Quality Research Laboratory at McMaster University. Nevertheless, this study did not comprehensively cover the full pacemaker specification, and there were some limitations in the validation, such as limit scheduling assumptions in the scheduling models.

The **Z** notation was used to present a formal specification of a cardiac pacing system as in [Gomes and Oliveira [2009]]. In [Gomes and Oliveira [2009]], a single state was presented as the base of the overall state of the pacemaker following the specification of [Boston Scientific [2007]] and [Li and Lara-Rosano [2000]]. Two modules were presented based on a single state, namely the pacing pulse module and the sensing module. Even though this study provided a formal specification of pacing system using **Z** notation with features such as event marketer, restrictions on the parameter changes, and functionalities based on the battery level, it faced a long delay in loading because of the extensive use of the schema calculus. As well, this study included over 250 schemas, which are counted on eighty-four pages (over 4000 lines of **Z** specification) for most (but not all) of requirements of the specification [Boston Scientific [2007]]. The validation of this study depends on reasoning to partially check specification (i.e., no safety conditions were performed). The total proven theorems were forty-six in over 1291 lines of proof script. The last results of [Gomes and Oliveira [2009]] have been translated using a tool named Perfect Developer [Carter et al. [2005]] and presented in [Gomes and Oliveira [2010]]. The outputs were verified and then refined into high-level programming languages.

RTS is a process algebra-based formalism with timed extensions to CSP [Tuan et al. [2010]]. The study of [Tuan et al. [2010]] presented an approach that is similar to the approach of [Gomes and Oliveira [2009]]. Nevertheless, the study of [Tuan et al. [2010]] was distinguishable by modelling the pacemaker on the timed aspect's fundamental. The proposed **RTS** model was constructed as a parallel of timed processes. The safety properties, which is, in this case, maintaining a normal heart rate, were defined as LTL (Linear Temporal Logic) formulae and timed refinements. In this study [Tuan et al. [2010]], four main parts were proposed: environment, sensor, rate controller, and pulse generator. Both the environment and rate controller were each modelled as a single process where the first generated events. At the same time, the latter determined the changes in heart rate and included the sensor model. The pulse generator, as the name suggests, generated paces upon needs. The study defined several categories of properties as assertions in the Process Analysis Toolkit (PAT).

PAT is a model checker applying state-of-the-art model checking techniques for system analysis. The defined categories of properties included deadlock freeness, lower and upper rate limits, refractory period, atrial-ventricular delay, and rate-controlling. The results from the proposed **RTS** models satisfied all the defined properties efficiently. Nevertheless, this study [Tuan et al. [2010]] was restricted to sensing and pacing behaviours. Complex behaviours, such as AtrialTachycardiaResponse [Boston Scientific [2007]], were uncovered.

The pacemaker was formally model in PROMELA using SPIN model checking tool in [Sharma [2010]]. The approach of [Sharma [2010]], which was inspired from [Macedo et al. [2008]], was evolved from sequential to concurrent and eventually to a distributed model using SPIN. It also verified and validated end to end (from requirements to implementation) throughout generating and validating C code from the proposed PROMELA model. Both [Tuan et al. [2010]] and [Sharma [2010]] are similar, however, the study of [Tuan et al. [2010]] used CSP and PAT while [Tuan et al. [2010]] used PROMELA and SPIN. The study of [Sharma [2010]] included five LTL properties that were not introduced in [Tuan et al. [2010]]; they are pace limit, triggering property, inhibiting property, tracked property, and hysteresis limit. Nevertheless, this study [Sharma [2010]] lacked some advanced modes such as rate controlled pacing and hysteresis pacing. The LTL properties for the distributed model were not fully validated for all the timing constraints. It did not include non-timing parameters such as amplitude width or supported features such as the pacing generator's diagnosis mode.

In [Méry et al. [2011]], an incremental approach was used to propose a formal model of the pacemaker. This study used Event-B as the modelling language and examined several properties. In this study [Méry et al. [2011]], the functions of pacemaker systems can be expressed by analyzed action-reaction and real-time patterns. The bounded time interval for every action, reaction, and action-reaction pair are the most common elements in the pacemaker system. Consequently, two design patterns were applied as action-reaction and time-based patterns. The action-reaction pattern was based on the relationship between the action and the corresponding reaction. In contrast, the time-based pattern was based on timed automation that presented the synchronization of the sensing and pacing stimulus functions. The models of this study [Méry et al. [2011]] covered all operating modes of both the one-electrode cardiac pacemaker and the two-electrode cardiac pacemaker. They were validated through supportive tools for 780 proof obligations.

The pacemaker and the heart were also modelled in a closed-loop system utilizing networks of timed automata as in [Jiang et al. [2014]]. The verification of this closed-loop system was conducted in the model checker UPPAAL where safety properties were defined in Timed Computational Tree Logic (TCTL). A series of abstractions was applied to drive the development of the models. In this study, the heart model produces cardiac events and reacts to pacemaker operations. To manage the complexity and accuracy of the models, a manual Counter-Example-Guided Abstraction and Refinement (CEGAR) framework was implemented. In [Jiang et al. [2014]], the case of a dual-chamber pacemaker was utilized to explain the verification of medical devices' specifications. This study analyzed two cases of cardiac arrhythmia along with the designated mode by the pacemaker. Two potential safety violations were found in this study and required further details of the pacemaker in order to properly refine the pacemaker model.

For further details about the pacemaker challenge, the reader is referred to [Méry et al. [2014]] and the references therein.

2.4 System Requirements

The pacemaker system is a hybrid embedded real-time system, and its development requirements are defined in Boston Scientific [2007]. The work of this thesis concentrates on the following requirements:

2.4.1 Model Type

The Pacemaker shall support single and dual chamber rate-adaptive pacing.

2.4.2 Pacing Pulse

The device shall output pulses with programmable voltages and widths (atrial and ventricular), which provide electrical stimulation to the heart for pacing.

- Pulse Amplitude
  - The atrial and ventricular pacing pulse amplitudes shall be independently programmable.
- Pulse Width
  - The atrial and ventricular pacing pulse width shall be independently programmable.
- Sensitivity Adjustment

– A means shall be provided for the physician to manually adjust the sensing threshold of the device for both the ventricular and atrial sense channels.

2.4.3 Bradycardia Operating Modes

The following bradycardia operating modes shall be programmable: Off, DDDR, VDDR, DDIR, DOOR, VOOR, AOOR, VVIR, AAIR, DDD, VDD, DDI, DOO, VOO, AOO, VVI, AAI, VVT and AAT.

OVO, OAO, ODO, and OOO shall be available in temporary operation. These operating modes are interpreted in Table 2.1.

Table 2.1: The interpretation of the pacemaker's operating modes

Operating Mode	interpretation
Off	The pacemaker is OFF
OOO	Temporary operation mode without any pacing nor sensing capabilities
OAO	A temporary operation mode where only the atrium is sensed without any pacing capabilities
OVO	A temporary operation mode where only the ventricle is sensed without any pacing capabilities

. . . continued

. . . continued

Operating Mode	interpretation
ODO	A temporary operation mode where both the atrium and ventricle are sensed without any pacing capabilities
AOO	An asynchronous operation mode where only the atrium receives a pacing stimulus from the pulse generator at a fixed rate without any sensing capabilities
VOO	An asynchronous operation mode where only the ventricle receives a pacing stimulus from the pulse generator at a fixed rate without any sensing capabilities
DOO	An asynchronous operation mode where the atrium and ventricle each receives a pacing stimulus independently from the pulse generator at separate fixed rates without any sensing capabilities

. . . continued

. . . continued

Operating Mode	interpretation
AAI	An atrial demand pacing operation mode where the atrium is sensed and only paced when a spontaneous atrial depolarization is undetected.
AAT	A triggered operation mode where the atrium is sensed and consequently paced when a spontaneous atrial depolarization is detected.
VVI	A ventricular demand pacing operation mode where the ventricle is sensed and only paced when a spontaneous ventricular depolarization is undetected.
VVT	A triggered operation mode where the ventricle is sensed and consequently paced when a spontaneous ventricular depolarization is detected.

. . . continued

. . . continued

Operating Mode	interpretation
VDD	A synchronous operation mode where both the atrium and ventricle are sensed and only the ventricle is paced when a spontaneous ventricular depolarization is undetected after a programmable AV delay. (i.e. this mode is triggered by a spontaneous atrial depolarization and inhibited by a spontaneous ventricular depolarization).
DDD	A synchronous operation mode where both the atrium and ventricle are sensed. Accordingly, when a spontaneous atrial depolarization is undetected within a defined timer, the atrium is paced, and similarly, when a spontaneous ventricular depolarization is undetected within a defined timer, the ventricle is paced.

. . . continued

. . . continued

Operating Mode	interpretation
DDI	A synchronous operation mode where both the atrium and ventricle are sensed. Consequently, when a spontaneous atrial depolarization is detected, the atrium's pacing is inhibited with no impact on the ventricle's pacing rate. Yet, when a spontaneous ventricular depolarization is detected, the subsequent pacing capabilities for both the atrium and ventricle are inhibited within the corresponding cycle.
AOOR, VOOR, DOOR, AAIR, VVIR, VDDR, DDIR, DDDR	Similar to the above operation modes. As well, any operation mode with assigned rate-adaptive pacing, i.e. XXXR, indicates the pacing capabilities are subject to a connected sensor regulating the heart rate according to the patient's activities.

The operation modes of the pacemaker are identified using a *Pacemaker Code*, which is a sequence of four valid letters representing four categories. Table 2.2 shows the categories and their valid letters. For example, the mode *AOO* represents that the pacemaker paces the Atria without sensing any chamber. *AAI* mode indicates that the pacemaker paces and senses the Atria and releases a pulse when there is no natural pulse. Groping operating codes are also possible by using the letter X as *wildcard notation* to denote any letter. For example, the mode AXX represents all modes that start with an A, such as AAI and AAT - along with others.

Table 2.2: The operating modes

Category	**I** Chambers Paced	**II** Chambers Sensed	**III** Response To Sensing	**IV** (optional) Rate Modulation
Letters	O–None A–Atrium V–Ventricle D–Dual	O–None A–Atrium V–Ventricle D–Dual	O–None T–Triggered I–Inhibited D–Tracked	R–Rate Modulation

- No Response To Sensing (O)
  - Pacing without sensing is asynchronous pacing. During asynchronous pacing, paces shall be delivered without regard to senses.
- Triggered Response To Sensing (T)
  - During triggered pacing, a sense in a chamber shall trigger an immediate pace in that chamber.
- Inhibited Response To Sensing (I)
  - During inhibited pacing, a sense in a chamber shall inhibit a pending pace in that chamber.

- Tracked Response To Sensing (D)
  - During tracked pacing, an atrial sense shall cause a tracked ventricular pace after a programmed AV delay, unless a ventricular sense was detected beforehand.

2.4.4 Bradycardia States

The following bradycardia states shall be available: Permanent, Temporary, Pace-Now, Magnet, and Power-On Reset (POR). Operating states shall be mutually exclusive.

- Permanent State
  - The permanent pacing state is the normal state of operation of the device. The normal pacing parameters programmed shall be used in the permanent brady state.
- Temporary Bradycardia Pacing
  - The temporary brady pacing state is independent of other pacing functions. The temporary brady parameters programmed shall be used in the temporary brady state. The temporary state shall be capable of being used to temporarily test various system parameters or provide patient diagnostic testing. Temporary brady pacing shall be terminated by one of the following: breaking the telemetry link, a Pace-Now pace, or a DCM command to the device to cancel temporary pacing.
- Pace-Now State

- Commanded emergency bradycardia pacing (Pace-Now) shall be available. The Pace-Now Pace parameter values are as follows:

  1. The mode Pace-Now pace parameter shall have a value of VVI.
  2. The lower rate limit Pace-Now pace parameter shall have a value of 65 ppm $\pm$8 ms.
  3. The amplitude Pace-Now pace parameter shall have a value of 5.0 V $\pm$0.5 V.
  4. The pulse width Pace-Now pace parameter shall have a value of 1.00 ms $\pm$0.02 ms.
  5. The ventricular refractory Pace-Now pace parameter shall have a value of 320 ms $\pm$8 ms.
  6. The ventricular sensitivity shall have a value of 1.5 mV.
  7. The first Pace-Now pacing pulse shall be issued within two cardiac cycles plus 500 ms from the time of the last user action required to activate the Pace-Now state.
  8. Once initiated, Pace-Now pacing shall continue until the DCM changes the device pacing mode.

- Magnet State
  - The Magnet State is used during the Magnet Test.
- Power-On Reset (POR) State
  - A Power-on-reset (POR) state shall be entered when the battery voltage drops so low that PG operation is not predictable. All functions shall be

disabled until the battery voltage exceeds the POR trip voltage. Above this trip voltage, the PG enters the POR state which is used to power-up the PG system to a known state and set of parameters. The POR parameter values are as follows:

1. The mode POR pace parameter shall have a value of VVI.
2. The lower rate limit POR pace parameter shall have a value of 65 ppm ± 8 ms.
3. The amplitude POR pace parameter shall have a value of 5.0 V ± 0.5 V.
4. The pulse width POR pace parameter shall have a value of 0.5 ms ± 0.02 ms.
5. The ventricular refractory POR pace parameter shall have a value of 320 ms ± 8 ms.
6. The ventricular sensitivity shall have a value of 1.5 mV.

2.4.5 Magnet Test

The magnet can be used to determine the battery status of the device. A standard cardiac donut magnet shall be detected by the device at a distance of 2.5 cm between the center of the labeled surface of the device and the surface of the magnet. When the magnet is in place, the device shall:

1. Pace asynchronously with a fixed pacing rate. The device mode shall be AOO if previous mode was AXXX, VOO if previous mode was VXXX, DOO if previous mode was DXXX, or OOO if previous mode was OXO modes.

2. At BOL the magnet rate shall be 100 ppm. At ERN the magnet rate shall decrease to 90 ppm. At ERT the magnet rate shall decrease further to 85 ppm. During post-ERT operation the rate interval may gradually decrease as the battery voltage continues to decrease.

3. When the magnet is removed the device shall automatically assume pretest operation.

4. The magnet mode shall have the capability to be programmed OFF, so that it will ignore magnet detection.

2.4.6 Bradycardia Therapy

User programmable parameters are provided for controlling the delivery of patient-tailored, bradycardia therapy. These parameters are described in Appendix A; which parameters are meaningful with which pacing mode are listed in Appendix B, Programmable Parameters for Bradycardia Therapy Modes.

CHAPTER

3

TIMED COLOURED PETRI NETS

This chapter presents the concepts of Timed Coloured Petri Nets (TCPN). Section 3.1 introduces Coloured Petri Nets. Section 3.2 discusses informal introduction to Coloured Petri Nets. Section 3.3 defines Coloured Petri Nets and Timed Coloured Petri Nets. Finally, section 3.2.7 presents the related computer tools of Timed Coloured Petri Nets.

3.1 Introduction

Coloured Petri Nets (CP-nets or CPNs), first proposed in [Jensen [1981]] and later substantially modified and enhanced in [K. Jensen [1994]], are an extension of Petri Nets (c.f. [Reisig [1991]]) which are often used to model behaviours of rather complex systems. CPNs preserve useful properties of Petri Nets and at the same time extend beyond initial formalism to allow for the distinction between tokens. Coloured Petri Nets is a graphical language for constructing models of concurrent systems and analyzing their properties. CPNs is a discrete-event modelling language combining the capabilities of Petri Nets with the capabilities of a high-level programming language. Petri Nets provide the foundation of the graphical notation and the basic primitives for modelling concurrency, communication, and synchronization. Coloured Petri Nets allow tokens to have a data value attached to them. Such an attached data value is called token colour. Although the colour can be of an arbitrarily complex type, places in CPNs usually contain tokens of one type. This type is referred to as the colour set of the place.

3.2 Informal Introduction to CPNs

Since the proposed models are based on Timed Coloured Petri Nets (TCPNs), a brief but proper informal introduction is presented to cover the main graphical notations of CPN-based models, CPN ML language basics, and the concepts of time in CPNs. In addition, an example system is illustrated to interpret the presented concepts further.

3.2.1 Example System

In this chapter, a part of the pacemaker's life cycle is utilized as an example system demonstrating Timed Coloured Petri Nets. This system is meant as only a simplified example for the purpose of introducing the main graphical notations of CPN-based models and the basics of CPN ML language. For further details around the pacemaker's life cycle, the reader is referred to [Boston Scientific [2007]]. The specification of the modelled example system is as follow:

- The pacemaker's life cycle shall start when a physician implants the device into the patient.
- In the system, the physician shall have an identified number and a name.
- The patient shall have a file number, a name, and their health condition.
- Each pacemaker shall store its serial number, operation mode, and timestamp of its events.
- During a follow-up visit, the physician shall set the settings according to the patient's needs if necessary.
- Frequent follow-up visits shall not be covered in this example system.
- Following the follow-up visit, the pacemaker shall store both old and new modes.
- After some time, the physician shall properly explant the device.

3.2.2 The Structure of CPN-based Models

The graphical structure of a CPN-based model can be built using places, transitions, and arcs as shown in Figure 3.1.

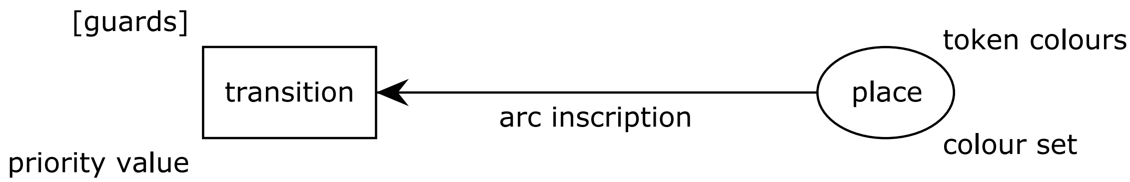


Figure 3.1: The essential components of CPN-based models

Places are drawn as ellipses that represent the states of the desired model. Each place is associated with one colour set and either the multi-set of the token colours or the empty set. The colour set, which is written as an inscription near the place, determines the syntactic format of the data values that are attached to tokens. The possible color sets are given in section 3.2.3.

Arcs are drawn as arrows that guide the flow of the model. Each arc has an inscription that declares the quantities explicitly and the types of the transferred token colours. Each arc is either an input arc or an output arc. An input arc connects a place to a transition, and an output arc connects a transition to a place.

Transitions are drawn as rectangles that represent the events of the model. Each transition has input and output places to which they are connected through arcs (i.e., input places are those linked by arcs going into transitions while output places are connected by arcs going from transitions). In addition, each transition may be associated with guards and/or a priority value. By default, they are located at the top-left corner and the bottom-left corner of the transition, respectively. The guards enable or disable the transitions while the priority values administer the sequence of enabled transitions.

Places and transitions may have names, which are written inside them. Even though names are not formal (i.e., do not affect the execution of the model), meaningful names can dramatically improve the model's readability.

In the given example system, one possible way to model it is by representing the system's states and stakeholders using places. The proposed places for the system are addressed in Table 3.1.

In this thesis, colour sets are expressed in two forms; mathematical form (as in Table 3.2) and programmatical form (as in Listing 3.1). The former form provides more precise definitions of the sets. The latter form is based on the CPN ML language, and it shows the designated translation of the mathematical forms into the CPN ML forms.

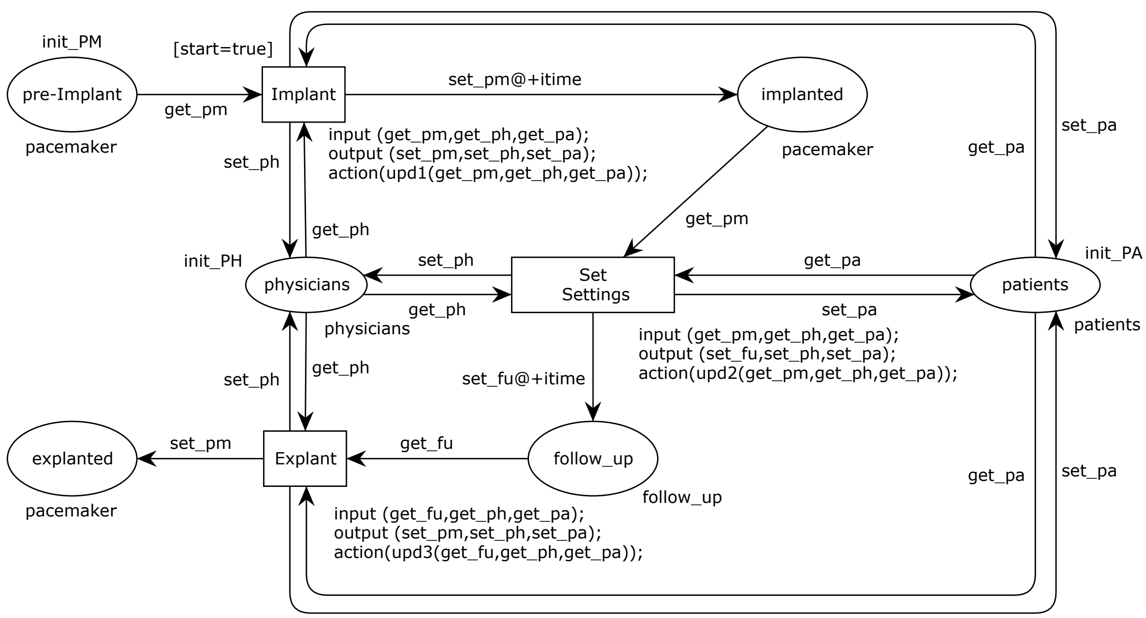


Figure 3.2: The example system illustrating TCPNS

Table 3.1: The places of the example system

Place	Colour set	Representation
physicians	**physicians**	*Any physician who interacts with the demanded actions such as implanting the pacemaker*
patients	**patients**	*Any patient who receives the treatment of the pacemaker*
pre-implant	**pacemaker**	*All the stages that are prior to the implant of the pacemaker*
implanted	**pacemaker**	*All the stages that are after the implant of the pacemaker but before the follow-up stages*
follow_up	**follow_up**	*It includes all the stages that occur between the implant and the explant of the pacemaker*
explanted	**pacemaker**	*The final stages that complete the life cycle of the pacemaker*

Table 3.2: The definitions of colour sets of the example system

Colour Set	Definition
string	*A set of characters that include letters, spaces, numbers, punctuation marks, and symbols*
physicians	$\{(id,\ name) \mid id \in \mathbb{Z}_{\geqslant 0},\ name \in string\}$
patients	$\{(file\ no,\ name,\ condition) \mid file\ no \in \mathbb{Z}_{\geqslant 0},\ name,\ condition \in string\}$
pacemaker	$\{(SN,\ mode,\ timestamp) \mid SN \in \mathbb{Z}_{\geqslant 0},\ mode,\ timestamp \in string\}$
follow up	$\{(SN,\ old\ mode,\ new\ mode,\ timestamp) \mid SN \in \mathbb{Z}_{\geqslant 0},\ old\ mode,\ new\ mode,\ timestamp \in string\}$

```
(* Color Sets *)
colset INT = int;
colset STRING = string;
colset physicians = record id:INT * name:STRING;
colset patients = record file_no:INT * name:STRING * condition:STRING;
colset pacemaker = record SN:INT * mode:STRING * timestamp:STRING timed;
colset follow_up = record SN:INT * old_mode:STRING * new_mode:STRING * timestamp:STRING timed;
```

Listing 3.1: The colour sets of the example system

3.2.3 CPN ML Language

CPN ML language is a flexible, expressive, and extensible language founded on the functional programming Language: **Standard ML** (SML/NJ implementation). CPN ML provides modelling ability similar to high-level programming languages, where both places and transitions are explicitly described, and data types and hierarchical decomposition are supported. Through the use of the CPN ML language, the CPN-based models have not only types and inscriptions but also colour sets, token colours, variable declarations, and functions.

The colour set can be simple or compound. A simple colour set is defined by any of the primary types such as *integer (int), real, string, Boolean (bool), or unit.* These types are inherited from Standard ML into CPN ML (also exist in major programming languages), and therefore they are self-explanatory. The construct *"with"* allows defining both a subset of a basic set and a simple enumeration set.

A compound set is a combination of different pre-defined colour sets. It is formed by one of three constructors; *product*, *record*, or *list*. The former two constructors differ syntactically because the *record* constructor labels each combined set while the *product* constructor does not. However, they are similar semantically as they define a fixed content structure. In contrast, the *list* constructor defines a flexible length list of a pre-defined set.

A non-empty token colour is constructed as "$M`S$" where M denotes the multiplicity of the token colour S. Multiple token colours are structured by a particular operator [++], as:

$$M1`S1 + +M2`S2 + +M3`S3\ldots$$

An arc inscription is an expression that evaluates to a multi-set. An expression is structured from constants, declared variables, defined functions, or a combination of them. Similar to expressions in programming languages, arc expressions also may contain arithmetic operators. Besides distributing token colours over places, arc expressions may also alter the values of transferred token colours.

A transition guard is a Boolean expression that evaluates to either true or false. It is enclosed by square brackets, and requires at least one of comparison operators such as $=, <> \ (means \neq), <, >, <= \ (means \leq)$, or $>= \ (means \geq)$. Additionally, a compound expressions is formed by logic operators such as *not*, *andalso* (means *and*), *orelse* (means *or*).

Functions are the best candidate for complex calculations or extended expressions. Functions may be utilized with arc inscriptions, transition guards, or initialized token colours. Note that node functions and arc expression functions allow multiple arcs to connect the same pair of nodes with different arc expressions.

The above concepts are interpreted clearer when any graphical example is seen as in Figure 3.3, for instance. Figure 3.3 shows a model that demonstrates the process of the buttons illumination. In this model, each button has a single state of either *the button is unilluminated* or *the button is illuminated*. To switch between the two states, we need two transitions, *illuminate* and *unilluminate*. Finally, the arcs connect the places and the transitions.

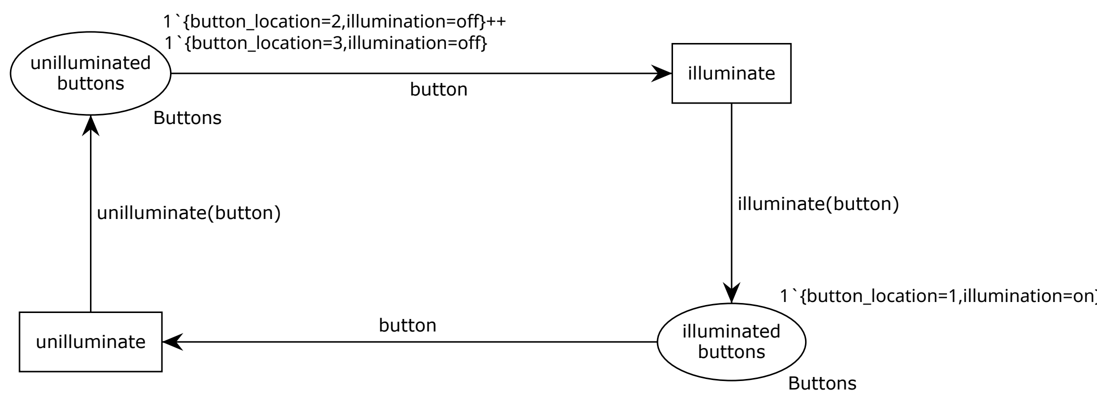


Figure 3.3: An example of utilizing CPN ML Language within a CPN-based model

In Figure 3.3, the CPN ML language defines the colour sets of the places and the arcs' expressions. Firstly, both places have the colour set **Buttons**, which is a compound colour set consists of two simple sets given in Listing 3.2. Consequently, each token has the button's number and illumination's state. Secondly, the arcs that connect the places to the transitions are called input arcs and expressed by the variable *button*. In contrast, the arcs that connect the transitions to the places are called output arcs and expressed by functions *illuminate* and *unilluminate* (see Listing 3.2).

```
colset button = int; (*simple colour set *)
colset switch = bool with (on,off); (*enumeration colour set*)
colset Buttons = record button_location:button *
        illumination:switch; (*compound colour set*)
var button:Buttons;(*variable*)
fun illuminate(button:Buttons) = (*function*)
 {button_location=(#button_location button), illumination=on}
fun unilluminate(button:Buttons) = (*function*)
 {button_location=(#button_location button), illumination=off}
```

Listing 3.2: Demonstrating the basics of CPN ML language

In Figure 3.3, the process of illuminating buttons starts from place *unilluminated buttons* when each token represents a button of the colour set **Buttons**. Illuminating a button requires transition *illuminate* to fire (i.e. be enabled). In this model, all transitions have no guards, and thereby they are always enabled as long as there is a token in the input places. After the firing of transition *illuminate*, a token is removed from the input place *unilluminated buttons* and placed through connected arcs into the output place *illuminated button*. The process of unilluminating buttons is performed similarly.

In addition, CPNs support hierarchical modelling in a fashion similar to programs being constructed from modules. For instance, Figure 3.4 has one place that represents the state of *unilluminated buttons*, and a hierarchical model that represents the *Illumination process*. Hence, Figure 3.4 and Figure 3.3 are considered equivalent in term of represeting the same system.

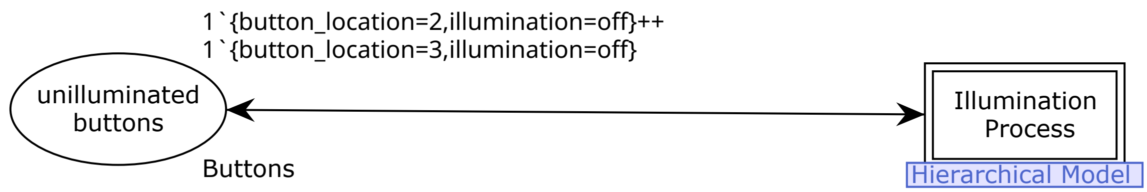


Figure 3.4: An example of a hierarchical CPN-based model

Revisiting the example system from section 3.2.1, its colour sets are defined in Table 3.2 and Listing 3.1. In this model and all the subsequent models, value declarations and variable declarations are employed to represent parameters and variables, respectively. While both declarations bind values to identifiers, value declarations, unlike variable declarations, work as constants whose values are holding during the model's execution. The parameters of the example system are provided in Listing 3.3. Typically, the selection of parameters is subject to the model's specifications.

Table 3.3: The parameters of the example system

Parameter	Value	Explanation
PM SN	$\mathbb{Z}_{\geqslant 0}$	*The serial number of the pacemaker*
initial mode	*string*	*The nominal operation mode of the pacemaker*
new mode	*string*	*The new operation mode after the follow-up visit*
ctime	*string*	*The timestamp is programmed to be automatically determined via the applying of predefined functions*
file no	$\mathbb{Z}_{\geqslant 0}$	*The file number of the patient*
p name	*string*	*The name of the patient*
p cond	*string*	*The health condition of the patient*
d name	*string*	*The name of the physician*
d id	$\mathbb{Z}_{\geqslant 0}$	*The identified number of the physician*
start	*Boolean*	*The value of the transition's guard controls the firing of the transition*
itime	$\mathbb{Z}_{\geqslant 0}$	*The assigned duration time unit for designated events*

```
(* parameters *)
val PM_SN = 13478621
val initial_mode = "DDD"
val new_mode = "VVD"
val ctime = Date.fmt "%Y-%m-%d %H:%M:%S" (Date.fromTimeUniv (
   Time.now ()))
val file_no = 4521876
val p_name = "John"
val p_cond = "symptoms"
val d_name = "David"
val d_id = 1545
val start = true
val itime = 50
```

Listing 3.3: Parameters of the example system

The variables of the arc inscriptions are presented in Listing 3.4. While CPNs accept any valid names for the variables, meaningful names can significantly improve the model's readability. Consequently, the arc inscriptions' variables involved the prefixes "*get*" with arcs connected input places to transitions and "*set*" with arcs connected transitions to the output places. Declaring distinct variables for the same colour set assist in properly adopting the code segments, which are added within the transitions' inscriptions as shown in Listing 3.5. Code segments are executed upon the firing of their parent transitions. Each code segment includes an optional input pattern, an optional output pattern, and a mandatory code action. The input pattern lists the associated variables that can be accepted but not changed by the code action. In contrast, the output pattern lists the variables that result from the execution of the code action. The code action is an ML expression that applies constants, operations and functions as in Listing 3.5, for instance.

```
(* arc var *)
var get_ph,set_ph:physicians;
var get_pa,set_pa:patients;
var get_pm,set_pm:pacemaker;
var get_fu,set_fu:follow_up;
```

Listing 3.4: Arc inscriptions of the example system

```
input (get_pm,get_ph,get_pa);
output (set_pm,set_ph,set_pa);
action(upd1(get_pm,get_ph,get_pa));
```

Listing 3.5: Code segment of transition ”Implant”

The initial token colour sets are defined in accordance with their associated colour set. In this example system, all colour sets are initialized with token colour sets, as in Listing 3.6, based on the above parameters' values.

```
(* Initial Token Colour Sets *)
val init_PM = {SN=PM_SN,mode=initial_mode,timestamp=ctime};
val init_PA = {file_no=file_no,name=p_name,condition=p_cond}
val init_PH = {id=d_id,name=d_name}
```

Listing 3.6: Initial token colour sets of the example system

In this example system, three functions were constructed to perform specific actions. All those functions were utilized with the code action pattern within the code segment explained above. Each time a transition is fired, the embedded function assists in facilitating the process of the model. In Listing 3.7, the function *upd1* primarily updates the timestamp of the token colour set **pacemaker** to the current timestamp upon the transition's firing, which indicates the timestamp of implanting the pacemaker.

```
fun upd1(pm:pacemaker,ph:physicians,pa:patients) =
({SN=PM_SN,mode=initial_mode,timestamp=ctime},ph,pa)
```

Listing 3.7: Function of transition "Implant"

Unlike the previous function, in Listing 3.8, the function *upd2* of the code action associated with the transition *Set Settings* not only updates the timestamp but also translates the token from the colour set **pacemaker** to the colour set **follow up**. Likewise, in Listing 3.9, the function *upd3* updates the timestamp and translates the token between the related colour sets.

```
fun upd2(pm:pacemaker,ph:physicians,pa:patients) =
let
   val sn = #SN pm
   val old_mode= #mode pm
   val timestamp= ctime
in
   ({SN=PM_SN,old_mode=old_mode,new_mode=new_mode,timestamp=
      timestamp},ph,pa)
end;
```

Listing 3.8: Function of transition "Set Settings"

```
fun upd3(fu:follow_up,ph:physicians,pa:patients) =
let
   val sn = #SN fu
   val mode= #new_mode fu
   val timestamp= ctime
in
   ({SN=PM_SN,mode=mode,timestamp=timestamp},ph,pa)
end;
```

Listing 3.9: Function of transition "Explant"

3.2.4 Time in CPNs

Time in CPNs can be represented in three different ways: firing duration, holding duration, and enabling duration [Bowden [2000]]. In this work, the holding duration technique is used to classify tokens according to availability where only available tokens can enable a transition. A duration is assigned to each transition in which, once fired, the removal and creation of tokens are conducted instantaneously. However, the created tokens are unavailable to enable new transitions until the tokens are placed into the output place for the time specified by the transition that created them. In [Jensen and Kristensen [2009]], the notion of token unavailability is defined implicitly by a timing attribute. Attached to desired tokens, this is called a timestamp preceded by the symbol @. Time in CPNs is represented as simulated time: a symbolic representation of time. Simulated time and real physical time have no intrinsic relationship whatsoever, and a built symbolic representation of real-time timestamps belongs to Time Set (TS), which is equal to $\mathbb{R}_{\geqslant 0}$. The timed markings are multi-sets on TS, where TS_{MS} is denoted by a timestamps collection.

In the example system, two colour sets are declared as timed colour sets; **pacemaker** and **follow up**. To execute a model with time in CPNs, at least one colour set must be timed by appending the keyword *timed* (see Listing 3.1). In addition, the processing time for events, such as implanting and updating settings, is added up by the value of parameter *itime* presented in Table 3.3. Upon the firing to transitions *Implant* and *Set Settings*, the model's time is increased consequently.

3.2.5 Dynamic Behaviour

Petri Nets and its extensions, such as CPNs and TCPNs, are typically executed via transitions' occurrence. In CPNs and TCPNs, each transition occurs if and only if it is enabled according to the following rules:

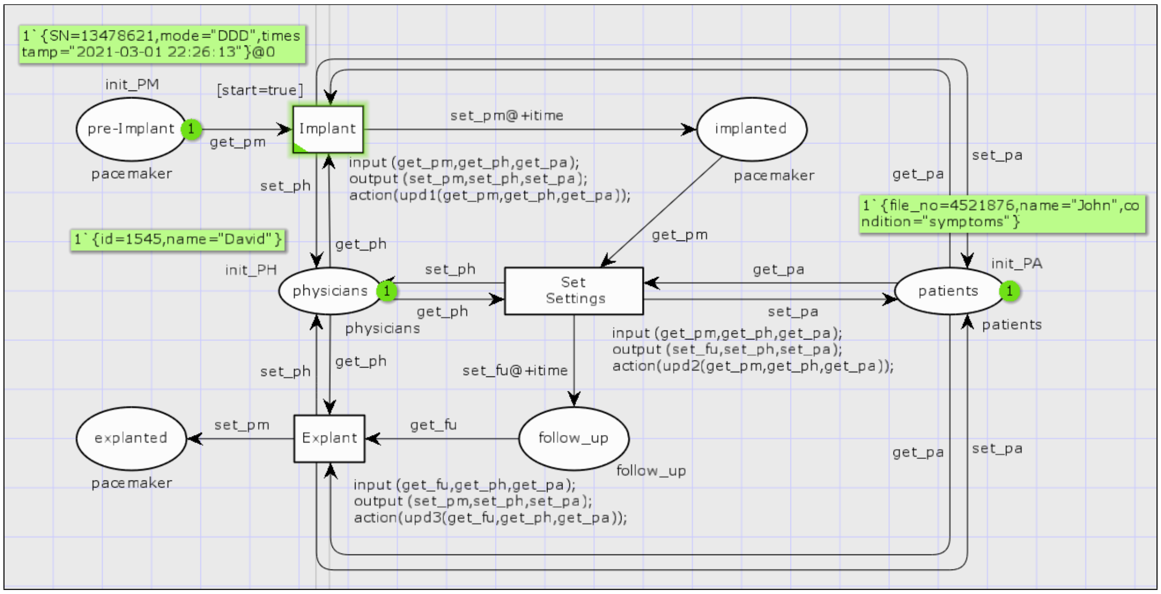
- There are sufficient tokens in the input places, and the tokens agree with the inscription of the input arcs.
- The Boolean expression of the transition's guard returns true.

In the example system presented in section 3.2.1, the input places for the transition *Implant* are *pre-implant*, *physicians*, and *patients*. Initially, each place has a sufficient number of tokens defined in Listing 3.6, which are also based on the colour sets defined in Table 3.2 and Listing 3.1. As well, each token agrees with the inscription of the connected input arcs defined in Listing 3.4. The Boolean expression of the transition, *Implant*, always returns true as long as the parameter's value, *start*, equals true, as shown in Listing 3.3. Subsequently, both rules of enabling the transition, Implement, are satisfied.

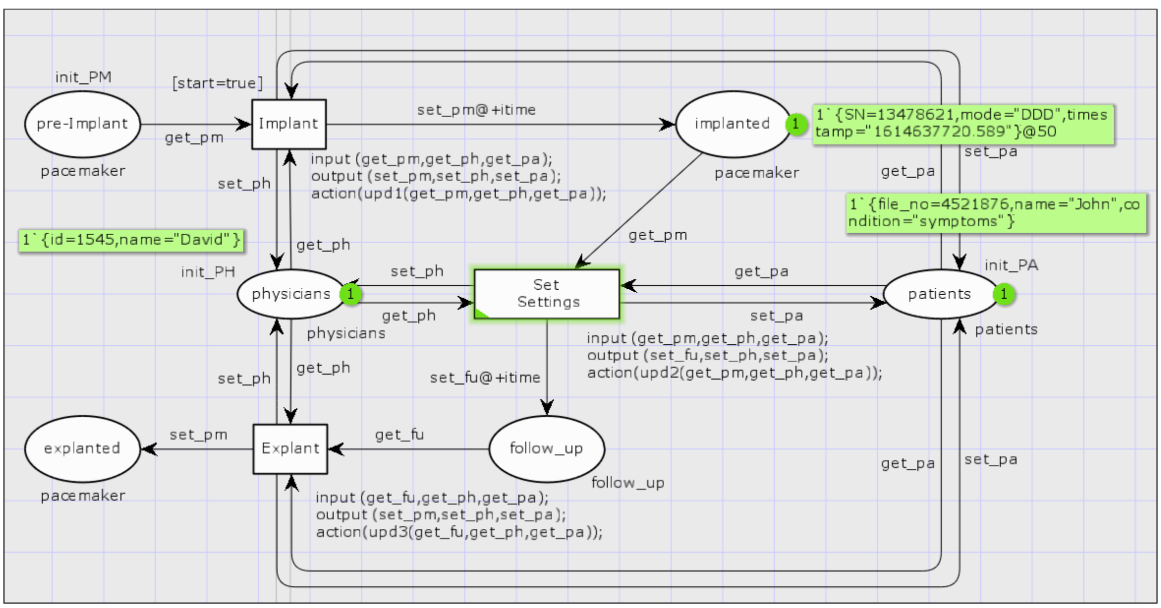
When a transition is enabled, binding elements can be recognized. Each input arc expression evaluates to one or more colours defined the corresponding input place. The binding elements of the example system are presented in Table 3.4 and Figure 3.4.

Table 3.4: The binding elements of the example system

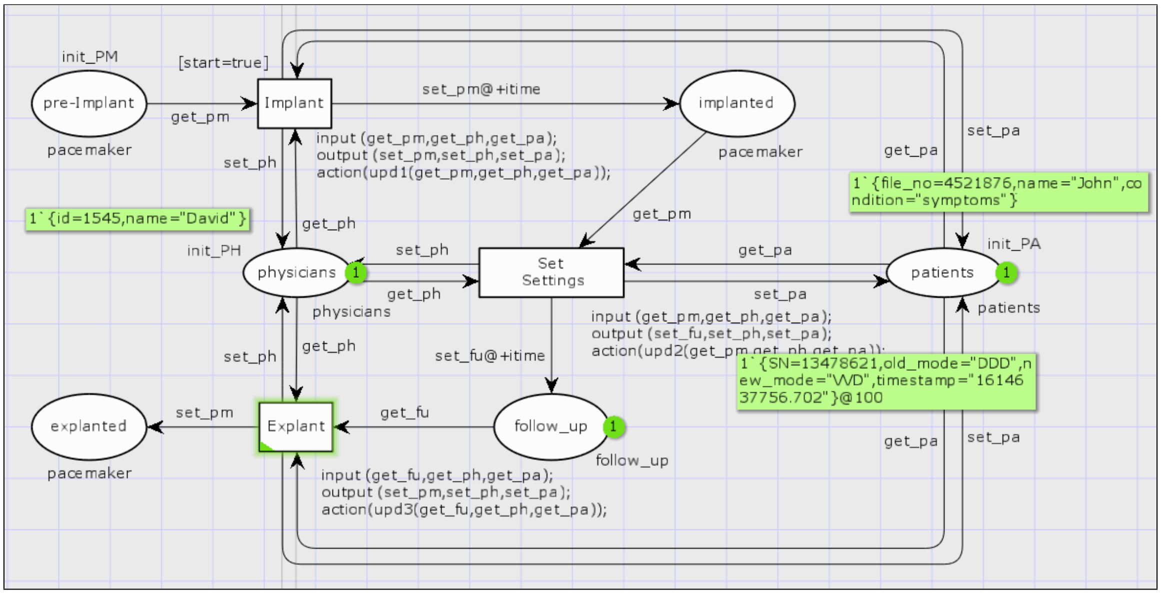
Step	Binding Element (Transition, Binding)
1	(Inplant, <get_ph={id=1545,name=”David”}, get_pa={file_no=4521876,name=”John”,condition=”symptoms”}, get_pm={SN=13478621,mode=”DDD”,timestamp=”2019-03-02 01:59:09”}@0}>)
2	(Set Settings, <get_ph={id=1545,name=”David”}, get_pa={file_no=4521876,name=”John”,condition=”symptoms”}, get_pm={SN=13478621,mode=”DDD”,timestamp=”2019-03-02 01:59:09”}@50}>)
3	(Explant, <get_ph={id=1545,name=”David”}, get_pa={file_no=4521876,name=”John”,condition=”symptoms”}, get_pm={SN=13478621,old_mode=”DDD”,new_mode=”VVD”, timestamp=”2019-03-02 01:59:09”}@100}>)



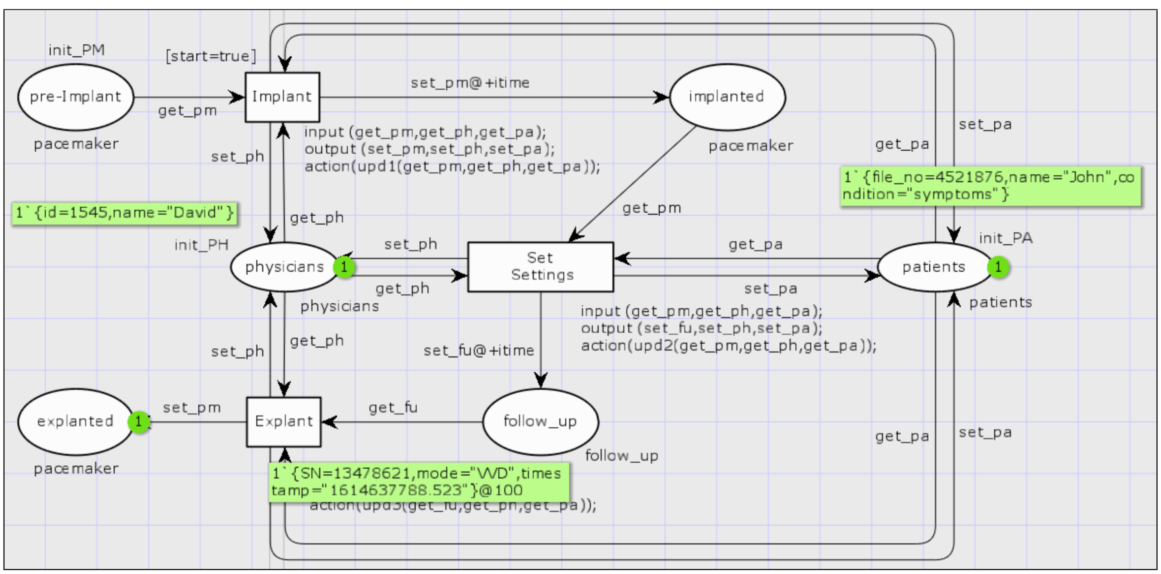
(a) Step 1: Initializing the model



(b) Step 2: After transition, **Implant**, was fired.



(c) Step 3: After transition, **Set Settings**, was fired.



(d) Step 4: After transition, **Explant**, was fired.

Figure 3.4: The dynamic behaviour of the example system

Even though the example system presents a determinism in its model, PNs also support non-determinism when required. Nevertheless, such a choice should always be subject to the specification of the proposed system. Enabled binding elements are either enabled concurrently when they can co-occur or enabled in conflict when enabled but cannot be enabled concurrently. Even though all binding elements in the example system are enabled concurrently, a possible demonstration for binding elements is by introducing another requirement. The new requirement states that if the pacemaker's battery is *Elective Replacement Time (ERT)*, then the new operating mode is set to *VOO* instead of *VVD*. Hence, two binding elements can possibly be enabled but not concurrently enabled.

3.2.6 Code Generation

Since the inception of compilers, code generation ordinarily utilizes high-level programming languages to produce machine code. Code generation can be articulated as generating a code from one form to another for specific purposes. In contrast, Model-to-Text transformations, as the name suggests, typically transfer pre-designed models into a high-level programming language.

Several studies and tools have been proposed to fully or partially carry Petri Nets (PNs) or Coloured Petri Nets (CPNs) toward an executable programing language. In [Simonsen [2014]], PetriCode, a tool for generating protocol software from CPNs, was presented. PetriCode uses a template-based method to generate code from CPN-based models annotated with code generation pragmatics. A two-way automatic translation process between specification and code was proposed in [Gomes and Barros [2001]], where Hierarchical Reactive Petri Nets were primarily utilized for the model role.

The work of [Carlsson [2018]] develops two tools to automatically generate code from CPNs specifications for game development with Unity3D. The first tool is a CPN editor for building CPNs models, while the other tool is an automatic code generator, as the name suggests, to generate code from CPNs. This approach has flaws and limitations, such as the incapability of dealing with concurrency. In [Mortensen [2000]], a practical method was described for the full automatic implementation of systems based on CPNs models. The simulation code of CPNs models is identical to the generated code of the final system implementation, which holds the same analysis results for the final executable system as for the CPNs models.

3.2.7 Computer Tools

There are a variety of tools that can be used for building and analyzing CPNs models (c.f. [Jensen et al. [2006], Westergaard [2013]]). In this thesis the **CPN Tools v.4.0.1** from [AIS Group [2013]] has been used. The CPN Group initially developed the CPN Tools at Aarhus University, Denmark, and since 2010 by the AIS group, Eindhoven University of Technology, The Netherlands.

However, even though CPN Tools provides a graphical execution of nets, it is not always convenient to visually observe the desired properties. Therefore, some tools were developed to improve the visual effects of a CPN model. For example, the PNV tool [Kindler and Páles [2004]] provides 3D visualization of PN-based models. Another tool, the BRITNeY Suite visualization tool [Westergaard [2006]] is compatible with CPN Tools of [AIS Group [2013]]. In addition, the fourth version of CPN tools [Westergaard [2013]] allows third parties to use standard Java-based applications directly.

In this thesis, the following components were used from the suite of CPN Tools; Editor, Simulator, and State Space Tool. The graphical Editor assists in constructing models. The Simulator executes the CPN ML language, which is based on the Standard ML. The simulation can be performed manually by selecting the enabled transition values, randomly with interactive feedback regarding the coloured tokens, or automatically where only the final marking is displayed. The State Space tool uses the executable CPN ML code to verify the models' properties via a set of functions. These functions can refer to specific parts of the state space, inquire about the state space, and report some state space's standard properties.

3.3 Formal Definitions

A non-hierarchical and a hierarchical Coloured Petri Nets are defined as follows [Jensen and Kristensen [2009]]:

- A Non-hierarchical Coloured Petri Net is a tuple:

$$CPN = (P, T, A, \Sigma, C, N, E, G, I)$$

  where:

  - P is a set of places.
  - T is a set of transitions.
  - A is a set of arcs. In CPN sets of places, transitions, and arcs are pairwise disjoint $P \cap T = P \cap A = T \cap A = \emptyset$

- Σ is a set of colour sets defined within CPN model. This set contains all possible colours, operations and functions used within CPN.
- C is a colour function that maps places in P into colours in Σ.
- N is a node function that maps A into $(P \times T) \cup (T \times P)$.
- E is an arc expression function that maps each arc $a \in A$ into the expression e. The input and output types of the arc expressions must correspond to the type of nodes that the arc is connected to.
- G is a guard function. It maps each transition $t \in T$ into guard expression g. The output of the guard expression must evaluate to a Boolean value of true or false.
- I is an initialization function. It maps each place p into an initialization expression i. The initialization expression must evaluate to multi-set of tokens with a colour corresponding to the colour of the place $C(p)$.

• A Hierarchical Coloured Petri Nets is a tuple:

$$CPN_M = (CPN, T_{sub}, P_{port}, PT)$$

where:

- $CPN = (P, T, A, \Sigma, C, N, E, G, I)$ is a non-hierarchical Coloured Petri Net.
- $T_{sub} \subseteq T$ is a set of substitution transitions.
- $P_{port} \subseteq P$ is a set of port places.
- $PT : P_{port} \rightarrow IN, OUT, I/O$ is a port type function. It maps a port types into port places.

- Timed Coloured Petri Nets (TCPN) using holding durations is defined as follows [Boukredera et al. [2012]]:

$$TCPN = (CPN, f, M_0)$$

where:

  - $CPN = (P, T, A, \Sigma, C, N, E, G, I)$ is a non-hierarchical Coloured Petri Net.
  - $f : T \rightarrow TS$ is a transition function which maps each transition $t \in T$ as a non-negative deterministic duration.
  - $M : P \rightarrow TS_{MS}$ is a timed marking in which the initial marking of $TCPN$ is denoted by M_0.

  In this thesis, $\mathbb{R}$ is used to denote Reals and $\mathbb{Z}$ to denote integers.

For more details and a thorough theory of CPN, the reader is referred to [Jensen and Kristensen [2009]].

3.4 Analysis Methods

Various techniques and tools can be applied to analyze CPN-based models [van der Aalst and Stahl [2011]]. In this work, two techniques were used, namely the simulation-based performance analysis and the reachability analysis.

3.4.1 Simulation-based Analysis

The simulation of a CPN-based model can be expressed as the process of firing a sequence of enabled trasitions. The simulation-based performance analysis is based on continually simulating the model while its data is being monitored, recorded, and then measured and compared. The primary advantages of the simulation-based analysis includes the following:

- To obtain more solid understanding and reliance in the correctness of the model's design. Simulating the model considerably assists in recognizing and debugging errors in the model's constructions and definitions.
- To achieve further comprehensive specification of the model as incomplete specifications will prohibit the model's execution as desired.
- To correct design flaws and examine particular sequences that reflect selected scenarios' behaviour in the system.
- To conduct a formal analysis of uncomplex models through executing all likely sequences of enabled binding elements, which is known as State Space analysis.

3.4.2 State Space Analysis

Although the simulation-based technique is flexible, it is also labour-intensive, and it cannot guarantee that all possible executions were covered. Therefore, State Space analysis (also called the reachability analysis) is conducted to verify certain functional and performance properties.

The State Space analysis is a formal verification technique typically performed by a computer tool to construct a directed graph of arcs and nodes (shown as rounded boxes). The nodes represent reachable markings (the model's states), and the arcs represent enabled binding elements (state transformations). In other words, in the reachability analysis or the Space State analysis, enabled binding elements are executed until all reachable markings are generated, giving all possible model states. Thus, various properties of the model can then be proven or even disproven.

Nevertheless, the calculation of timed state space can be complicated and laborious in part because the size of the reachability graph can be infinite as several timed markings with global clock and timestamps can be distinguished. In such a case, the model can then be analyzed for a defined period of time to limit the infinite markings due to the global clock.

The strongly-connected-component graph (SCC graph) is often derived from the State Space's graph structure. CPN State Space tool uses the SCC graph to assess the standard behavioural properties of the model. Figure 3.5 shows the directed graph of the full State Space analysis for the example system from section 3.2.1. This analysis results in four nodes (states) and three arcs (state transitions). The node's descriptor is displayed in a rectangular box to the right of each node. Each node in Figure 3.5 corresponds to the simulation steps presented in Figure 3.4.

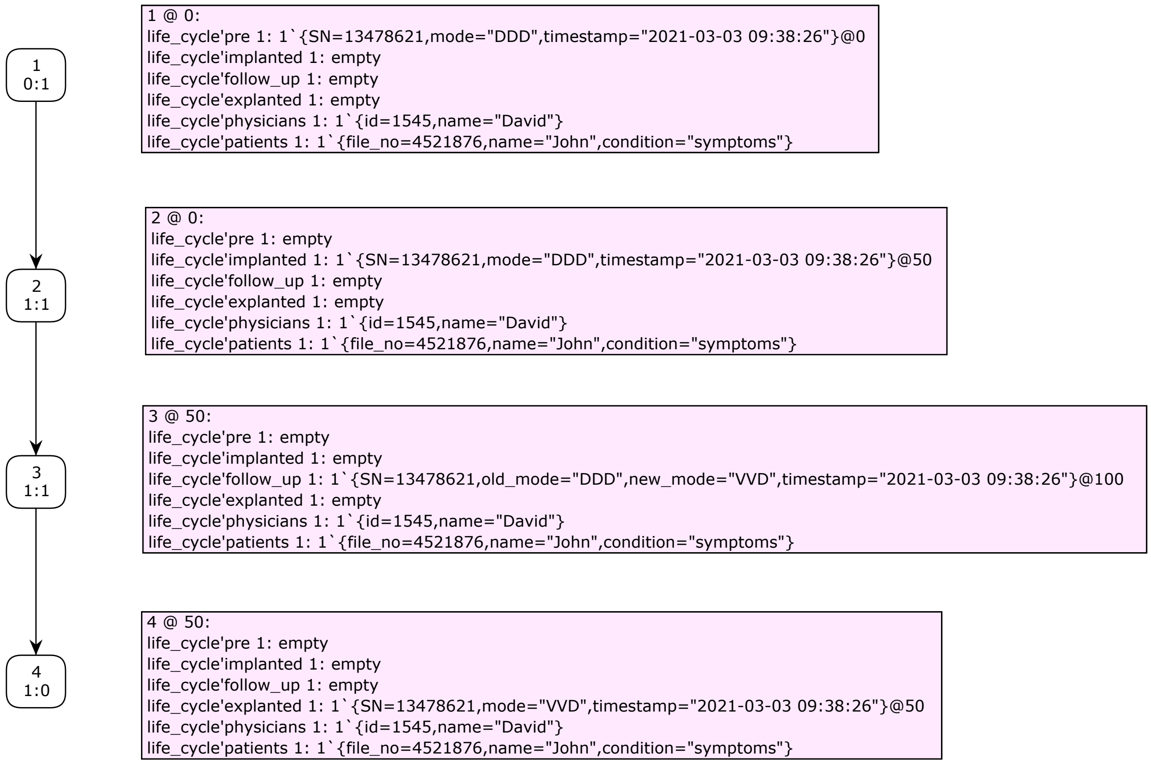


Figure 3.5: The directed graph of the example system

The behavioural properties reported by the Space State analysis include some statistical information as shown in Listing 3.10, boundedness properties as shown in Listing 3.11 and 3.12, and other properties as shown in Listing 3.13.

The statistics in Listing 3.10 show the size of both the State Space and SCC graph by reporting the number of nodes and arcs. The time consumed to complete this analysis is also reported in seconds with an indication of the analysis's status as either a full analysis where all possible states were covered or a partial analysis where only some states were covered, generally due to problems such as the state explosion problems.

In the Space State analysis of the example system, the reported statistical information in Listing 3.10 agrees with the directed graph in Figure 3.5 regarding the number of nodes and arcs. In addition, due to the trivial size of the example system, the time used to perform this analysis was less than one second. Since the CPN State Space tool records the time in seconds using integer numbers only, milliseconds are not reported. In the example system, it is understood that the analysis was completed in full in milliseconds.

Even though the number of nodes and arcs are equal for both State Space and SCC graph in the example system, these numbers may not always agree based on the model's specifications. When the number of SCC-graph nodes is fewer than State-Space nodes, it indicates that there are non-trivial SCCs and cycles in the State Space of the model (i.e. may not terminate).

```
Statistics
------------------------------------------------------------

  State Space
     Nodes:  4
     Arcs:   3
     Secs:   0
     Status: Full

  Scc Graph
     Nodes:  4
     Arcs:   3
     Secs:   0
```

Listing 3.10: Space State: Statistics

The reachability properties indicate whether an occurrence sequence exists between specific markings represented by nodes in the State Space. There are two methods to examine the reachability properties; checking the directed graph or performing standard query functions. Some instances for these functions include the following:

Reachable(n, n'), which returns a Boolean value based on the existence of a path in the State Space from node n to node n'.

SccReachable(n, n'), which also returns a Boolean value based on the existence of a path in the SCC graph from node n to node n'.

```
Boundedness Properties
--------------------------------------------------

 Best Integer Bounds
                             Upper      Lower
    life_cycle'explanted 1  1          0
    life_cycle'follow_up 1  1          0
    life_cycle'implanted 1  1          0
    life_cycle'patients 1   1          1
    life_cycle'physicians 1 1          1
    life_cycle'pre 1        1          0
```

Listing 3.11: Space State report: integer bounds

As shown in Listing 3.11 and 3.12, boundness properties report the number of tokens for places after executing all reachable markings. In *Best Integer Bounds*, the *upper* and *lower* columns respectively show the maximal and minimal numbers of tokens that can remain, in any reachable marking, on each place. Based on the example system specification, Listing 3.11 shows that both the places, physicians and patients, each always has a single token for both best upper and lower integer bounds indicating these places always hold a constant number of tokens. On the other hand, the other places have at most one token for some markings and also no token for other markings. All these reported numbers are expected according to the given specification of the system.

```
Best Upper Multi-set Bounds
    life_cycle'explanted 1 1`{SN=13478621,mode="VVD",
       timestamp="2021-03-03␣09:38:26"}
    life_cycle'follow_up 1 1`{SN=13478621,old_mode="DDD",
       new_mode="VVD",timestamp="2021-03-03␣09:38:26"}
    life_cycle'implanted 1 1`{SN=13478621,mode="DDD",
       timestamp="2021-03-03␣09:38:26"}
    life_cycle'patients 1  1`{file_no=4521876,name="John",
       condition="symptoms"}
    life_cycle'physicians 1  1`{id=1545,name="David"}
    life_cycle'pre 1 1`{SN=13478621,mode="DDD",timestamp="
       2021-03-03␣09:38:26"}

Best Lower Multi-set Bounds
    life_cycle'explanted 1  empty
    life_cycle'follow_up 1  empty
    life_cycle'implanted 1  empty
    life_cycle'patients 1  1`{file_no=4521876,name="John",
       condition="symptoms"}
    life_cycle'physicians 1  1`{id=1545,name="David"}
    life_cycle'pre 1    empty
```

Listing 3.12: Space State report: multi-set bounds

Listing 3.12 presents the *best upper and lower multiset bounds*, which state the number of tokens for each place with the assigned colour set in any reachable marking.

The *empty* multiset indicates the absence of tokens with the assigned colour set.

The home properties indicate if there is a home marking that is reachable from any other markings. The results of the home properties are beneficial in detecting specific errors, such as mismanaging resources. The example system, in 3.2.1, has no home marking, as shown in Listing 3.13.

```
 Home Properties
------------------------------------------------

  Home Markings
     Initial Marking is not a home marking
```

Listing 3.13: Space State report: home properties

The liveness properties address *dead markings*, *dead transition instances*, and *live transition instances*. *Dead markings* are those with unenabled binding elements. A single dead marking is typically expected with systems that should terminate at a certain point, enhancing the proposed model's correctness. *Dead markings* also result from nodes holding tokens but with no outgoing arcs or disabled transitions. Some standard functions can be performed to elaborate further details regarding the reported *dead markings*.

The assessment of whether such models with *dead markings* are expected or not depends heavily on the specifications of the desired systems. A single marking is reported in the example system as shown in Listing 3.14, which is expected due to the specification 3.2.1.

The results of *dead transition instances* indicate the transitions that never enabled (i.e. fired) during the model's execution. When the result of *dead transition instances* are reported as *None*, this means all the transitions in the model have the opportunity of firing (occurring) at least once. The example system has no *dead transition instances* as shown in Listing 3.14, which means all transitions occurred at least once, as expected and demonstrated in the directed graph in Figure 3.5.

```
 Liveness Properties
------------------------------------------------------

  Dead Markings
     [4]

  Dead Transition Instances
     None

  Live Transition Instances
     None
```

Listing 3.14: Space State report: liveness properties

On the other hand, the results of *live transition instances* state the transitions that are reachable in the occurrence sequence of any reachable marking. The existence of some *live transition instances* typically indicates some infinite occurrence sequences in the model. Models with dead markings regularly have no *live transition instances* since the model is terminated either as expected or not.

Dead transitions are not the opposite of live transitions since a non-dead transition must be enabled at least once while a live transition should continue to be enabled.

As reported in Listing 3.14, the example system has *None live transition instances*, which means each transition can not always be found in the occurrence sequence of any reachable marking.

```
 Fairness Properties
-------------------------------------------------------

     No infinite occurrence sequences.
```

Listing 3.15: Space State report: fairness properties

The results of the fairness properties represent a list of all impartial transitions, which occur infinitely. These results help identify transitions that, when are removed or restricted, all infinite occurrence sequences of the model will be subsequently eliminated if desired. As reported in Listing 3.15, the example system has no infinite occurrence sequences as expected in accordance with the system's specification.

For more details and a thorough theory of CPNs analysis, the reader is referred to [Jensen and Kristensen [2009] and van der Aalst and Stahl [2011]].

CHAPTER

4

THE CARDIAC CONDUCTION SYSTEM (CCS)

This chapter presents the modelling of the cardiac conduction system. Section 4.1 introduces the system. Section 4.2 demonstrates the proposed model. Finally, section 4.3 discusses the analysis results of the model.

4.1 Introduction

The heart is the environment where the pacemaker is dedicated to working. Modelling all functions of the heart is a laborious and time-consuming mission. Nevertheless, since one of the aims of this thesis is to formally present and produce realistic synthetic events of the heart's cardiac electrical activities, modelling the cardiac conduction system (CCS) appears to be a satisfactory method. The TCPN-base model of the cardiac conducting system assists not only in understanding the electrical activities but also in validating the various parameter of the pacemaker in a comparable realistic environment. In practice, the cardiac conduction system's activities are ordinarily observed, measured, and analyzed by means of an electrocardiogram (ECG or EKG). The ECG represents the recording of aggregate signals caused by cardiac cell population over time after appropriately placing a series of electrodes onto the human body's surface.

The accuracy of new biomedical signal processing algorithms is commonly analyzed with available real databases such as the PhysioNet database [Goldberger et al. [2000]]. Nevertheless, in different clinical settings with a range of noise levels and sampling frequencies, assessing the overall performance and validity can be challenging [McSharry et al. [2003]]. Further analysis with formal artificial ECG events may effectively and comprehensively improve the overall outcomes. The formal mathematical representation of ECG events must be inclusive and comprehensive to present a wide variety of rhythms, yet it must also be uncomplicated to facilitate different algorithms' formulation. In this chapter, a new methodology for constructing a graphical and mathematical model is described. The model precisely presents and produces a wide variety of time-based cardiac rhythms, which ultimately support the validation of the pacemaker system.

In this chapter, a formal model representing the cardiac conduction system is proposed. The model generates artificial yet realistic ECG events reflecting the activities of the cardiac conduction system. The model is based on Timed Coloured Petri Nets (TCPNs) and covers various known characteristics of cardiac rhythms in great detail. The generating of different sampling frequencies and noise levels of ECG events, which demonstrate the traditional human ECG event, can assist not only in assessing and optimizing signal processing techniques but also in comparing and contrasting these techniques for the most optimal outcomes. This model may also empower clinicians to form insightful decisions about electing ideal biomedical signal processing techniques in a specified application.

Unlike differential equation-based models or agent-based models, this proposed TCPN-based model does not require users to hold a strong knowledge of mathematics nor any programming language. The use of TCPNs, which combine the graphical notations, hierarchical structures, and supported types of data sets, empowers the representation of the proposed model to be more intuitive and user-friendly as well as formally defined and analyzed. Moreover, since biology is hierarchical, constructing the proposed model using TCPNs potentially supports multi-scale modelling [Chen and Hofestädt [2014]] through a proper extension to the structure of submodels and definition of colour sets.

4.2 TCPN-based Modelling of CCS

For flexible structure and clear presentation, the proposed model is composed of six interconnected submodels: the **event-structure submodel**, the **atrial-depolarization**

submodel, the **AV-node submodel**, the **ventricular-depolarization submodel**, the **ST-segment submodel**, and the **ventricular-repolarization submodel**. The functions and connection between these submodels, as shown in Figure 4.1, are described as follows: The **event-structure submodel** represents the common places or elements among other submodels. It represents an integral part of almost all other submodels that exchange data across the places of this submodel (i.e. the event-structure submodel). In a healthy heart, the cardiac cycle begins with the firing of the SA node (i.e. the natural pacemaker) followed by the depolarization of atrial musculature producing the recordable P-wave in the ECG. These activities are captured via the **atrial-depolarization submodel**. When the atria depolarization ends, action potentials spread through the AV node resulting in the PR segment in ECG. PR segment is the flat line between the end of the P wave and the start of the QRS complex. This event is addressed by the **AV-node submodel**. Then, the right and left ventricles start to depolarize and generate the recordable QRS complex, which is processed via the **ventricular-depolarization submodel**. The time interval between the depolarization and repolarization of the ventricular, called ST segment in ECG, is addressed via the **ST-segment submodel**. Eventually, the **ventricular-repolarization submodel**, as the name suggests, presents ventricular repolarization, which is the last stage of the cardiac cycle.

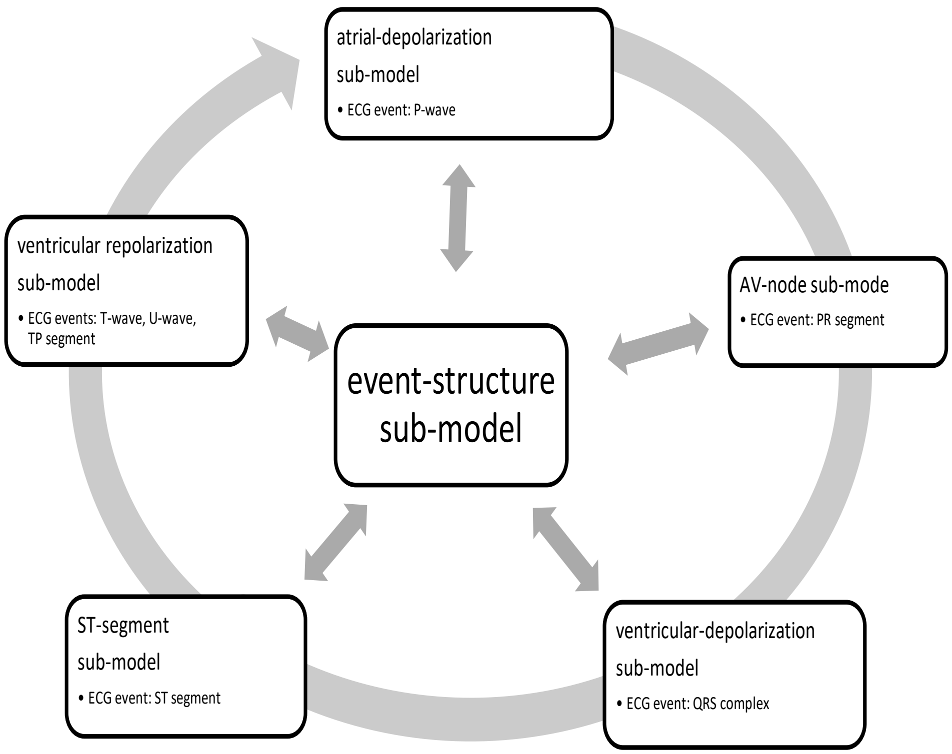


Figure 4.1: An overview of how the different submodels interact

4.2.1 Event-Structure Submodel

This submodel, as shown in Figure 4.2, consists of three places: $\langle$*Pacemaker*$\rangle$, $\langle$*Event*$\rangle$ and $\langle$*Interval and Ratio*$\rangle$. These also belong to other submodels. The main reason for modelling the event-structure as an independent submodel is that almost all other submodels contain these places of the event-structure submodel. Therefore, the places are defined in an independent submodel that has a recognizable name.

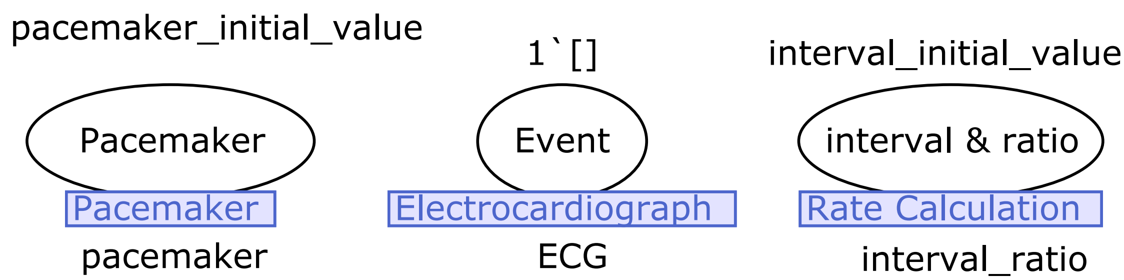


Figure 4.2: The Event-Structure Submodel

4.2.1.1 The Place: Pacemaker

The first place has the colour set **Pacemaker**, which is a record colour set or the Cartesian product of the sets as described in Table 4.1.

Table 4.1: The definitions of colour set Pacemaker

Colour Set	Definition
pacing source	*SA node* $\cup$ *AV node* $\cup$ *Purkinje Fibers* $\cup$ *None*
rate condition	*Fibrillation* $\cup$ *Flutter* $\cup$ *Tachycardia* $\cup$ *Normal* $\cup$ *Bradycardia* $\cup$ $\{NA\}$
pacemaker	$\{($*source, assigned bpm, rate condition, Pwave-based bpm, Pwave rate condition, Rwave-based bpm, Rwave rate condition, heartbeats counter*$)$ $\mid$ *source* $\in$ *pacing source, assigned bpm, Pwave-based bpm, Rwave-based bpm, heartbeats counter* $\mathbb{Z}_{\geqslant 0}$, *rate condition, Pwave rate condition, Rwave rate condition* $\in$ *rate condition* $\}$

```
colset pacing_source = with SA_node | AV_node |
   Purkinje_Fibers | None;
colset rate_condition_set = with Fibrillation | Flutter |
   Tachycardia | Normal | Bradyarrhythmia | NA;

colset pacemaker = record source:pacing_source * assigned_bpm:
   INT * rate_condition:rate_condition_set * Pwave_based_bpm:
   INT * Pwave_rate_condition:rate_condition_set *
   Rwave_based_bpm:INT * Rwave_rate_condition:
   rate_condition_set * heartbeats_counter:INT timed;
```

Listing 4.1: The CPN ML definitions of colour set Pacemaker

In a healthy heart, the Sinoatrial node (SA) is the natural pacemaker as it has the most rapid firing. Nevertheless, other cells have the capability of firing spontaneously at a slower rate as subsidiary pacemakers if the SA node permanently or temporarily ceases to fire. Atrioventricular node (VA), for instance, regularly fires at approximately 50 beats per minute (bpm), which is suppressed as long as the SA node's function is normal. Otherwise, the VA node can take over under conditions such as sinus arrest. The rate of the spontaneous depolarization of the cells is gradually slower in a downward trend. These behaviours reflect normal automaticity in the heart. In contrast, abnormal automaticity may occur under the conditions of ischemia or due to obscure reasons in pathological tachycardias [Curtis [2010]]. The fastest pacer not only determines the pace of the heart but also leads the rest of the other pacers after each beat. Consequently, in the colour set **Pacemaker**, $\langle$*source*$\rangle$ is determined based on the value of the parameter $\langle$*set bpm*$\rangle$ as defined in Table 4.8, and it is controlled by means of guard function on selected transitions (see individual submodels below for further details). The rhythmicity rates of some natural pacemakers [Aehlert [2001]] are presented in Table 4.2 below.

Table 4.2: The Rhythmicity Rate of Natural Pacemakers

Natural Pacemakers	beats per minute (bpm)
SA node	$100 - 60$
AV node	$60 - 40$
Purkinje fibers	$40 - 20$

Table 4.3: The conditions of the heart rate for adults

Rate Condition	Heart Rate (bpm)
Fibrillation	> 350
Flutter	$251 - 350$
Tachycardia	$101 - 250$
Normal	$60 - 100$
Bradycardia	< 60

The heart rate condition is defined based on the heartbeat rate per minute, and it is typically subject to gender and age group. For example, an adult's normal heart rate is between $60 - 100$ bpm, while the normal heart rate for a newborn baby is between $120 - 160$ bpm. In this proposed model in this chapter, the heart rate condition calculation is based by default on adults as demonstrated in Table 4.3. These rate conditions are translated into CPN ML as conditional calculation shown in Listing 4.2. However, the function in Listing 4.2 can be appropriately altered to fit any desired specifications for heart rate conditions. The ⟨*rate condition*⟩ in the colour set **Pacemaker** returned the calculated condition based the value of parameter ⟨*set bpm*⟩ with accordance the the conditional calculation in Listing 4.2.

```
fun rate_conditions(b:INT) =
if 350 < b then
    Fibrillation
else if 250 < b andalso b <= 350 then
    Flutter
else if 100 < b andalso b <= 250 then
    Tachycardia
else if 60 < b andalso b <= 100 then
    Normal
else if 20 < b andalso b <= 60 then
    Bradyarrhythmia
else
    NA
```

Listing 4.2: The CPN ML conditional calculations of heart rate

In the colour set **Pacemaker**, ⟨*heartbeat counter*⟩ tracks the number of generated full heartbeats. The importance of this parameter is to cause the model to terminate intentionally when the number of produced heartbeat is reached. Otherwise, the model will never stop since it is timed with the global clock (i.e. the model infinitely simulates cardiac activities unless some hardware limitations case an uncontrolled interruption).

On the other hand, since the heart rhythm is not ideally static but instead fluctuates based on various causes, such as physical activities, dynamic heart rate calculation is expected. In ECG, one method to calculate the current heart rate is to measure the atrial rate, indicated by *PP-interval*, or the ventricular rate, indicated by *RR-interval*. *PP-interval* and *RR-interval* represent the time between successive *P waves* and *R waves*, respectively. In the colour set **Pacemaker** ⟨*Pwave-based bpm*⟩ and ⟨*Rwave-based bpm*⟩ refer to the heart rate based on the calculation of PP-interval and RR-interval. The applied equation is as follows:

$$rate = \frac{60,000}{(\gamma - \delta)} \tag{4.1}$$

where:

- $60,000$ is one minute converted into a millisecond since ECG events are in milliseconds.
- γ is the value of the last interval of the desired wave.
- δ is the value of the previous interval of the same desired wave.
- Both γ and δ are calculated for each heartbeat and maintained their values by the sets ⟨*P previous interval*⟩, ⟨*P last interval*⟩, ⟨*R previous interval*⟩, and ⟨*R*

last interval$\rangle$ from the colour set **interval ratio** defined in Table 4.5.

4.2.1.2 The Place: ECG

The second place, as shown in Figure 4.2, has the colour set **ECG**. In principle, this is a list of generated synthetic information of $\langle$*ECG Event*$\rangle$. $\langle$*ECG Event*$\rangle$ is defined as a record colour set or the Cartesian product of the sets described in Table 4.4.

By default, each CCS event can be observed using ECG. Hence, when a CCS event occurs, a set of $\langle$*ECG Event*$\rangle$ is inserted into the list of the set $\langle$*ECG*$\rangle$. However, in some conditions, such as fibrillation or flutter, a number of similar fluctuated waves occur. These events also insert an equal number of sets of $\langle$*ECG Events*$\rangle$ once into the list of $\langle$*ECG*$\rangle$. Yet, these events are properly calculated in term of cumulative times based on the value of the parameters $\langle Fx \rangle$ and $\langle fx \rangle$ as demonstrated in Table 4.11.

To avoid possible overflow conflicts, upon generating a new CSS event, the content of the list of $\langle$*ECG*$\rangle$ is dropped, and then the new event is inserted. Nonetheless, the monitoring tool within CPN Tools was properly adopted and defined to allow all produced events to be optimally exported into text files, as discussed in a later section concerning the simulation-based analysis.

Table 4.4: The definition of colour set ECG

Colour Set	Definition
event	$\{$*P-wave, Q-wave, R-wave, S-wave, T-wave, U-wave, PR-segment, ST-segment, TU-segment, TP-segment, F-wave, f-wave, dropped-event*$\}$
bipolar limb leads	$\{(I,\ II,\ III\ ) \mid \textit{bipolar limb leads} \in \mathbb{R}\}$
augmented limb leads	$\{(aVR,\ aVL,\ aVF\ ) \mid \textit{augmented limb leads} \in \mathbb{R}\}$
precordial leads	$\{(V1,\ V2,\ V3,\ V4,\ V5,\ V6\ ) \mid \textit{precordial leads} \in \mathbb{R}\}$
leads (i.e. the standard 12 leads)	*bipolar limb leads* $\cup$ *augmented limb leads* $\cup$ *precordial leads*
ECG Event	$\{$(*ID, title, start time, end time, assigned duration, calculated duration, amplitude*) $\mid$ *ID* $\in \mathbb{Z}_{\geqslant 0}$, *title* $\in$ *event*, *start time, end time, assigned duration, calculated duration* $\in \mathbb{R}_{\geqslant 0}$, *amplitude* $\in$ *leads*$\}$
ECG	[*ECG Event*]

```
colset event = with P_wave | Q_wave | R_wave | S_wave | T_wave
    | U_wave | PR_segment | ST_segment | TU_segment |
   TP_segment | F_wave | f_wave | dropped_event;
colset leads = record I:REAL * II:REAL * III:REAL * aVR:REAL *
    aVL:REAL * aVF:REAL * V1:REAL * V2:REAL * V3:REAL * V4:
   REAL * V5:REAL * V6:REAL;
colset voltages = record defult:REAL * I:REAL * II:REAL * III:
   REAL * aVR:REAL * aVL:REAL * aVF:REAL * V1:REAL * V2:REAL *
    V3:REAL * V4:REAL * V5:REAL * V6:REAL;
colset ECG_Event = record id:INT * title:event * start_time:
   REAL * end_time:REAL * assigned_duration:REAL *
   calculated_duration:REAL * amplitude:leads;
colset ECG = list ECG_Event;
```

Listing 4.3: The CPN ML definitions of colour set ECG

As demonstrated in Table 4.4 and Listing 5.2, each token of ⟨*ECG Event*⟩ has a unique ⟨*ID*⟩ and ⟨*title*⟩ for the purpose of event identification. Each event shall start at a defined time for a specified duration and end eventually end. Accordingly, the ⟨*start time*⟩ marks the beginning of the event, which typically equals the ⟨*end time*⟩ of the previous event. However, the ⟨*start time*⟩ of the initial event equals the value of the parameter ⟨*initial time*⟩ introduced in Table 4.7.

Each event's duration can be either static or dynamic based on the defined values for the parameter ⟨*Event Duration*⟩. A static duration indicates that the event will always end after a static value of time for each heartbeat cycle. For example, an event constantly finishes after five milliseconds from its inception. On the other hand, a dynamic duration regularly returns a bounded arbitrary value of duration time for each cardiac rhythm, which is comparable to the heart's actual behaviour. Consequently, the $\langle$*end time*$\rangle$ of the $\langle$*ECG event*$\rangle$ is calculated as follows:

$$calculated\ duration \sim U(\omega - \kappa, \omega + \kappa) \tag{4.2}$$

$$end\ time = calculated\ duration + start\ time \tag{4.3}$$

where:

- U stands for a uniform distribution (aka a rectangular distribution), which has a constant probability.
- ω is the value of the duration of the parameter $\langle$*Event Duration*$\rangle$.
- κ is the value of the range of the parameter $\langle$*Event Duration*$\rangle$.

The implementation of these equations in CPN ML is presented in Listing 4.4.

```
fun calculate_time(duration:REAL * REAL) =
let
    val start_time= time()+0.0
    val dis = uniform(((#1 duration) - (#2 duration)),((#1
       duration) + (#2 duration)))
    val end_time= start_time+dis
    val time_diff = (end_time-start_time)
in
   (start_time,end_time,time_diff)
end
```

Listing 4.4: The CPN ML implementation for the calculation of the duration

The above function, *uniform*, is one of several predefined functions for random distribution. The reason for electing *uniform* over other functions is that this function returns a drawing from a continuous uniform distribution between two real numbers and these two real numbers shall be in order and unequal. Therefore, in ⟨*ECG Event*⟩, the ⟨*calculated duration*⟩ and consequently the ⟨*end time*⟩ can either be variable for each event in each single heartbeat, or they can be constant (w.r.t. ⟨*start time*⟩) if the value of ⟨*range*⟩ in the parameter ⟨*Event duration*⟩ is set to zero (see Table 4.9).

For example, if an event duration was declared as $(10.0, 5.0)$, then for each cardiac rhythm, the event will be assigned a value between 5.0 and 15.0 randomly. The previous example can be declared as $(10.0, 0.0)$ for the constant duration, which means the event duration will always be 10.0. In contrast, in ⟨*ECG Event*⟩, ⟨*assigned duration*⟩ presents the **mean** value of the declared event duration in the ⟨*Event Duration*⟩.

The ⟨*calculated duration*⟩ of all events is calculated as explained above. However, the event between cardiac rhythm cycles is represented as *TP segment* in ECG, and its calculated duration is calculated slightly different. In ECG, the *TP segment* starts from the last wave of the ventricular repolarization (i.e. *T wave* or *U wave*) till the next *P wave* (i.e. atrial depolarization). In this proposed model, the ⟨*calculated duration*⟩ of the *TP segment* can be either declared manually or calculated approximately. The formal declaration is triggered when a value greater than zero is assigned to the duration of the parameter ⟨*TPd*⟩ (see Table 4.9). Otherwise, the estimated calculation is triggered when the duration of the parameter ⟨*TPd*⟩ equals zero. Hence, the computed value for⟨*TPd*⟩ is based on the following equations:

$$TPd = \frac{60,000 - \beta}{\langle set\ bpm \rangle} \tag{4.4}$$

where:

- $60,000$ is one minute converted into a millisecond since ECG events are in milliseconds.
- β is the total cumulative durations of all enabled events in accordance to the value of parameters ⟨*Event Enablement*⟩ defined in Table 4.12 (i.e. it represents the sum of only enabled events).
- ⟨*set bpm*⟩ is a parameter that stored the assigned number of beats per minute explained in Table 4.8.

Nonetheless, it is possible for the above equation to return a value that is below zero. This issue is typically due to declare abnormal durations for some or all events.

The model, however, is trained to correct and adjust the durations of all events when the returns value of the above equation is slightly below zero. Otherwise, extreme abnormal values break this model because of limited flexibility with the CPN Tools. In case the above equation returns a value that is slightly below zero, then a dynamic adjustment is applied automatically on each enabled event as follows:

$$adjusted\ TPd = |TPd| \tag{4.5}$$

$$adjusted\ duration = \Big(\frac{\rho - adjusted\ TPd}{\sigma}\Big) - \psi) \tag{4.6}$$

where:

- *adjusted* TPd is the absolute value of the duration of $\langle TPd \rangle$.
- *adjusted duration* is the equation that is applied on all events excluding $\langle TPd \rangle$
- ρ is the original assigned duration of the calculated event.
- σ is the number of enabled events other than *TP segment*.
- ψ is the original assigned range of the calculated event.

$\therefore$ The adjusted duration of $\langle TPd \rangle$ is the absolute value of the originally calculated duration from 4.4.

The ECG can describe CCS events as recording accumulative signals produced by the cardiac cells over time. The measurements of those events' amplitudes commonly identify the characteristics of various events. In the standard ECG, twelve leads, which require accurate configuration, report the detected cardiac electrical activities from twelve different perspectives. These 12 leads are as follows:

- Three bipolar limb leads ⟨*I, II, III*⟩
- Three augmented limb leads ⟨*aVR, aVL, aVF*⟩
- Six precordial leads (aka chest leads) ⟨*V1, V2, V3, V4, V5, V6*⟩

In ⟨*ECG Event*⟩, the ⟨*amplitude*⟩ is of the colour set **leads** (defined in Table 4.4), which represents the total height of events via the values of 12 standard ECG leads. The value of ⟨*amplitude*⟩ is supplied by the parameter ⟨*Event Amplitude*⟩ defined in Table 4.10.

While the 12 leads are the ECG standard, lead ⟨ II ⟩ is typically accepted as the primary representation of detected cardiac activities. Therefore, the proposed model supports two methods for setting the values of the parameter ⟨*Event Amplitude*⟩. The formal and reliable method is to manually declare each lead's value in accordance with required specifications.

The other method is suggested to reduce the complexity of declaring the values of all limb leads (i.e. bipolar and augmented limb leads). Hence, this proposed model is equipped with an algorithm that estimates other limb leads' values by getting only one declared limb leads (i.e. ⟨ II ⟩). This algorithm is designed for only returning expected values that might not always be reliable estimations. This algorithm applies some related laws, such as Einthoven's law, and it utilizes several pre-defined parameters. The determination of using the algorithm should be ruled with the desired results.

Furthermore, when developing the model, another made decision was to force the static declaration for all the leads' values in order to decrease the overall complexity when declaring the leads' values. This judgment implies that all leads' values will retain the same values for all cardiac rhythms. While this is not a realistic condition, the model is still capable of supporting the bounded and controlled arbitrary assignments in a fashion that is quite similar to the above implementation with the event's duration.

To trigger the algorithm, in the parameter ⟨*Event Amplitude*, the value of ⟨*default* shall set to a non-zero real number. Otherwise, each component of ⟨*amplitude*⟩ shall be assigned according to the desired value. The algorithm, when applies, calculates the values of limb leads, which include bipolar limb leads (*i.e.* ⟨ *I, II, III* ⟩) and augmented limb leads (*i.e* ⟨ *aVR, aVL, aVF* ⟩). The first objective is to measure the potential difference between two points [Novosel et al. [1999]]. The second objective is to provide estimated values for limb leads based on the value of *default* in the parameter ⟨*Event Amplitude*⟩ along with the axes of ⟨*Electrical Vectors*⟩ and ⟨*Bipolar Limb Leads Axes*⟩ (defined in Table 4.15).

In order to properly explain the algorithm, each step is discussed profoundly as follows:

The first step of the algorithm is to calculate the value of one lead according to the values of the other two leads. This step is based on Einthoven's Law, which is as follows:

$$Lead\ II = Lead\ I + Lead\ III \tag{4.7}$$

Consequently, substitute ⟨*lead II*⟩ by ⟨*default*⟩, the values of the other two leads can be presented as a linear equation in two variables as follows:

$$default = x + y \tag{4.8}$$

where:

- $\langle default \rangle$ represents the value of $\langle$*lead II*$\rangle$.
- x represents the value of $\langle$*lead I*$\rangle$.
- y represents the value of $\langle$*lead III*$\rangle$.

The second step of the algorithm is to use the values of the axes of $\langle$*Electrical Vectors*$\rangle$ and $\langle$*Bipolar Limb Leads Axes*$\rangle$ to guess informed estimations for the values of the above $\langle x \rangle$ and $\langle y \rangle$. Consequently, two possible cases can be encountered.

The first case is that the selected examined $\langle$*Electrical Vector*$\rangle$ is perpendicular to one of the $\langle$*Bipolar Limb Leads Axes*$\rangle$. Therefore, the estimated values when the selected $\langle$*Electrical Vectors*$\rangle$ is perpendicular to a lead are as follow (where $\nu_1 \perp \nu_2$ means ν_1 is perpendicular to ν_2) :

$$\nu \perp leadII \Rightarrow (\chi \times 0) = (+\chi) + (-\chi) \tag{4.9}$$

$$\nu \perp leadI \Rightarrow \chi = (\chi \times 0) + \chi \tag{4.10}$$

$$\nu \perp leadIII \Rightarrow -\chi = (-\chi) + (\chi \times 0) \tag{4.11}$$

where:

- ν is the selected examined $\langle$*Electrical Vector*$\rangle$.
- χ is the assigned value of the parameter $\langle$*default*$\rangle$.
- The left side of the equation points to the value of $\langle$*lead II*$\rangle$.
- The right side of the equation points to the value of $\langle$*lead I*$\rangle$ and $\langle$*lead III*$\rangle$, respectively.

- Equation 4.9 indicates that the selected ⟨*Electrical Vectors*⟩ is perpendicular to ⟨*lead II*⟩
- Equation 4.10 indicates that the selected ⟨*Electrical Vectors*⟩ is perpendicular to ⟨*lead I*⟩
- Equation 4.11 indicates that the selected ⟨*Electrical Vectors*⟩ is perpendicular to ⟨*lead III*⟩

In the above equations, signs are changed oppositely (i.e. positive signs become negative and vice versa) based on the direction and position of the ⟨*Electrical Vectors*⟩.

For example, in 4.9, the ⟨*Electrical Vectors*⟩ is perpendicular with one of the ⟨*Bipolar Limb Leads Axes*⟩, but it moves towards the opposite direction. Hence the equation of *lead II* becomes as follows:

$$-\nu \perp leadII \Rightarrow (\chi \times 0) = (-\chi) + (+\chi) \tag{4.12}$$

The second case is when the above cases are inapplicable (i.e. all other cases). That means the selected examined *Electrical Vector* is not perpendicular to any of the ⟨*Bipolar Limb Leads Axes*⟩. Therefore, further steps in the algorithm are executed to measure the distance between the ⟨*Electrical Vector*⟩ and the *perpendicular vector* of the desired lead. Then based on the deference, a coefficient number is identified and used to estimate the values of the two other leads.

The perpendicular vector of a lead, based on the triaxial reference system, can be identified as follows:

$$\begin{aligned} &if\ \rho > 90\ then\ \mu = -((180 - \rho) + 90) \\ &else\ if\ \rho < -90\ then\ \mu = \rho - (-90) \\ &else\ \mu = \rho + 90 \end{aligned} \tag{4.13}$$

$$\begin{aligned} &if\ \mu > 0\ then\ \mu' = -(180 - \mu) \\ &else\ \mu' = \mid 180 - (-\mu) \mid \end{aligned} \tag{4.14}$$

where:

- ρ is the value of the lead's axis.
- μ is the perpendicular vector for a lead moving clockwise.
- μ' is the perpendicular vector moving counter-clockwise.

$\therefore$ since the axis of $\langle$*lead II*$\rangle$ is 60°, its perpendicular vectors are 150° moving clockwise and −30° moving counter-side.

Next, the difference between the $\langle$*electrical vector*$\rangle$ and the lead's perpendicular vector is calculated as follows:

$$\chi = \mid \upsilon \mid + \mid \rho \mid \tag{4.15}$$

$$\chi' = ((180 - \mid \upsilon \mid) + 180) - \mid \rho \mid \tag{4.16}$$

$$if\ (\upsilon\ \wedge\ \rho > 0)\ \vee (\upsilon\ \wedge\ \rho < 0)\ then\ \delta =\mid \upsilon - \rho \mid \tag{4.17}$$

$$if\ \chi' > \chi\ then\ \delta = \chi\ else\ \delta = \chi' \tag{4.18}$$

where:

- χ is the distance via clockwise direction between two vectors.
- χ' is the distance via counter-clockwise direction.
- υ is the value of the $\langle$*Electrical Vector*$\rangle$.
- ρ is the value of the lead's *perpendicular vector*.
- δ is the calculated deference.

Next, a coefficient number is to be identified and used to assist in calculating projected values of $\langle$*lead I*$\rangle$ and $\langle$*lead III*$\rangle$ based on the value of the parameter $\langle$*default*$\rangle$, which represents $\langle$*lead II*$\rangle$. One possible way to find a coefficient number is by the value of the above calculated difference between the two vectors to be tested against the following conditions:

$$\begin{gathered}
if\ 0 \leqslant \delta \leqslant 10\ then\ \alpha = 1 \\
else\ if\ 11 \leqslant \delta \leqslant 30\ then\ \alpha = 4 \\
else\ if\ 31 \leqslant \delta \leqslant 70\ then\ \alpha = 3 \\
else\ if\ 71 \leqslant \delta \leqslant 110\ then\ \alpha = 2 \\
else\ if\ 111 \leqslant \delta \leqslant 150\ then\ \alpha = 3 \\
else\ if\ 151 \leqslant \delta \leqslant 170\ then\ \alpha = 4 \\
else\ \alpha = 0
\end{gathered} \tag{4.19}$$

where:

- δ is the calculated difference from the previous step.
- α is the value of the coefficient number.

Next, the estimated values for the leads are as follows:

$$\chi = \Big((\frac{\chi}{\alpha}) \times (\alpha - 1) \Big) + \Big(\frac{\chi}{\alpha} \Big) \tag{4.20}$$

where:

- χ is the assigned value of the parameter $\langle$*default*$\rangle$.
- α is the value of the coefficient number.
- The left side of the equation points to the value of $\langle$*lead II*$\rangle$.
- The right side of the equation points to the value of $\langle$*lead I*$\rangle$ and $\langle$*lead III*$\rangle$, respectively.

Finally, the signs of the projected values are determined by the position of the *Electrical Vector* in accordance with the Hexaxial reference system.

The third step of the algorithm is deriving the values of the augmented limb leads from the preceding values of bipolar limb leads. As demonstrated in Novosel et al. [1999], the used equations here are as follows:

$$aVL = \beta 1 - (\beta 3 \div 2) \tag{4.21}$$

$$aVR = -\beta 1 + (\beta 2 \div 2) \tag{4.22}$$

$$aVF = \beta 2 + (\beta 3 \div 2) \tag{4.23}$$

where:

- $\beta 1$ is the value of $\langle$*lead I*$\rangle$.
- $\beta 2$ is the value of $\langle$*lead II*$\rangle$.
- $\beta 3$ is the value of $\langle$*lead III*$\rangle$.

Similar to the limb leads, the values of the precordial leads can be constant via the parameter $\langle$*Event Amplitude*$\rangle$ or variable that are calculated in consonance with the values of the parameter $\langle$*Precordial Lead Variables*$\rangle$ defined in Table 4.18 as follows:

$$wct = \frac{(RR + LA + LL)}{3} \tag{4.24}$$

$$\nu_i = Viv - wct \tag{4.25}$$

where:

- wct is the value of a common reference point known as Wilson's central terminal
- ν_i is the value of the selected lead of the ⟨*Precordial Leads*⟩.
- Viv is the leads' variables from the parameter ⟨*Precordial Lead Variables*⟩.
- i is the lead's number from 1 to 6.

The connection of leads is presented by the parameters ⟨*Bipolar Limb Leads*⟩, ⟨*Augmented Limb Leads*⟩, and ⟨*Precordial Leads*⟩ as defined in Table 4.17. If the parameter of a corresponding lead is set to ⟨*false*⟩, then the amplitude of that lead is zero, which theoretically states an inactive lead. Otherwise, the amplitude of the lead is calculated as previously discussed.

4.2.1.3 The Place: interval & ratio

The third presented place in Figure 4.2 is ⟨*interval & ratio*⟩, and it has the colour set **interval ratio** defined in Table 4.5 and Listing 4.5. In principle, it is a set of numerical trackers that assist in calculating the intervals and ratios of the produced ECG events.

Table 4.5: The definition of colour set "interval and ratio"

Colour Set	Definition
interval ratio	$\{($*P previous interval, P last interval, R previous interval, R last interval, heartbeat counter, heartbeat ratio, P ratio, Q ratio, R ratio, S ratio, T ratio, U ratio, PR ratio, ST ratio, TU ratio*$)$ $\mid$ *interval ratio* $\in \mathbb{Z}_{\geqslant 0}$ $\}$

```
colset interval_ratio = record P_previous_interval:INT *
   P_last_interval:INT * R_previous_interval:INT *
   R_last_interval:INT * heartbeat_counter:INT *
   heartbeat_ratio:INT * P_ratio:INT * Q_ratio:INT * R_ratio:
   INT * S_ratio:INT * T_ratio:INT * U_ratio:INT * PR_ratio:
   INT * T_ratio:INT * TU_ratio:INT;
```

Listing 4.5: The CPN ML definitions of colour set interval & ratio

In the colour set **interval ratio**, the ⟨*P previous interval*⟩ and ⟨*P last interval*⟩ represent the *PP-interval*. Similarly, ⟨*R previous interval*⟩ and ⟨*R last interval*⟩ represent the *RR-interval*. Both *PP-interval* and *RR-interval* are parts of the heart rate calculation previously explained in colour set **Pacemaker** as in Equation 4.1. The ⟨*heartbeat counter*⟩, as the name suggests, a counter that maintains tracking of the number of produced cardiac rhythms. This tracker is crucial to perform other calculations and functions, such as terminating the model when the desired number of generated heartbeats is reached.

The ratio trackers, in the colour set **interval ratio**, power the model to have a more realistic representation. Various cardiac arrhythmia can be classified according to their characteristics, and ratios are essential for such classification. In mathematics, a ratio is a quantitative relation between two amounts that show the number of times where one value is contained or contains within the other value.

Consequently, the event ratio can be expressed as a pair as follows:

$$(\eta : \mu) \tag{4.26}$$

where:

- η is the frequency of dropping the event (w.r.t. $\langle$*Event Duration*$\rangle$)
- μ is the total number of occurrences of the event before the event is dropped.

The parameters of the event ratio determine the times of appearance of the corresponding event. Two parameters rule the event ratio, namely $\langle$*Event Enablement*$\rangle$ and $\langle$*Event Ratio*$\rangle$ as defined in Table 4.13. The main distinction between these two parameters is their consideration of the event's duration time. The behaviour of the event ratios' parameters is as follows:

- If the $\langle$*Event Enablement*$\rangle$ is set to $\langle$*false*$\rangle$, then the event will disappear from the ECG events (with zero duration time) regardless of the value of the parameter $\langle$*Event Ratio*$\rangle$.
- However, if the $\langle$*Event Enablement*$\rangle$ is set to $\langle$*true*$\rangle$, then whenever the $\langle$*Event Ratio*$\rangle$ is fulfilled, the event is substituted by the set of $\langle$*dropped-event*$\rangle$, which holds the same event's assigned duration but whit zero amplitudes.

The ratio relationship for the $\langle P\ wave \rangle$ event can be express according to the pair in 4.26 as follows:

$$(\langle Pr \rangle : \langle P\ ratio \rangle) \tag{4.27}$$

where:

- $\langle Pr \rangle$ is a parameter defined in Table 4.13. It identified the number of cardiac rhythm cycles before dropping the $\langle P\ wave \rangle$ event.
- $\langle P\ ratio \rangle$ is in the colour set **interval ratio**. It increases by *one* with every detectable $\langle P\ wave \rangle$ within a heartbeat. Also, it reset to *zero* whenever the $\langle P\ wave \rangle$ is dropped.
- The $\langle P\ wave \rangle$ is dropped when $\langle Pr \rangle = \langle P\ ratio \rangle$
- All other events' ratios are behaving in a similar way.

However, while the previous parameters (i.e. $\langle Event\ Enablement \rangle$ and $\langle Event\ Ratio \rangle$) control the ratio of every single event independently, the parameters $\langle Dropped\ Beat\ Enablement \rangle$, $\langle Dropped\ Beat\ Ratio \rangle$ and $\langle Dropped\ Beat\ Duration \rangle$ control the entire cardiac rhythm for each cardiac conduction cycle, which includ all events within the same rhythm.

The parameters regulating the cardiac rhythms ratio and the other parameters regulating the events ratios do not behave similarly (i.e. they deliver different concepts). For instance, while $\langle Event\ Enablement \rangle$ substitutes an event with a zero duration time (i.e. the event is not produced), the $\langle Dropped\ Beat\ Enablement \rangle$ behaves as follows:

- If the ⟨*Dropped Beat Enablement*⟩ is set to ⟨*true*⟩, the ⟨*Dropped Beat Duration*⟩ is set to a value *bigger than zero*, and the ⟨*Dropped Beat Ratio*⟩ is fulfilled, then the duration of the dropped beat will as long or as short as the value of the parameter ⟨*Dropped Beat Duration*⟩.
- If the ⟨*Dropped Beat Enablement*⟩ is set to ⟨*true*⟩, the ⟨*Dropped Beat Duration*⟩ is set to *zero*, and the ⟨Dropped Beat Ratio⟩ is fulfilled, then the duration of the dropped beat will be equal to the value of one beat duration as defined in Equation 4.1.
- If the ⟨*Dropped Beat Enablement*⟩ is set to ⟨*false*⟩, then no beat is dropped regardless of the values of the other related parameters.

4.2.1.4 The model variables and parameters

Typically, variables are not the same as parameters. In a single simulation, variables are used to denote states of the model, and their values may be replaced. In contrast, parameters are utilized to represent objects statically, and their values are ordinarily constant unless updated by the user to observe a different behaviour of the model. In this model, CPN variables are used in CP-net inscriptions, and these variables are bound to various values from their colour set. Table 4.6 shows the CPN variables of this model and their colour set.

Table 4.6: The CPN variables of the colour sets

Colour Set	CPN variable(s)
REAL	*duration*
interval ratio	*get_interval, set_interval*
ECG	*get_ecg, set_ecg*
pacemaker	*pacemaker*

Modelling the CCS events involves conducting their best simulation based on the selected values of the parameters. More accurate values of the parameters lead to more reliable results generated by the model. In this chapter, the proposed model provides parameters utilized for mechanical identification. In addition, these parameters assist in generating various sinus rhythms through following significant concepts. These concepts characterize rhythms and therefore assist in identifying arrhythmias (i.e. irregular heartbeats). Some concepts include the speed and regularity of the rhythm, the presence of certain waves, and the length of certain intervals [Garcia [2014]]. The adopted parameters are classified here according to their purposes, and they are as follows:

Electrocardiogram Setup

Table 4.7 shows three parameters that control the simulated behaviour of the ECG machine. Typically, the ECG recording is run for a finite time, represented by the parameter ⟨*number_of_heartbeats*⟩. The parameter ⟨*initial_time*⟩ denotes the behaviour of initiating the ECG recording after a particular time. The last parameter in this group is ⟨*paper_speed*⟩. Even though the ECG paper speed is regularly (25 mm/sec), it may be increased to (50 mm/sec) to define waveforms better.

Table 4.7: Parameters for Electrocardiogram Setup

Parameter	Valid Value	Annotations
number_of_heartbeats	$\mathbb{Z}_{\geqslant 0}$	The maximum number of simulated heartbeats. If the input value equals zero, then the number of simulated heartbeats will be unrestricted.
initial_time	$\mathbb{R}_{\geqslant 0}$	The initial start time of the model
paper_speed	$\mathbb{Z}^{+}$	mm/sec

Heartbeat and Rhythmicity Rates

As shown in Table 4.8, the first parameter defines the heartbeat rate as the number of beats per one minute. Some equations depend on this parameter, such as Equation 4.4, to calculate some events' duration. The rest of the parameters regarding the rhythmicity rates provide control and definition for natural pacemakers (Table 4.2). These parameters allow additional investigation toward wide varieties of pacemakers' behaviours.

Table 4.8: Parameters for Heartbeat and Rhythmicity Rates

Parameter	Valid Value	Annotations
set_bpm	$\mathbb{Z}_{\geqslant 0}$	beats/min
SA_node_rate	$(max,\ min) \mid max,\ min \in \mathbb{Z}_{\geqslant 0} \wedge max \geqslant min$	For the pacemakers defined in Table 4.2
AV_node_rate		
purkinje_fibers_rate		

Events Durations

As explained in equation 4.2, each event has a defined value for its duration, and the duration for each event can be either constant or dynamic (i.e. bounded arbitrary value for each cardiac cycle). Table 4.9 shows the wave duration parameters and the segment duration parameters. The column ECG sequence refers to the order in which the event appears in a normal ECG.

Table 4.9: Parameters for Segments Durations

Parameter	Valid Value	Annotations	ECG Sequence
Pd	$(duration,\ range) \mid duration,\ range \in \mathbb{R}_{\geqslant 0}$	P wave	1
Qd		Q wave	3
Rd		R wave	4
Sd		S wave	5
Td		T wave	7
Ud		U wave	9
PRd		PR segment	2
STd		ST segment	6
TUd		TU segment	8
TPd		TP segment (Equation 4.4)	10
Fd		F wave(s)	NA
fd		f wave(s)	NA

Waves Amplitudes

Table 4.10 represents the wave amplitude parameters. In ECG, the amplitude (or voltage) is displayed in a vertical dimension over time. Typically in standard ECG, waves can be distinguished by their amplitudes measured in volts (V). As shown in Table 4.4, each event can be reported from 12 different perspectives through 12 leads. However, all segments ordinarily tend to hold zero amplitude (i.e. no electrical activities) and therefore, the amplitudes for all segments are denoted by a single parameter since there are no distinct values.

Table 4.10: Parameters for Waves Amplitude

Parameter	Valid Value	Annotations
Pa	{defult= $\mathbb{R}$, I= $\mathbb{R}$, II= $\mathbb{R}$, III= $\mathbb{R}$, aVR= $\mathbb{R}$, aVL= $\mathbb{R}$, aVF= $\mathbb{R}$, V1= $\mathbb{R}$, V2= $\mathbb{R}$, V3= $\mathbb{R}$, V4=$\mathbb{R}$, V5= $\mathbb{R}$, V6= $\mathbb{R}$ }	P wave
Qa		Q wave
Ra		R wave
Sa		S wave
Ta		T wave
Ua		U wave
Fa		F wave
fa		f wave
Seg		All segments

Frequency of Fluctuated Waves

The flutter waves (F waves) and fibrillatory waves (f waves) are usually observed in atrial flutter and atrial fibrillation, respectively. These waves tend to occur several times per a single cardiac cycle. The medical causes and treatments are beyond this chapter. This proposed model, however, only proved these parameters to support the behaviour simulation of such waves.

Table 4.11: Parameters for Frequency of Fluctuated Waves

Parameter	Valid Value	Annotations
Fx	$\mathbb{Z}_{\geqslant 0}$	F wave
fx		f waves

Event Enablement

All events in this model can either be *enabled* or *disabled* using the parameters in Table 4.12, excluding the ⟨*TP segment*⟩. An enabled event will be produced by the model when its conditions are met. In contradiction, disabled events will not be generated nor shown in ECG data. The enablement of the ⟨*TP segment*⟩ is not entirely explicit. As discussed in Equation 4.4, it is possible to reduce the duration of the ⟨*TP segment*⟩ to nearly *zero*, but due to this model's mathematical calculation, the ⟨*TP segment*⟩ can not be disabled. The model uses the ⟨*TP segment*⟩ as an adjustable coordinator between the cardiac rhythms. For example, if the defined durations of all enabled events are less than the defined value of the parameter ⟨*set_bpm*⟩ (Table 4.8), then the Equation 4.4 will increase the duration of the ⟨*TP segment*⟩ to satisfy the desired number of beats per minute, which keep the model properly executable.

Table 4.12: Parameters for Event Enablement

Parameter	Valid Value	Annotations
P_ENA	*true, false*	P wave
PR_ENA		PR segment
Q_ENA		Q wave
R_ENA		R wave
S_ENA		S wave
ST_ENA		ST segment
T_ENA		T wave
TU_ENA		TU segment
U_ENA		U wave

Event Ratio

In the colour set **interval and ratio** defined in Table 4.5, the event ratio was discussed thoroughly with an example as in 4.27. If the input value of an event ratio equals *zero*, then the event will always be generated (i.e. never dropped). These parameters in Table 4.13 depend on the parameters defined in Table 4.12. If an event is *disabled*, then its ratios will be dismissed. Also, there is no ratio for the event $\langle TP\ segment \rangle$ for the same reason as stated in the previous section for (c.f. Table 4.12)

Table 4.13: Parameters for Event RatioSetup

Parameter	Valid Value	Annotations
Pr	$\mathbb{Z}_{\geqslant 0}$	P wave
PRr		PR segment
Qr		Q wave
Rr		R wave
Sr		S wave
STr		ST segment
Tr		T wave
TUr		TU segment
Ur		U wave

Dropped Beat

Heart arrhythmia can be classified in accordance with some properties, such as dropped beats within a specified time. These parameters in Table 4.14 proved the feasibility of controlling dropped beat in order to simulate complex cardiac conduction system characteristics. The functions of these parameters have been discussed in the colour set **interval and ratio** defined in Table 4.5.

Table 4.14: Parameters for Dropped Beat

Parameter	Valid Value	Annotations
dhb_ENA	*true, false*	Enable heartbeat dropping (c.f. Dropped Beat Ratio)
dhb_ratio	$\mathbb{Z}_{\geqslant 0}$	(beat:beat(s)) as explained in 4.26
dhb_cd	$\mathbb{R}_{\geqslant 0}$	If the input value equals *zero*, then the duration of one dropped heartbeat automatically equals the total time of all enabled events

Heart and Leads Axes

The parameters of Table 4.15 and Table 4.16 are applied in the proposed algorithm, in the colour set **ECG**, to simplify the declaration process when defining the leads' values such as in Equation 4.2.1.2. Even though most of those axes are usually retain certain values, these parameters shown in Table 4.15 and

Table 4.16 support the further analysis of different heart behaviours and external circumstances. The use of these parameters and their related algorithm should be based on the desired results as either formal or estimated declaration of leads' values.

Table 4.15: Parameters for Leads Axes

Parameter	Valid Value	Annotations
lead_I	$\mathbb{Z}$	The axis from Lead I
lead_II		The axis from Lead II
lead_III		The axis from Lead III

Table 4.16: Parameters for Heart Axes

Parameter	Valid Value	Annotations
atrial_dep_axis	$\mathbb{Z}$	atrial depolarization
septal_dep_axis		septal depolarization
major_ventricular_dep_axis		major ventricular depolarization
basal_dep_axis		basal depolarization
ventricular_rep_axis		ventricular repolarization
papillary_muscles_rep_axis		papillary mucles repolarization
zero_axis		It is used to return zero when another axis is multiplied by this parameter.

Activate Leads

These parameters, in Table 4.17, denote the behaviour of selecting some leads for ECG recording. When a parameter is set to *false*, then no data related to this lead will be calculated, but rather the lead's value holds zero.

Table 4.17: Parameters for Activate Leads

Parameter	Valid Value	Annotations
lead_I_connected	*true, false*	Bipolar Limb Leads
lead_II_connected		
lead_III_connected		
lead_aVR_connected		Augmented Limb Leads
lead_aVL_connected		
lead_aVF_connected		
lead_V1_connected		Precordial Leads
lead_V2_connected		
lead_V3_connected		
lead_V4_connected		
lead_V5_connected		
lead_V6_connected		

Precordial Lead Variables

Table 4.18 shows parameters that can be applied to further calculation with precordial leads. As demonstrated with the proposed algorithm, some parameters may be utilized to apply some leads' values to estimate other leads' values.

Table 4.18: Parameters for Precordial Lead Variables

Parameter	Valid Value	Annotations
V1v	$\mathbb{R}$	Lead V1
V2v		Lead V2
V3v		Lead V3
V4v		Lead V4
V5v		Lead V5
V6v		Lead V6

4.2.2 Atrial-Depolarization submodel

As shown in Figure 4.3, this submodel demonstrates the CCS event; atrial depolarization. In a healthy heart, atrial depolarization is typically recorded in ECG as a wave named $\langle P\ wave \rangle$. The $\langle P\ wave \rangle$ reflects the depolarization of both the left and right atrium. In ECG, the ⟨*P wave*⟩ is normally smooth and rounded. The regular duration of the ⟨*P wave*⟩ is no more than 0.11 seconds, and its natural amplitude is no more than 2.5 mm. The normal ⟨*P wave*⟩ is always positive in the ⟨*lead I*⟩, ⟨*lead II*⟩, ⟨*lead aVF*⟩, and all the precordial leads (i.e. ⟨leads *V1* through *V6*⟩).

Like the natural pacemaker of the heart, this submodel simulates the Sinoatrial Node (SA) by the place ⟨*SA node*⟩, which has a colour token of the colour set **pacemaker** defined in table 4.1. The behaviour of the natural SA node is producing impulses that cause atrial depolarization. Accordingly, the token of the place ⟨*SA node*⟩ represents a produced impulse that begins the atrial depolarization when the transition ⟨*start new heartbeat*⟩ fires. This transition is only enabled if its guard, which represents appropriate rules, is holding. The specific rules that must be satisfied are as follows:

- If the value of the parameter ⟨*number of heartbeats*⟩ (defined in Table 4.7) equals zero, then the transition is always enabled and fires when a token is available in its input place (i.e. the place ⟨*SA node*⟩)
- If the value of the parameter ⟨*number of heartbeats*⟩ is greater than zero, then the transition is enabled and fires as many times as the value of the parameter.
- If the value of the parameter ⟨*number of heartbeats*⟩ is less than zero, then the whole model fails to execute any simulation.

In addition, the token of the place a has the timestamp in the form of the real number. The value of this timestamp initially equals the value of the parameter ⟨*initial time*⟩ (defined in Table 4.7), and thereafter equals the continuous cumulative timestamps of the model.

After firing transition ⟨*start new heartbeat*⟩, the colour token of the colour set **pacemaker** is removed from place ⟨*SA node*⟩. In CPNs, transitions allow the transformation of colour tokens from one colour set to another colour set according to the model's specified controls. In this submodel, the firing of transition ⟨*start new heartbeat*⟩ transfer the colour token from the colour set **pacemaker** to the colour set **myocytes**. This transformation is performed through the inscription of the output arc, which was defined as ⟨*cation*⟩ (see Listing 4.7). Consequently, the assigned colour token into the place ⟨*atria pre-stimulation*⟩ is valued to ⟨*electropositive*⟩, which has the colour set **myocytes** as defined in Table 4.19.

Table 4.19: The definition of colour set myocytes

Colour Set	Definition
myocytes	$\mathit{electronegative} \cup \mathit{electropositive}$

```
colset myocytes = with
  electronegative (* i.e. depolarization *) |
  electropositive (* i.e. repolarization *) timed;
```

Listing 4.6: The CPN ML definitions of colour set myocytes

```
val cation = electropositive;
val anion= electronegative;
```

Listing 4.7: The CPN ML definitions of cation and anion as colour tokens

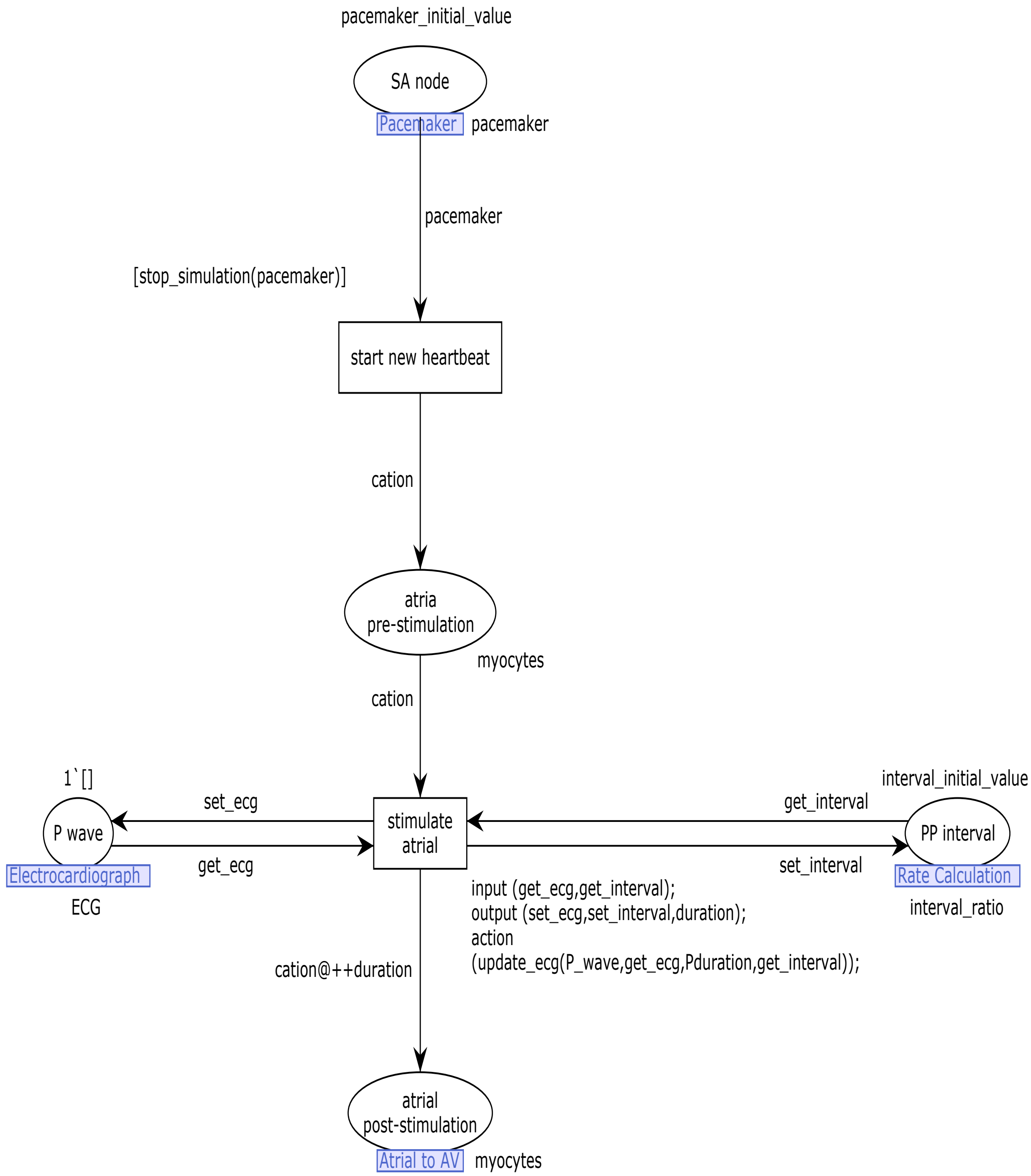


Figure 4.3: The Atrial-Depolarization submodel

In this submodel and some other submodels, each CCS event is approached by three stages; *pre-stimulation*, *stimulation*, and *post-stimulation*. This applied divide-and-conquer approach not only helps in understanding the event but also facilitates potential future expansion for the model to efficiently adopt additional desired details.

```
input(get_ecg,get_interval);
output(set_ecg,set_interval,duration);
action(update_ecg(P_wave,get_ecg,Pduration,get_interval));
```

Listing 4.8: The code segment of the transition ⟨*stimulate atrial*⟩

The transition ⟨*stimulate atrial*⟩ is enabled by default once colour tokens are available in the input places. However, this transition ⟨*stimulate atrial*⟩ has an attached code segment, which executes a CPN ML code upon firing the transition. This code segment, as demonstrated in Listing 4.8, involves input pattern, output pattern and code action. The behaviour of this transition, as a result of the code segment, is limited to three potential cases as follows:

- The first case is the ⟨*P wave*⟩ event is not enabled, and therefore both colour tokens of the places ⟨*P wave*⟩ and ⟨*PP interval*⟩ are returned with no further processing. Accordingly, this case simulates the disappearance of atrial depolarization (i.e. This event has zero duration in the ECG)

- The second case is the interval of the ⟨*P wave*⟩ is enabled and satisfied, then the colour token of the place ⟨*P wave*⟩ is updated by an event defined as dropped event, which has the same duration as the dropped ⟨*P wave*⟩ event but with zero amplitude. Also, the related trackers in the colour token in place ⟨*PP interval*⟩

are updated consequently. Accordingly, this case simulates the dropped of atrial depolarization (i.e. This event has some duration but zero amplitude in the ECG)

- The third case is the typical case where the colour token in the place $\langle$*P wave*$\rangle$ is replaced with a new $\langle$*P wave*$\rangle$ event according to its associated parameter. The relevant trackers of the colour token in the place $\langle$*PP interval*$\rangle$ are also updated correspondingly. Accordingly, this case simulates the regular atrial depolarization (i.e. This event has some duration and some amplitude in the ECG)

- In all three cases above, the colour token in place $\langle$*atrial pre-stimulation*$\rangle$ is transferred to place $\langle$*atrial post-stimulation*$\rangle$ with the attached timestamp of the simulated event's duration in accordance with the above case.

In this submodel, the recorded $\langle$*P wave*$\rangle$ is a list of the colour set **ECG** (defined in Table 4.4 and Listing 5.2), representing helpful information about the event. Furthermore, the colour set **ECG** can be redefined and extended to include extra information as required. This concludes the atrial-depolarization submodel.

4.2.3 AV-Node submodel

In CCS, after the atrial depolarization, the produced action potentials spread through the AV node towards the ventricular. This time delay after the atrial depolarization and before the ventricular depolarization is represented in ECG as a flat line name the $\langle$*PR segment*$\rangle$. The $\langle$*PR segment*$\rangle$ usually lasts between $0.12 - 0.20$ seconds with *zero* amplitude.

As shown in Figure 4.4, this submodel represents the $\langle$*PR segment*$\rangle$. This event begins when a token is received into place $\langle$*atrial post-stimulation*$\rangle$, which is a fusion place connecting this submodel from the previous submodel.

In CPNs, fusion places are functionally identical, implying all places under the same set of places will maintain the same functions and developments all the time. For example, if the declaration or the colour tokens of a place in a fusion set are modified, all the membered places in the same set will have the exact new declaration or colour tokens.

The overall behaviour of this submodel is consistent with the behaviour of the previous submodel. When the transition $\langle$*reaches AV node*$\rangle$ fires, the token is moved from place $\langle$*atrial post-stimulation*$\rangle$ to place $\langle$*VA pre-stimulation*$\rangle$. Subsequently, the firing of transition *stimulate VA* shifts the token from place $\langle$*VA pre-stimulation*$\rangle$ and updates the tokens of places $\langle$*PR seg*$\rangle$ and $\langle$*VA ratio*$\rangle$ concurrently in a similar approach as in the previous submodel.

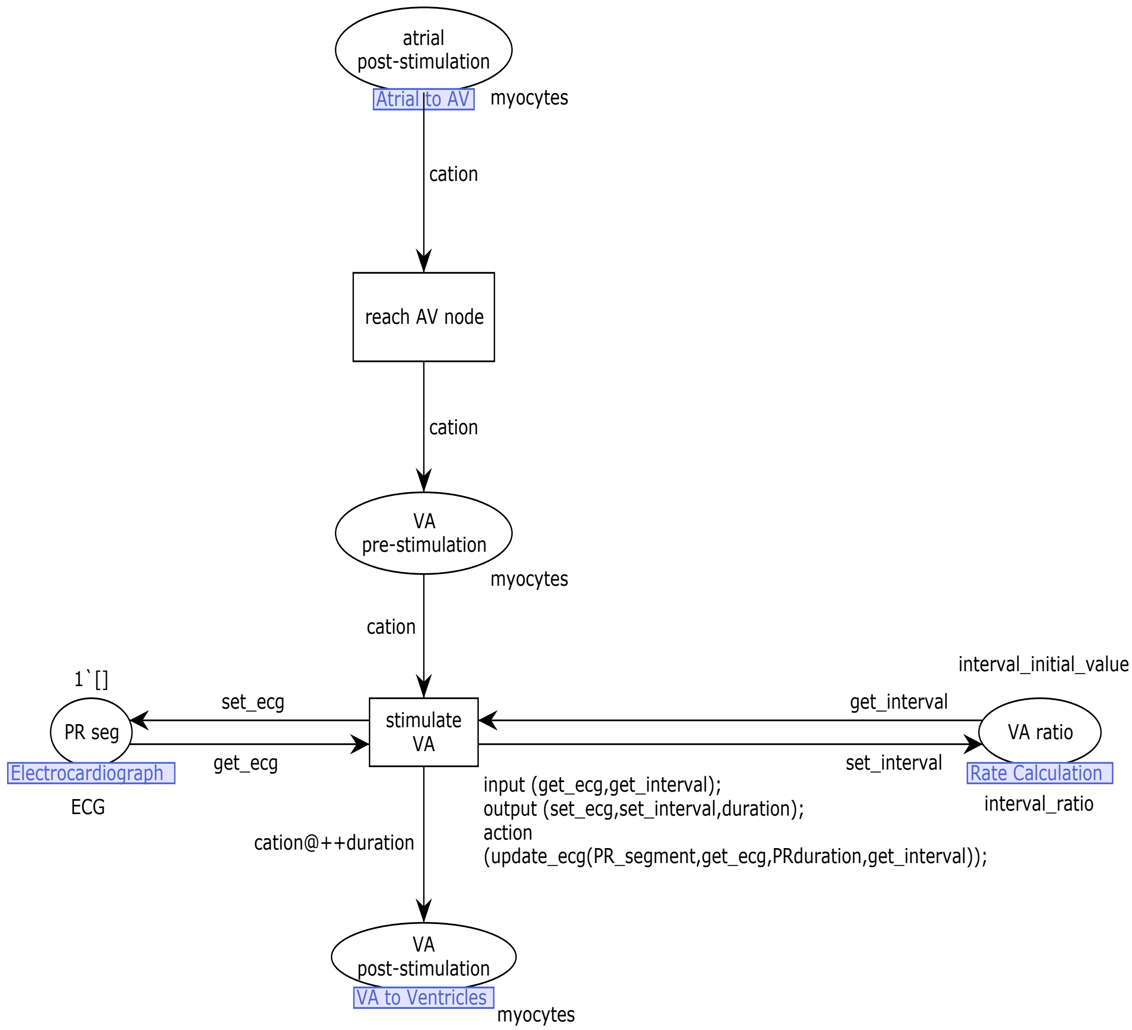


Figure 4.4: The VA-Node submodel

4.2.4 Ventricular-Depolarization submodel

This submodel, as shown in Figure 4.5, discusses the CSS event; ventricular depolarization. Due to the nature and size of the ventricular, its depolarization occurs sequentially within three different positions; septal, major ventricular, and basal. Consequently, in ECG, the regular ventricular depolarization consists of three waves named $\langle$*Q wave*$\rangle$, $\langle$*S wave*$\rangle$, and $\langle$*R wave*$\rangle$.

All these three waves form the *QRS complex*. The standard duration of the *QRS complex* is 0.06 up to 0.10 seconds, and the overall directions of the *QRS complex* can be normally observed positive in $\langle$*lead I*$\rangle$, $\langle$*lead II*$\rangle$, $\langle$*lead aVL*$\rangle$, $\langle$*lead V4*$\rangle$, $\langle$*lead V5*$\rangle$, and $\langle$*lead V6*$\rangle$, as well as, negtive in $\langle$*lead III*$\rangle$, $\langle$*lead aVF*$\rangle$, $\langle$*lead aVR*$\rangle$, $\langle$*lead V1*$\rangle$, $\langle$*lead V2*$\rangle$, and $\langle$*lead V3*$\rangle$.

Like the previous two submodels, this submodel also follows the same implemented approach and procedures. When a colour token arriving at the place $\langle$*VA post-stimulation*$\rangle$, the colour token is transferred to the place $\langle$*QRS complex pre-stimulation*$\rangle$ upon the firing of the transition $\langle$*reach ventricle*$\rangle$. This action holds the exact timestamp as before and after firing the transition.

While this transition, $\langle$*reach ventricle*$\rangle$, carries no significant role in modelling the ventricular-depolarization event, it is intended to serve two objectives. The first objective is to simplify the modelled event, in particular when colour tokens are exchanged between distinct submodels, and the second objective is to support further extension for the submodel.

The colour token of the place $\langle$*QRS complex pre-stimulation*$\rangle$ moves sequentially across desired places via the firing of associated transitions until the colour token reaches the place $\langle$*QRS complex post-stimulation*$\rangle$. Consequently, the other colour tokens of connected places are appropriately updated.

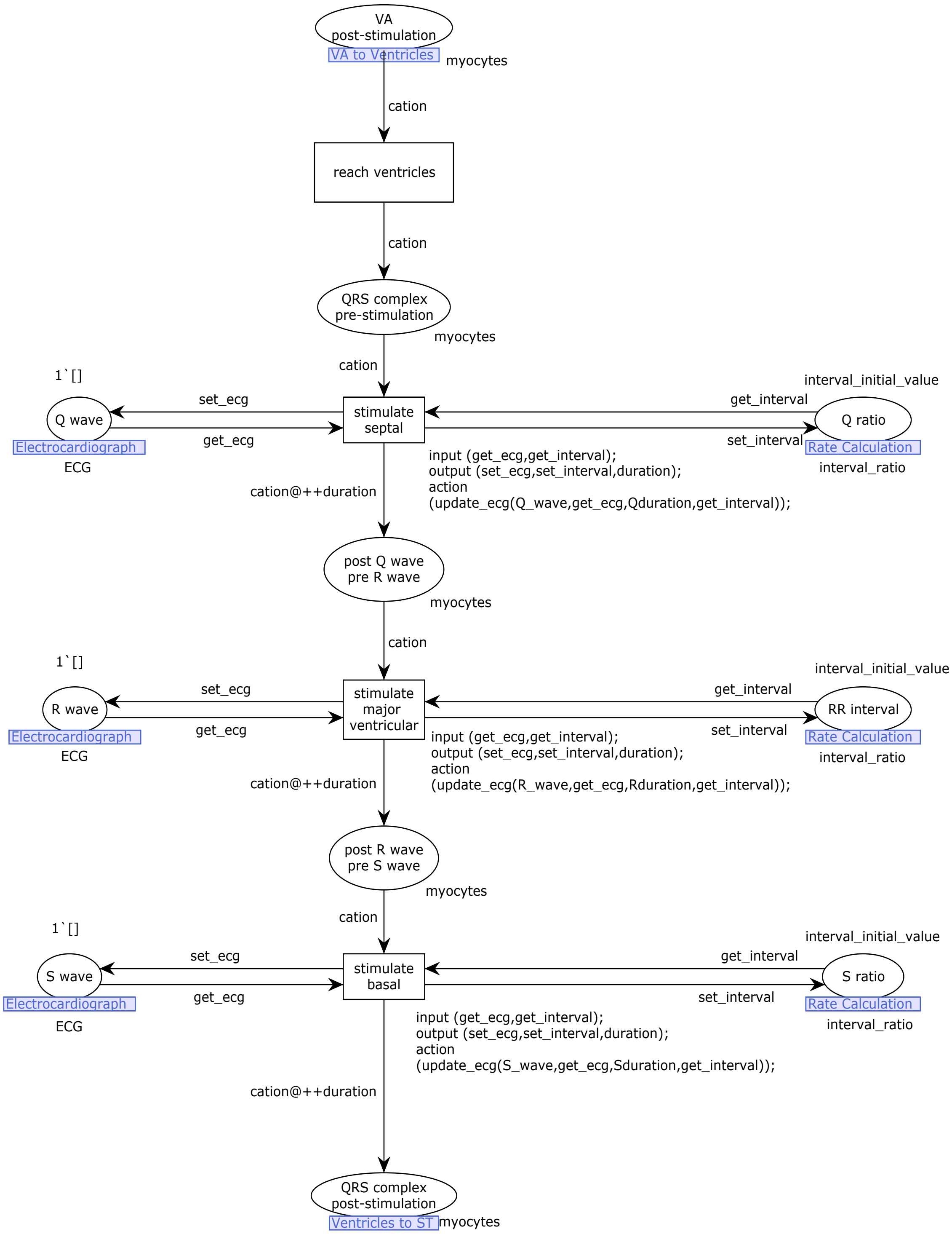


Figure 4.5: The Ventricular-Depolarization submodel

4.2.5 ST-Segment submodel

In Figure 4.6, the ST-segment submodel, as the name suggests, represents the ⟨*ST segment*⟩. In CCS, the ⟨*ST segment*⟩ indicates the delay after the ventricular depolarization and before its repolarization. In ECG, the ⟨*ST segment*⟩ is normally a flat line that lasts between 0.080 and 0.12 seconds.

This submodel continues to utilize the same approach as in the previous submodels. A received colour token in the place ⟨*QRS complex post-stimulation*⟩ is moved to the place ⟨*start of ST seg*⟩ when the transition ⟨*initiate ST segment*⟩ is fired. Accordingly, this colour token is redefined from cation to anion (see Table 4.19 and Listing 4.6) as a simulated behaviour of the associated CCS events. The colour token is then moved to the place ⟨*end of ST segment*⟩ after the transition ⟨*present ST segment*⟩ is fired and the colour tokens of places ⟨*ST seg*⟩ and ⟨*ST ratio*⟩ are properly updated.

4.2.6 Ventricular-Repolarization submodel

This last submodel demonstrates ventricular repolarization as the last CSS event in a single rhythm cycle. While it is possible to distinguish the repolarization of ventricular in ECG, the repolarization of atrial has no noticeable wave in ECG. The atrial depolarization occurs typically during ventricular depolarization, and since the amplitude of atrial repolarization is relatively smaller than the ventricular depolarization, the atrial repolarization is masked by the ventricular depolarization (i.e. QRS complex). Nevertheless, if an experiment decided to use this proposed model and requires a submodel for atrial repolarization, the already established atrial depolarization submodel (see 4.2.2) can then be cloned and adjusted adequately with the declaration of additional parameters.

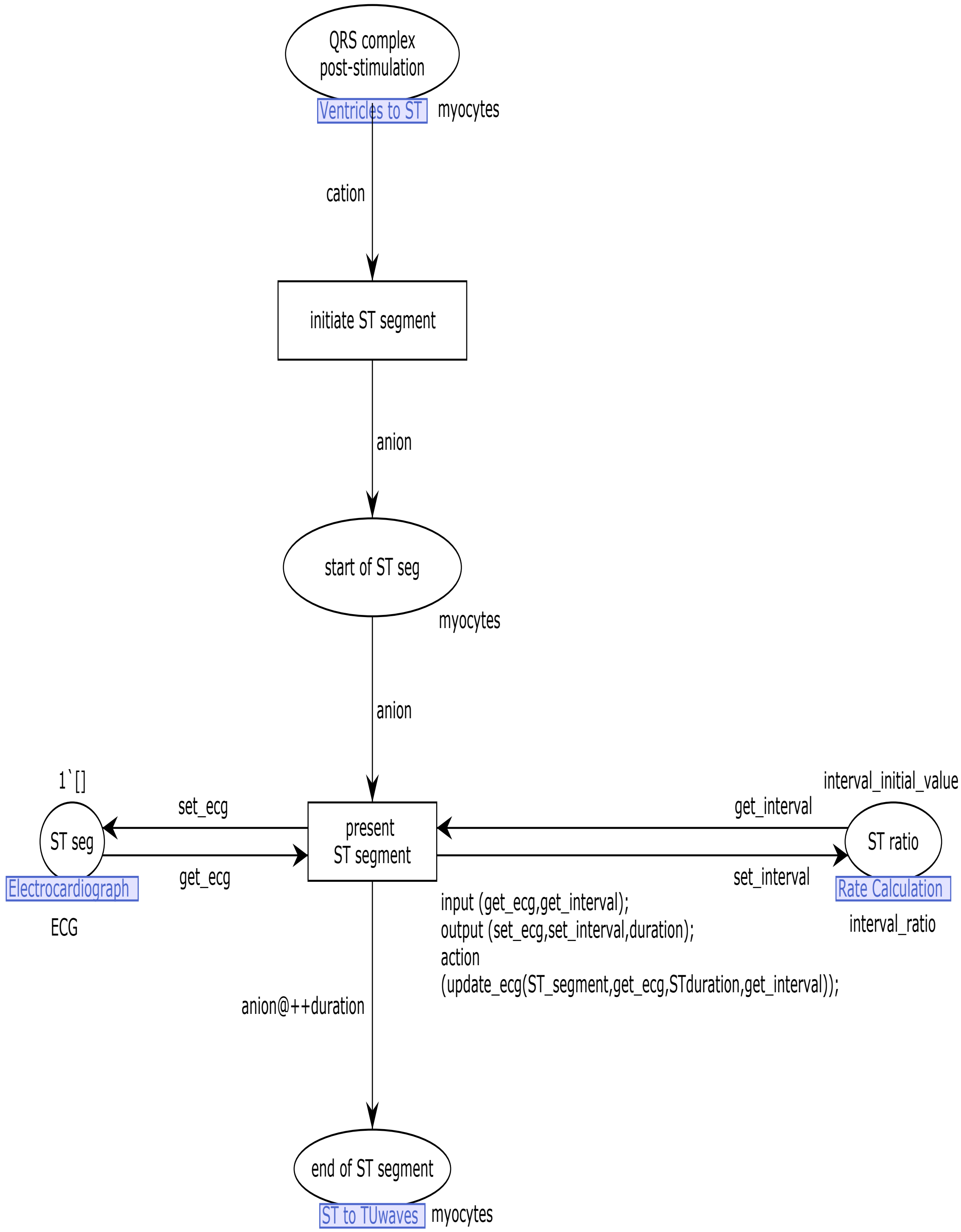


Figure 4.6: The ST-Segment submodel

This submodel includes the following ECG events: ⟨*T wave*⟩, ⟨*TU segment*⟩, ⟨*U wave*⟩, and also ⟨*TP segment*⟩. These events are correlated to the CSS event, ventricular repolarization. In ECG, the normal ⟨*T wave*⟩ appears slightly asymmetric as its peak tends to be near the end of the wave rather than its beginning. The ⟨*T wave*⟩ duration is between $0.10 - 0.25$ seconds, and it is ordinarily positive in the ⟨*lead I*⟩, ⟨*lead II*⟩, and leads ⟨*V2*⟩ through ⟨*V6*⟩, yet it is negative in ⟨*lead aVR*⟩. In ECG, the ⟨*U wave*⟩ is uncommon for many people. However, for some biological reasons, the ⟨*U wave*⟩ can be observed. If the ⟨*U wave*⟩ appears, it usually follows the same direction as the ⟨*T wave*⟩, and the flat line between the two waves is called the ⟨*TU segment*⟩.

This submodel is consistent with the previous submodels as it utilized the same approach and producers. Once a colour token is available in the place ⟨*End of ST segment*⟩, it is shifted via the transition ⟨*start ventricular repolarization*⟩ to the place ⟨*ventricular pre-stimulation*⟩ with the same *timestamp*. As this colour token is transferred down across parallel places by firing connected transitions, other colour tokens of other associated places are also updated thoroughly with respect to the values of the relevant *parameters* as discussed in the event-structure submodel.

While the segment codes of most transitions are similar, the transition ⟨*full heartbeat*⟩ holds a distinct segment code, as shown in Listing 4.9 (c.f. Listing 4.8).

```
input(get_ecg,get_interval);
output(set_ecg,set_interval,duration,pacemaker);
action(update_rate(TP_segment,get_ecg,(TP,0.0),get_interval));
```

Listing 4.9: The code segment of the transition ⟨*full heartbeat*⟩

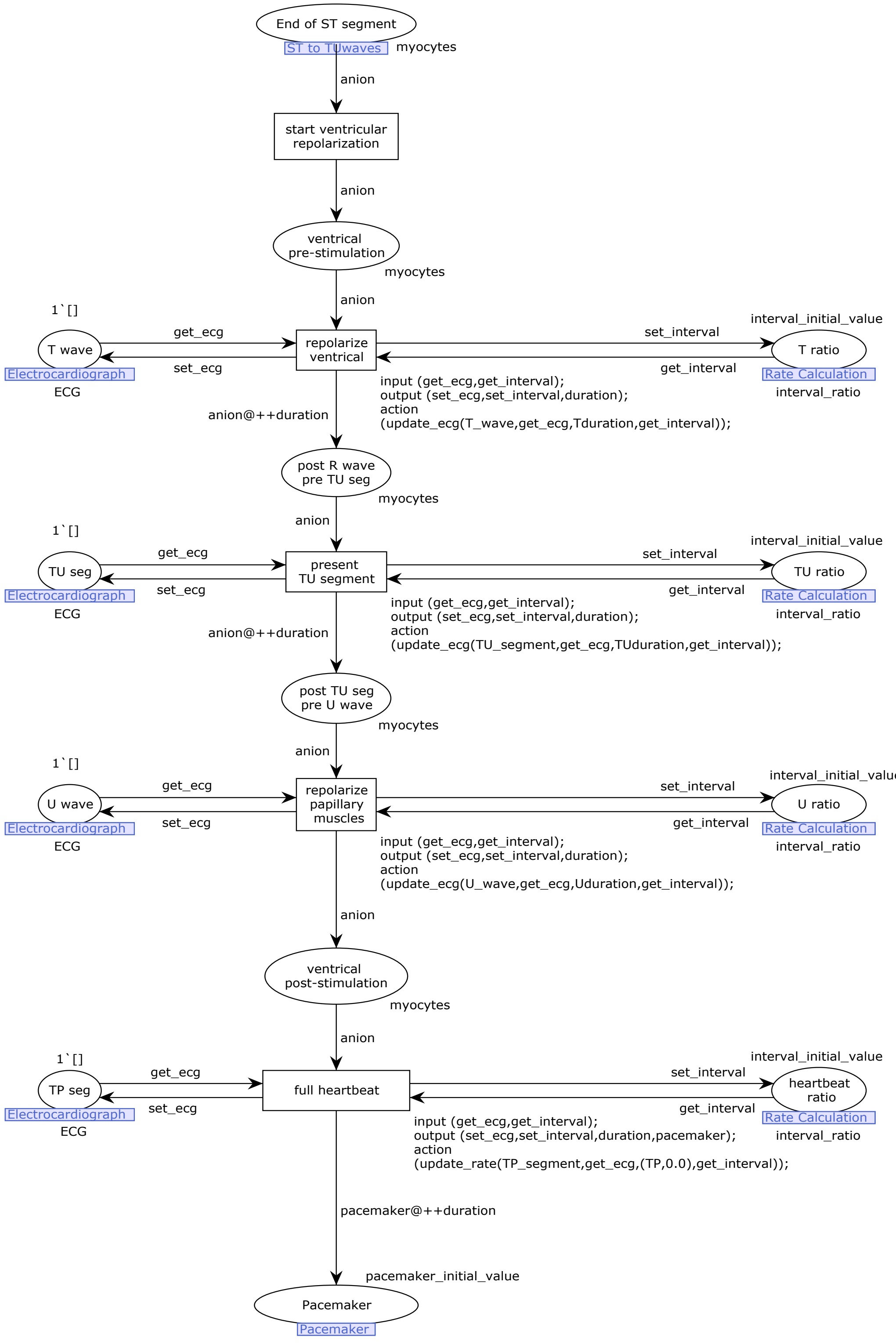


Figure 4.7: The Ventricular-Repolarization submodel

The simulated behaviour of this code segment, as in Listing 4.9, is as follows:

- Update the colour tokens of the place $\langle$*TP seg*$\rangle$ by inserting an ECG event for the $\langle$*TP segment* according to the defined parameters and calculated values.
- Update the colour tokens of the place $\langle$*heartbeat ratio*$\rangle$ by properly increasing the heartbeat counter and setting the complete cycle ratio.
- Calculate and return the event's duration time.
- Transform the received colour token from the place $\langle$*ventrical post-stimulation*$\rangle$ from $\langle$*anion*$\rangle$, which is based on the colour set **myocytes**, to $\langle$*pacemaker*$\rangle$, which is based on the colour set **pacemaker**. This transformation includes calculating the heart rate, set the heart rate's current condition, increasing the heartbeat counter, correctly determine the event's duration if the requirements of dropping a heartbeat hold.

4.3 Model Analysis

This section presents the analysis results of this proposed model in this chapter. Section 4.3.1 demonstrates the adoption of heart rhythms and arrhythmias into the model to review its capabilities and limitations. Sections 4.3.2 and 4.3.3 discuss the utilized tools and the documented results.

4.3.1 Heart Rhythms and Arrhythmias

In interpreting ECG data, different heart rhythms and arrhythmias can be identified according to specific characteristics [Garcia [2014]]. These characteristics are as follows:

- **Rate:** The rhythm rate, where the rate range, as presented in Table 4.3, is a critical key to recognize many abnormal rhythms.
- **Regularity:** The rhythm regularity, within intervals between P waves and QRS complexes, can follow a regular or irregular pattern.
- **P wave:** The P waves' presence and similarity are usually subject to some atrial or supraventricular components.
- **P:QRS ratio:** This ratio is beneficial to simulate some AV nodal blocks.
- **PR interval:** This aids in distinguishing a wandering atrial pacemaker.
- **QRS width:** This leads to examining the impulses moving towards the ventricular.
- **Grouping:** This allows informed determination regarding AV nodal block or recurrent premature complexes.
- **Dropped beats:** The occurrence of dropped beats is a sign of AV nodal blocks or sinus arrest.

This proposed model is capable of simulating different heart rhythms and arrhythmias. The majority of rhythms and arrhythmias, presented in [Garcia [2014]], are satisfactory by the model. For instance, Tables 4.20, 4.21, and 4.22 show different rhythms, in which the model can simulate with no further modifications (i.e. no extra places, transitions, parameters, or colour sets). In each Table, the characteristics of the normal sinus rhythm are given for reference. The normal sinus rhythm is the state where the SA node is normally controlling the heart pace. This rhythm has consistent intervals following the normal rage. The normal sinus rhythm solely concerns the atrial rate regardless of the state of the ventricular.

In Table 4.20, the sinus bradycardia is the state where the heart's rate is slower than 60 BPM, which may occur due to vagal stimulation or some medicines. The sinus pause is when the impulses from the sinus pacemaker are absent for some time periods.

In Table 4.21, the sinus arrhythmia is the state where rhythms are slower during exhalation but faster upon inhalation, which may happen due to some inhalation conditions. The idioventricular rhythm is when a ventricular functions as the primary pacemaker.

In Table 4.22, the first-degree heart block happens when the AV node experiences a prolonged physiologic block, which may cause by medication, disease, among others. The Mobitz II second-degree Heart Block is the state where beats are grouped, and some are dropped between each group. This state may occur by a diseased AV node.

However, a few rhythms require the model to have additional extensions in order to adopt more rhythms into the model. In Table 4.23, the ventricular fibrillation is the state of cardiac chaos where different heart parts are firing simultaneously with no organized activity. The Torsade de Pointes means twisting of points, and it happens when the QRS complexes repeatedly switch from positive to negative and vice versa.

In Table 4.24, the ventricular premature contraction (VPC) is the state where the firing of the ventricular pacer occurs before the firing of the normal SA node. Hence, upon firing the standard pacer (i.e. SA node), the ventricles are still in their refractory state and do not contract at the correct time. On the other hand, the

Table 4.20: The characteristics of some rhythms 1

Characteristics	Normal Sinus Rhythm	Sinus Bradycardia	Sinus Pause/Arrest
Rate:	$60 - 100$ BPM	Less than 60 BPM	Varies
Regularity:	Regular	Regular	Irregular
P wave:	Present	Present	Present except in areas of pause/arrest
P:QRS ratio:	1:1	1:1	1:1
PR interval:	Normal	Normal to slightly prolonged	Normal
QRS width:	Normal	Normal to slightly prolonged	Normal
Grouping:	None	None	None
Dropped beats:	None	None	Yes

Table 4.21: The characteristics of some rhythms 2

Characteristics	Normal Sinus Rhythm	Sinus Arrhythmia	Idioventricular Rhythm
Rate:	$60-100$ BPM	$60-100$ BPM	$20-40$ BPM
Regularity:	Regular	Varies with respiration	Regular
P wave:	Present	Normal	None
P:QRS ratio:	1:1	1:1	None
PR interval:	Normal	Normal	Normal
QRS width:	Normal	Normal	Wide ($\geqslant 0.12$ seconds)
Grouping:	None	None	None
Dropped beats:	None	None	None

Table 4.22: The characteristics of some rhythms 3

Characteristics	Normal Sinus Rhythm	First-Degree Heart Block	Mobitz II Second-Degree Heart Block
Rate:	$60 - 100$ BPM	Depends on underlying rhythm	Depends on underlying rhythm
Regularity:	Regular	Regular	Regular irregular
P wave:	Present	Normal	Normal
P:QRS ratio:	1:1	1:1	X:X-1
PR interval:	Normal	Prolonged > 0.20 seconds	Normal
QRS width:	Normal	Normal	Normal
Grouping:	None	None	Prsent and variable
Dropped beats:	None	None	Yes

ventricular escape beat is when the typical pacer's firing is seized (i.e. no firing from the SA node). Hence, a new timing cycle is reset by the pacer, and a different rate may appear.

Excluding the normal sinus rhythm, which is presented as a reference, all the mentioned rhythms in Tables 4.23 and 4.24 are a few examples of rhythms that can not be simulated with the proposed model. This limitation of the model's current state is due to the current definitions of parameters and colour sets. Nevertheless, one potential approach to facilitate the adoption of such rhythms is by extending the current colour sets to add new events and then defining new parameters that allow the new events to hold both arbitrary durations and inconsistent amplitudes.

For further details and classification of arrhythmia, the reader is referred to [Garcia [2014]], [Sembulingam and Sembulingam [2012]], and [Garcia [2014]].

4.3.2 Simulation-based Analysis

This section presents the analysis results of the proposed model using the simulation-based performance analysis, which was introduced in section 3.4. In the simulation-based performance analysis, the model was constantly executed to simulate various cardiac rhythms of different characteristics (section 4.3.1) and proper parameters (section 4.2.1.4). Consequently, the achieved results of this analysis are as follows:

- This proposed TCPN-based model enhanced a more solid understanding of the adopted concepts of the cardiac conduction system (CCS). In this analysis, several logical and programmatical errors were identified and then appropriately resolved.

Table 4.23: The characteristics of some rhythms 4

Characteristics	Normal Sinus Rhythm	Ventricular Fibrillation (VFib)	Torsade de Pointes
Rate:	60 – 100 BPM	Indeterminate	200 – 250 BPM
Regularity:	Regular	Chaotic rhythm	Irregular
P wave:	Present	None	None
P:QRS ratio:	1:1	None	None
PR interval:	Normal	None	None
QRS width:	Normal	None	Variable
Grouping:	None	None	Variable sinusoidal pattern
Dropped beats:	None	No beats at all	None

Table 4.24: The characteristics of some rhythms 5

Characteristics	Normal Sinus Rhythm	Ventricular Premature Contraction (VPC)	Ventricular Escape Beat
Rate:	$60 - 100$ BPM	Depends on the underlying rhythm	Depends on the underlying rhythm
Regularity:	Regular	Irregular	Irregular
P wave:	Present	not present on the VPC	None in the VPC
P:QRS ratio:	1:1	No P waves on the VPC	None in the VPC
PR interval:	Normal	None	None
QRS width:	Normal	Wide (= 0.12 seconds), bizarre appearance	Wide (= 0.12 seconds), bizarre appearance
Grouping:	None	Usually not present	None
Dropped beats:	None	None	None

- The simulations show that the model correctly terminates in the desired consistent state in accordance with its specification, which indicates the proposed model adequately met its required specifications of the CCS.
- This analysis also exposed the model limitations of simulating some cardiac rhythms as discussed in section 4.3.1.

In addition, user-defined monitors, as shown in Figure 4.8, were defined to optionally export into text-based files some or all of the generated colour tokens of the ECG events. These exported data can then be utilized to evaluate and optimize different ECG algorithms and techniques sufficiently. Listings 4.10 and 4.11 show, respectively, sample parameters and their generated data, which the defined monitors exported.

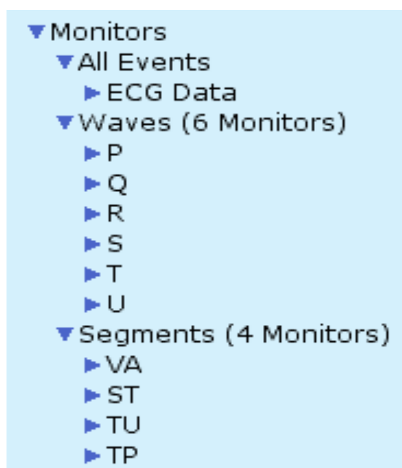


Figure 4.8: User-defined monitors to export generated events

```
     (* Waves Duration (ms :: milliseconds) *)
val Pd = (90.0,10.0);
val Qd = (10.0,5.0);
val Rd = (20.0,10.0);
     (* Waves Amplitude (V :: volts) *)
val Pa = {defult=2.5,I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF
   =0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0};
val Qa = {defult=2.0,I=0.0,II=0.0,III=0.0,VR=0.0,aVL=0.0,aVF
   =0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0};
val Ra = {defult=5.0,I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF
   =0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0};
val Seg = {defult=0.0,I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF
   =0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0};
```

Listing 4.10: Sample parameters used with user-defined monitors

```
[{id=1,title=P_wave,start_time=0.0,end_time=80.5837173018,assigned_duration=90.0,calculated_duration=80.5837173018,amplitude={I=1.25,II=2.5,III=1.25,aVR=~1.875,aVL=0.0,aVF=1.875,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
[{id=2,title=PR_segment,start_time=80.5837173018,end_time=171.120407978,assigned_duration=90.0,calculated_duration=90.5366906759,amplitude={I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF=0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
[{id=3,title=Q_wave,start_time=171.120407978,end_time=181.200502932,assigned_duration=10.0,calculated_duration=10.0800949545,amplitude={I=1.0,II=2.0,III=1.0,aVR=~1.5,aVL=0.0,aVF=1.5,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
[{id=4,title=R_wave,start_time=181.200502932,end_time=193.512303558,assigned_duration=20.0,calculated_duration=12.3118006263,amplitude={I=3.5,II=7.0,III=3.5,aVR=~5.25,aVL=0.0,aVF=5.25,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
[{id=5,title=S_wave,start_time=193.512303558,end_time=217.945428972,assigned_duration=20.0,calculated_duration=24.4331254132,amplitude={I=1.0,II=2.0,III=1.0,aVR=~1.5,aVL=0.0,aVF=1.5,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
[{id=6,title=ST_segment,start_time=217.945428972,end_time=315.484249038,assigned_duration=90.0,calculated_duration=97.5388200663,amplitude={I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF=0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]
```


Listing 4.11: Sample data recorded by the user-defined monitors

Furthermore, the exported data via the defined monitors can also be visualized, as shown in Figure 4.9, through tools such as [himanshu orsolutions [2020]] or [zapaaris [2020]] for the purpose of visual inspection or rhythms comprehension.

Figure 4.9: Visualizing the exported data from the user-defined monitors

Nevertheless, the simulation-based technique cannot guarantee that all possible executions were covered; therefore, the State Space analysis is also conducted to verify other functional and performance properties.

4.3.3 State Space Analysis

This section discusses the analysis results of the proposed model using the reachability analysis, which was introduced in section 3.4.

Applying the State Space analysis on various cardiac rhythms with characteristics compatible with the model (see section 4.3.1), consistent results are obtained. Just to mention a few, four different rhythms are presented in this section. These rhythms are sinus arrhythmia, sinus bradycardia, sinus tachycardia, and first-degree heart block with a total of 100, 200, 300, and 650 heartbeats, respectively. While section 4.3.1 introduces these rhythms, additional details regarding these rhythms can be found in [Garcia [2014]].

Limiting the number of heartbeats for selected rhythms is based on the behaviour of the modelled system and TCPNs. The model's analysis is conducted by generating the occurrence graph and its corresponding Strongly Connected Component (SCC) graph. Consequently, the reachability graph's size for rhythms with unlimited heartbeats can be infinite as several timed markings with global clock and timestamps can be distinguished. Thus, each rhythm was analyzed with a finite number of heartbeats.

The model is analyzed by generating the occurrence graph and its corresponding Strongly Connected Component (SCC) graph. However, the calculation of timed state space can be complicated and laborious in part because the size of the reachability graph can be infinite as several timed markings with global clock and timestamps can be distinguished. Therefore, four different rhythms were chosen and analyzed. These rhythms are sinus rhythm, sinus bradycardia, sinus tachycardia, and first-degree heart block (refer to [Garcia [2014]] for details about these rhythms) with a total of 100, 200, 300, and 650 heartbeats, respectively.

Table 4.25: Space State report: Statistics and Liveness Properties

Statistics					
Rhythms		1	2	3	4
Number of beats		100	200	300	650
Occurrence Graph (State Space)	Arcs	2250	3000	4500	6750
	Nodes	2251	3001	4501	6751
	Status	Full	Full	Full	Full
SCC Graph	Nodes	2251	3001	4501	6751
	Arcs	2250	3000	4500	6750
Liveness Properties					
Dead Markings		2251	3001	4501	6751
Dead Transition Instances		None	None	None	None
Live Transition Instances		None	None	None	None

Table 4.25 presents some statistical information and liveness properties, which were included in the analysis results of the elected rhythms. The size of both the State Space and SCC graph is reported by the statistics in Table 4.25 as the numbers

of nodes and arcs for each analyzed rhythm. The analysis's status for all rhythms was full, which indicates the applied analysis covered all possible states. According to the reported statistical information in Table 4.25, the numbers of nodes and arcs are matched in both of the occurrence graph and SCC graph for each rhythm. As expected, this implies that the model has no cyclic behaviour.

For all rhythms, the reported statistical results by state space have similar features. As shown in Table 4.25, the numbers of nodes and arcs in the SCC graph are always identical to the corresponding numbers of the state space. As expected, this implies that the model has no cyclic behaviour. Also, as the model terminated intentionally when the desired limited value of generated heartbeat was reached, a single dead marking is reported which causes the model to have no live transition instances. In contrast, the model has no dead transition instances, which indicates that all of the specified actions were executed.

In addition, Table 4.25 presents the liveness properties, which include *dead markings*, *dead transition instances*, and *live transition instances. Dead markings* indicate the unenabled binding elements. For each rhythm, a single dead marking is reported. It is typically expected since the model should terminate at a certain point (i.e. when the model reached the maximum number of assigned heartbeats).

Dead transition instances represent the transitions that were never enabled through the model's execution. All rhythms return *None* for the *dead transition instances*, which implies that all the transitions in the model fired at least once, and therefore all of the specified actions were executed.

In contrast, *live transition instances* denote the reachable transitions in the occurrence sequence of any reachable marking. All rhythms return *None* for *live transition* *instances* as the model is terminated intentionally following established rules (i.e. the number of heartbeat). Dead transitions and live transitions are not the reverses of each other. While a non-dead transition must fire at least once, a live transition should continue to fire.

The long Listings in Appendix C show the boundness properties, which report the number of the colour tokens in the places after executing all reachable markings.

The *Best Integer Bounds* (section 6.5) show the *upper* and the *lower* columns, which respectively indicate the maximal and minimal numbers of colour tokens remain in each place in any reachable marking. In the proposed model in this chapter, the places of the colour set **ECG** or **interval ration** always have a single token for both best upper and lower integer bounds meaning these places always hold a constant number of tokens. On the other hand, the other places hold at most one colour token when a colour token is available in the place, and also no colour token when an enabled transition transferred the colour token. All these reported numbers are accurate according to the specification of the system.

The *Best Upper and Lower Multi-set Bounds*, (sections 6.5 and 6.5) represent the produced colour tokens for each place in accordance with its defined colour set in any reachable marking. The *empty* multiset shows the absence of colour tokens by the specified colour set. Due to the massive number of produced colour tokens, this section contains just a few samples.

As shown in Listing 4.12, the home properties for all rhythms return no home marking, which indicates the initial marking is unreachable from any other markings. This result is accurate due to the system's specifications and time restrictions (i.e. a new beat with a different time for each cardiac cycle).

In addition, Listing 4.12 shows the fairness properties for the proposed model. As expected, all rhythms return no infinite occurrence sequences following the system's specification. In this proposed model, when a transition fires, it must transfer the assigned colour token, which keeps finite occurrence sequences of all markings. For some CCS events that are determined to be dropped, the transition will proceed to fire but update that event as a ⟨*dropped event*⟩ (see Table 4.4).

```
 Home Properties
-------------------------------------------------

  Home Markings
     Initial Marking is not a home marking



 Fairness Properties
-------------------------------------------------

     No infinite occurrence sequences.
```

Listing 4.12: Space State report: Home and Fairness Properties

CHAPTER

5

THE CARDIAC PACEMAKER SYSTEM (CPS)

This chapter presents the modelling of the cardiac pacemaker system. Section 5.1 introduces the system. Section 5.2 demonstrates the proposed model. Finally, section 5.3 discusses the analysis results of the model.

5.1 Introduction

The cardiac pacemaker system (CPS) can monitor and control heart rhythms, and also it can treat and reduce some risks associated with those heart rhythms. The CPS includes a pulse generator and one or more leads (among other components). The main functions of the pulse generator (also called the device) involve sensing and pacing operations.

In this chapter, a formal model for the cardiac pacemaker system is presented. Timed Coloured Petri Nets (TCPN) and CPN tools are adopted as modelling tools. The model, which includes various suitable parameters, covers numerous identified characteristics of operation modes and cardiac rhythms in substantial detail. The model can help facilitate a more solid understanding of the cardiac pacemaker system, validate and verify the interdisciplinary requirements of the cardiac pacemaker system, and provide customizable data to fully evaluate and optimize different algorithms and techniques for the cardiac pacemaker system.

Furthermore, while the model of the cardiac pacemaker system (CPS) inherits several colour sets and parameters from the previous model of the cardiac conducting systems (CCS), both models are not identical nor equivalent. The CCS simulates the cardiac electrical activities of the heart, which represents the labouring environment of the CPS. The CPS simulates the device that can affect the CCS. The CCS is an independent system that normally does not require the CPS unless CCS is suffering some abnormal conditions. The CPS, however, is an example of embedded systems that demand an environment in which to operate.

5.2 TCPN-based Modelling of CPS

Different from former studies regarding the cardiac pacemaker system's modelling, this proposed model is composed of seven interconnected but independent submodels. Such a structure not only empowers design simplicity and flexibility but also improves observability and design coupling. These submodels include the core-elements submodel, the atrial-depolarization submodel, the AV-node submodel, the ventricular-depolarization submodel, the ST-segment submodel, the ventricular-repolarization submodel, and the artificial-pacemaker submodel.

The functions and connections between these submodels are described as follows: The core-elements submodel represents the common four places among other submodels. This submodel is a part of all other submodels that exchange data across it. The atrial-depolarization submodel captures the activities of the depolarization of atrial musculature. In a healthy heart, the cardiac cycle begins with the firing of the SA node (i.e. the natural pacemaker) followed by the depolarization of atrial musculature producing the recordable P-wave in the ECG. The AV-node submodel addresses the event that follows the end of atrial depolarization. When action potentials spread through the AV node, this results in the PR segment in ECG. This segment is the flat line between the end of the P wave and the start of the QRS complex.

The ventricular-depolarization submodel processes the right and left ventricles' depolarization and the recordable QRS complex. The ST segment submodel addresses the time interval between the ventricular depolarization and repolarization, called the ST segment in ECG. The ventricular-repolarization submodel, as the name suggests, presents ventricular repolarization. Eventually, the artificial-pacemaker submodel monitors all the cardiac activities, ensures the implementation of pacemaker rules, and fires when specific rules, set by the programmed parameters, are met.

5.2.1 Core-Elements Submodel

This submodel, as shown in Figure 5.1, consists of four places: ⟨*Pacemaker parameters*⟩, ⟨*ECG Events*⟩, ⟨ *Intervals and Ratios*⟩ and ⟨*Timing Cycle*⟩. These places also belong to other submodels. The main reason for modelling the core-elements as an independent submodel is that all other submodels contain these places of the core-element submodel. Therefore, the places are defined in an independent submodel that has a recognizable name.

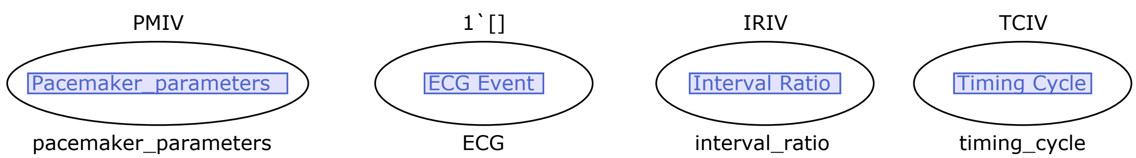


Figure 5.1: The Core-Elements Submodel

5.2.1.1 The Place: Pacemaker Parameters

The first place has the colour set **Pacemaker Parameters**, which is a record colour set or the Cartesian product of the sets as described in Table 5.1 and Listing 5.1. In the colour set **Pacemaker Parameters**, ⟨*mode*⟩ refers to the operation mode of the artificial pacemaker. The operation modes were interpreted thoroughly in Table 2.1.

In dual-chamber modes, the ⟨*last paced chamber*⟩ and ⟨*last sensed chamber*⟩ are utilized to maintain the order by the following rules:

- If the operation mode is set only to sense the atrial, then the ⟨*last sensed chamber*⟩ always equals ⟨*A*⟩.

- If the operation mode is set only to sense the ventricular, then the $\langle$*last sensed chamber*$\rangle$ always equals $\langle V\rangle$.

- If the operation mode is set to sense none, then the $\langle$*last sensed chamber*$\rangle$ always equals $\langle O\rangle$.

- If the operation mode is set to sense both the atrial and the ventricular, then the $\langle$*last sensed chamber*$\rangle$ is updated upon sensing the desired chamber. Consequently, if the $\langle$*last sensed chamber*$\rangle$ equals $\langle A\rangle$, then the last sensed chamber is updated to equal $\langle V\rangle$, and vice versa.

- The rules for the $\langle$*last paced chamber*$\rangle$ are applied similarly.

- The $\langle$*last paced chamber*$\rangle$ and the $\langle$*last sensed chamber*$\rangle$ are maintained individually in accordance with the meaning of the implemented operation mode.

```
colset pacing_source = with SA_node | AV_node |
   Purkinje_Fibers | None | artificial_pacemaker;
colset rate_condition_set = with Fibrillation | Flutter |
   Tachycardia | Normal | Bradyarrhythmia | NA;
colset code_symbole = with  O | A | V | D | I | T | R;
colset mode = with Off | DDDR | VDDR | DDIR | DOOR | VOOR |
   AOOR | VVIR | AAIR | DDD | VDD | DDI | DOO | VOO | AOO |
   VVI | AAI | VVT | AAT | OVO | OAO | ODO | OOO;
colset pacemaker_parameters = record mode:mode *
   last_paced_chamber:code * last_sensed_chamber:code * source
   :pacing_source * pre_assigned_bpm:INT * pre_rate_condition:
   rate_condition_set * Pwave_based_bpm:INT *
   Pwave_rate_condition:rate_condition_set * Rwave_based_bpm:
   INT * Rwave_rate_condition:rate_condition_set *
   heartbeats_counter:INT;
```

Listing 5.1: The CPN ML definitions of colour set Pacemaker Parameters

Table 5.1: The definitions of colour set Pacemaker Parameters

Colour Set	Definition
pacing source	*SA node* $\cup$ *AV node* $\cup$ *Purkinje Fibers* $\cup$ *None* $\cup$ *Artificial Pacemaker*
rate condition	*Fibrillation* $\cup$ *Flutter* $\cup$ *Tachycardia* $\cup$ *Normal* $\cup$ *Bradycardia* $\cup$ $\{NA\}$
code symbole	$O \cup A \cup V \cup D \cup I \cup T \cup R$
mode	*Off* $\cup$ *DDDR* $\cup$ *VDDR* $\cup$ *DDIR* $\cup$ *DOOR* $\cup$ *VOOR* $\cup$ *AOOR* $\cup$ *VVIR* $\cup$ *AAIR* $\cup$ *DDD* $\cup$ *VDD* $\cup$ *DDI* $\cup$ *DOO* $\cup$ *VOO* $\cup$ *AOO* $\cup$ *VVI* $\cup$ *AAI* $\cup$ *VVT* $\cup$ *AAT* $\cup$ *OVO* $\cup$ *OAO* $\cup$ *ODO* $\cup$ *OOO*
pacemaker Parameters	$\{($ *mode, last paced chamber, last sensed chamber, source, pre assigned bpm, pre rate condition, Pwave-based bpm, Pwave rate condition, Rwave-based bpm, Rwave rate condition, heartbeats counter*$)$ $\mid$ *mode* $\in$ *mode, last paced chamber, last sensed chamber* $\in$ *code symbole, source* $\in$ *pacing source, pre assigned bpm, Pwave-based bpm, Rwave-based bpm, heartbeats counter* $\in \mathbb{Z}_{\geqslant 0}$, *pre rate condition, Pwave rate condition, Rwave rate condition* $\in$ *rate condition* $\}$

Furthermore, in the colour set **Pacemaker Parameters**, $\langle$*source*$\rangle$ indicates the generator of a heartbeat, as shown in the colour set **pacing source**. The colour set **pacing source** is inherited from the previous model (i.e. the CCS model as in section 4.2.1.1), and in this model, the colour set **pacing source** is extended to include $\langle$*Artificial Pacemaker*$\rangle$ as a substantial pacing source.

In the colour set **Pacemaker Parameters**, the $\langle$*source*$\rangle$ is determined based on these rules:

- If the artificial pacemaker releases an impulse, the $\langle$*source*$\rangle$ equals $\langle$*Artificial Pacemaker*$\rangle$.
- Otherwise, the $\langle$*source*$\rangle$ follows the origin of the impulse in accordance with the rhythmicity rates as presented in Table 4.2.

In the colour set **Pacemaker Parameters**, the $\langle$*pre assigned bpm*$\rangle$ returns the value of the parameter $\langle$*set_bpm*$\rangle$, and according to its value, the $\langle$*pre rate condition*$\rangle$ is specified as explained in Table 4.3 and shown in Listing 4.2. The $\langle$*Pwave-based bpm*$\rangle$ and the $\langle$*Rwave-based bpm*$\rangle$ refer to the heart rate based on the calculation of PP-interval and RR-interval, as discussed thoroughly in section 4.2.1.1. The calculated results of $\langle$*Pwave-based bpm*$\rangle$ and the $\langle$*Rwave-based bpm*$\rangle$ are utilized to determine the conditions of $\langle$*Pwave rate condition*$\rangle$ and $\langle$*Rwave rate condition*$\rangle$ following Table 4.3 and Listing 4.2. Finally, the $\langle$*heartbeat counter*$\rangle$ traces the number of generated full heartbeats.

5.2.1.2 The Place: ECG Events

The second place, as shown in Figure 5.1, has the colour set **ECG**. This colour set is inherited and extended from the CSS model (section 4.2.1.2). The primary objective of this colour set is to present the produced ECG events describing the cardiac electrical activities.

Table 5.2: The definition of colour set ECG

Colour Set	Definition
event	{*P-wave, Q-wave, R-wave, S-wave, T-wave, U-wave, PR-segment, ST-segment, TU-segment, TP-segment, F-wave, f-wave, dropped-Pwave, dropped-Qwave, dropped-Rwave, dropped-Swave, dropped-Twave, dropped-Uwave, dropped-PRsegment, dropped-STsegment, dropped-TUsegment,paced-Pwave, paced-Qwave, paced-Rwave, paced-Swave, paced-Twave, paced-Uwave, paced-PRsegment, paced-STsegment, paced-TUsegment, paced-TPsegment,artificial-pacing, artificial-sensing, atrial-pacing, ventricular-pacing*}
bipolar limb leads	$\{(I,\ II,\ III\ )\mid bipolar\ limb\ leads \in \mathbb{R}\}$

. . . continued

...continued

Colour Set	Definition
augmented limb leads	$\{(aVR,\ aVL,\ aVF\ ) \mid augmented\ limb\ leads \in \mathbb{R}\}$
precordial leads	$\{(\ V1,\ V2,\ V3,\ V4,\ V5,\ V6\ ) \mid precordial\ leads \in \mathbb{R}\}$
leads (i.e. the standard 12 leads)	*bipolar limb leads* $\cup$ *augmented limb leads* $\cup$ *precordial leads*
ECG Event	$\{(ID,\ title,\ start\ time,\ end\ time,\ duration,\ amplitude) \mid ID \in \mathbb{Z}_{\geqslant 1},\ title \in event,\ start\ time,\ end\ time,\ duration \in \mathbb{R}_{\geqslant 0},\ amplitude \in leads\}$

Similar to the definition of the colour set **ECG Event** in Table 4.4 and Listing 5.2, each event is expressed through a colour token, which is recognized by a unique $\langle$*ID*$\rangle$ and a classified $\langle$*title*$\rangle$.

In this model, for each cardiac cycle, each cardiac electrical activity is reported in the ECG as a regular event, a dropped event, or a paced event (based on the model's parameters). For example, the event $\langle$*P wave*$\rangle$ is typically displayed as a regular $\langle$*P_wave*$\rangle$ for a regular atrial depolarization. If the atrial depolarization is absent during a heartbeat, then this event is recorded as a $\langle$*dropped_Pwave*$\rangle$ (c.f. section 4.2.1.2 for details regarding dropped events). The event $\langle$*Paced_Pwave*$\rangle$ is stated if the atrial depolarization occurs during or following artificial pacing. Typically, in ECG, paced events tend to show wider durations and longer amplitudes. Nevertheless, the parameters 5.2.1.5 allow for a precise declaration for each paced event.

As in the CSS model (section 4.2.1.2), the ⟨*start time*⟩ and ⟨*end time*⟩ state the time when the event began and finished. The calculated difference between ⟨*start time*⟩ and ⟨*end time*⟩ is stated in the ⟨*duration*⟩. The *amplitude* of the colour set **leads** (defined in Table 5.2) indicates the calculated total height of events through the values of 12 standard ECG leads.

```
colset event = with P_wave | Q_wave | R_wave | S_wave | T_wave
    | U_wave | PR_segment | ST_segment | TU_segment |
   TP_segment | dropped_Pwave | dropped_Qwave | dropped_Rwave
   | dropped_Swave | dropped_Twave | dropped_Uwave |
   dropped_PRsegment | dropped_STsegment | dropped_TUsegment |
    Paced_Pwave | Paced_Qwave | Paced_Rwave | Paced_Swave |
   Paced_Twave | Paced_Uwave | Paced_PRsegment |
   Paced_STsegment | Paced_TUsegment | Paced_TPsegment |
   F_wave | f_wave | artificial_pacing | artificial_sensing |
   atrial_pacing | ventricular_pacing | N;
colset leads = record I:REAL * II:REAL * III:REAL * aVR:REAL *
    aVL:REAL * aVF:REAL * V1:REAL * V2:REAL * V3:REAL * V4:
   REAL * V5:REAL * V6:REAL;
colset voltages = record defult:REAL * I:REAL * II:REAL * III:
   REAL * aVR:REAL * aVL:REAL * aVF:REAL * V1:REAL * V2:REAL *
    V3:REAL * V4:REAL * V5:REAL * V6:REAL;
colset ECG_Event = record id:INT * title:event * start_time:
   REAL * end_time:REAL * duration:REAL * amplitude:leads;
colset ECG = list ECG_Event;
```

Listing 5.2: The CPN ML definitions of colour set ECG Events

The reader is referred to section 4.2.1.2 for additional details about the components of this colour set ECG.

5.2.1.3 The Place: Intervals and Ratios

The third place as in Figure 5.1 is ⟨*interval ratio*⟩, which holds the colour set **Interval Ratio** defined in Table 5.3 and in Listing 5.3. In principle, the colour set of this place carries numerical trackers that maintain the intervals and ratios of events.

This place and its colour set are inherited from the CCS model (section 1) with no further extension nor modification (c.f. Tables 4.5 and 5.3). Therefore, the reader is referred to section 4.2.1.3 for additional details and specifications.

Table 5.3: The definition of colour set Interval Ratio

Colour Set	Definition
interval ratio	{(*P previous interval, P last interval, R previous interval, R last interval, heartbeat counter, heartbeat ratio, P ratio, Q ratio, R ratio, S ratio, T ratio, U ratio, PR ratio, ST ratio, TU ratio*) | *interval ratio* $\in \mathbb{Z}_{\geqslant 0}$ }

```
colset interval_ratio = record P_previous_interval:INT *
   P_last_interval:INT * R_previous_interval:INT *
   R_last_interval:INT * heartbeat_counter:INT *
   heartbeat_ratio:INT * P_ratio:INT * Q_ratio:INT * R_ratio:
   INT * S_ratio:INT * T_ratio:INT * U_ratio:INT * PR_ratio:
   INT * T_ratio:INT * TU_ratio:INT;
```

Listing 5.3: The CPN ML definitions of colour set Intervals and Ratios

5.2.1.4 The Place: Timing Cycle

The ⟨*Timing Cylce*⟩ place in Figure 5.1 has the colour set **Timing Cycle**, which is a record colour set or the Cartesian product of the sets as defined in Table 5.4 and Listing 5.4.

Time is an integral part of any pacemaker system since the accuracy of operating modes depends massively on the correct time. This place and its colour set are established to monitor and regulate the associated timers for each operating mode.

Table 5.4: The definition of colour set Timing Cycle

Colour Set	Definition
timers	$\{(ARP \cup VRP \cup LRL \cup URL \cup MSR \cup ATR \cup AV\text{-}delay \cup V\text{-}blanking \cup A\text{-}blanking \cup PVARP \cup VA\text{-}interval) \mid timers \in \mathbb{R}\ \}$
timing cycle	$\{(ID,\ next\ event,\ last\ event,\ event\text{-}chunked,\ pacing\text{-}rate,\ ecg\text{-}duration,\ remain\text{-}duration,\ time\text{-}tracker,\ current\text{-}pacing\text{-}occurrenc,\ last\text{-}pacing\text{-}occurrence,\ next\text{-}pacing\text{-}duration,\ timers\ ) \mid ID,\ event\text{-}chunked \in \mathbb{Z}_{\geqslant 1}\ next\ event,\ last\ event \in event,\ pacing\text{-}rate,\ ecg\text{-}duration,\ remain\text{-}duration,\ time\text{-}tracker,\ current\text{-}pacing\text{-}occurrenc,\ last\text{-}pacing\text{-}occurrence,\ next\text{-}pacing\text{-}duration \in \mathbb{R}_{\geqslant 0},\ timers \in\ timers\}$

```
colset timer = list STRING;
colset timing_cycle = record ID:INT * next_event:event *
    last_event:event * event_chunked:INT * pacing_rate:REAL *
    ecg_duration:REAL * remain_ecg_duration:REAL * time_tracker
    :REAL * current_pacing_occurrenc:REAL *
    last_pacing_occurrence:REAL * next_pacing_duration:REAL *
    timers:timer timed;
```

Listing 5.4: The CPN ML definitions of colour set Timing Cycle

Besides identification, the $\langle ID \rangle$ in the colour set **Timing Cycle** is also essential in coordinating the time for some operation modes where pacing is the only assigned operation. For example, in the operation mode $\langle AOO \rangle$, the pacemaker is only pacing based on the value of its lower rate limit (LRL). Hence, to correctly deliver the pacing operation for each cardiac cycle, the following equation is applied:

$$ID \times LRL \tag{5.28}$$

where:

- ID is the identification number of the current pacemaker operation cycle, and it increases by one each time the artificial pacemaker is sensing or pacing following the defined rules of the operation modes.
- LRL is the lower rate limit for the adopted operation mode.
- The calculated result of this equation returns the accumulated LRL for the operation mode in each cardiac cycle.

In the colour set **Timing Cycle**, the ⟨*next event*⟩ and ⟨*last event*⟩ are utilized for accurate concurrent operations between the pacemaker's operations and the electrical cardiac activities if the assigned pacemaker's operation mode declares specific engagements. When the pacemaker's operations (i.e. pacing or sensing) are set to maintain the cardiac activities, then the ⟨*next event*⟩ and ⟨*last event*⟩ force rules in order to keep the model simulating the desired specifications correctly. These rules are subject to three cases:

- The first case is the artificial pacemaker operation (APO) is triggered during an electrical cardiac activity (ECA). Hence, the current ECA is interrupted just before the start time of the APO, and the name of the current ECA is stored in the ⟨*last event*⟩. Also, the ⟨*next event*⟩ is set to ⟨*artificial_pacing*⟩ (or ⟨*artificial_sensing*⟩ based on the operation mode).

- The second case is after the complete delivery of the APO. Then, the ⟨*next event*⟩ is set to either the previous or next event of the ECA. The previous event of ECA is applicable when it was interrupted and not thoroughly conducted (i.e. its end time is beyond the end time of the APO). Otherwise, the ⟨*next event*⟩ of ECA in accordance with the available events of the ECA (c.f. event enablement parameters).

- The last case is after the ECA is entirely conducted. Therefore, if no APO is scheduled, then the ⟨*next event*⟩ is set to the next enable event of the ECA.

- While these cases serve the major possible cases, they can also be extended or adjusted based on the modelled behaviour requirements.

Next in the colour set **Timing Cycle** is the ⟨*event-chunked*⟩, which assists in preventing potential starvation caused by paced events. As some cardiac events tend to show different characteristics after artificial pacing, this proposed model allows for distinguishing events as regular and paced (see Table 5.2). Paced events typically have wider durations and higher amplitudes than regular events. For example, in ECG, the regular ⟨*P wave*⟩ event duration is slightly increased after atrial pacing by the artificial pacemaker. Therefore, the implemented rules for the ⟨*event-chunked*⟩ are as follows:

- If an electrical cardiac activity (ECA) is interrupted by the artificial pacemaker operation (APO), the ⟨*event-chunked*⟩ is increased by one.
- If the ⟨*next event*⟩ of the ECA is the exact event as before the APO, the ECA is defined as paced ECA, which probably holds different duration and amplitude based on the declaration of associated parameters (see 5.2.1.5)
- If paced ECA is scheduled repeatedly after APO and the value of the ⟨*event-chunked*⟩ is over one, then the simulation of the ⟨*next event*⟩ (i.e. the paced ECA) will focus on maintaining presenting the rest of the paced ECA without starving to renew its duration or amplitude.

Next in the colour set **Timing Cycle** are the ⟨*ecg-duration*⟩ and ⟨*remain-duration*⟩, which also perform roles in maintaining accurate synchronization of the events. Similar to the ⟨*event-chunked*⟩, some of the pacemaker's operation modes, such as in ⟨AOO⟩, can pace a chamber while the heart is conducting another operation. Hence, the ⟨*event-chunked*⟩, ⟨*ecg-duration*⟩ and ⟨*remain-duration*⟩ all compound control the computing of the concurrent events and, when applicable, distinguish paced events as defined in Table 5.2. The rules regulating ⟨*ecg-duration*⟩ and y are described as follow:

- Upon the initialization of the model, the ⟨*ecg-duration*⟩ is calculated in accordance with the first event's duration and the lower rate limit (LRL) of the operation mode.

- If the initial value of ⟨*ecg-duration*⟩ is beyond the end time of the ⟨*LRL*⟩, then the ⟨*ecg-duration*⟩ is set to the ⟨*start time*⟩ of the ⟨*LRL*⟩, and the remaining value of ⟨*ecg-duration*⟩ is retained in ⟨*remain-duration*⟩. Otherwise, the value of ⟨*remain-duration*⟩ is set to *zero*.

- Afterwards, the ⟨*ecg-duration*⟩ of each electrical cardiac activity (ECA) is calculated during the conducting of the previous event of ECA, and any event with ⟨*ecg-duration*⟩ that exceeds the ⟨*LRL*⟩ will be forwarded to ⟨*remain-duration*⟩.

- After the conclusion of an artificial pacemaker operation (APO), if the ⟨*remain-duration*⟩ is greater than *zero*, then the ⟨*next event*⟩ will be the same as the ⟨*last event*⟩, and event's duration is retrieved from ⟨*remain-duration*⟩ (w.r.t. the above rules and paced event declarations)

Next in the colour set **Timing Cycle** is the ⟨*pacing-rate*⟩. It is calculated based on the assigned operation mode. In most modes, the ⟨*Low Rate Limit (LRL)*⟩ is the ⟨*pacing-rate*⟩, but in other modes where the ⟨*Rate modulation*⟩ (i.e. XXXR) is required, the ⟨*Maximum Sensor Rate (MSR)*⟩ is considered during calculation. The ⟨*pacing-rate*⟩ is determined according to the following conditions:

- If the selected operation mode is a single-chamber mode, then the value of the ⟨*pacing-rate*⟩ is fixed to the ⟨*LRL*⟩ of the desired chamber.

- Otherwise, in dual-chamber modes, the value of the ⟨*pacing-rate*⟩ is constantly updated as follows:
  - Set the ⟨*pacing-rate*⟩ to the ⟨*LRL*⟩ for the artificial pacemaker operation (APO) considering the atrial
  - After the fulfilment of the above rule, for APO regarding ventricular, set the ⟨*pacing-rate*⟩ to this equation:

$$LRL + AV\ delay \tag{5.29}$$

  - Recursively implement two rules above within each cardiac cycle till the model is terminated.
- If the applied operation mode requires advanced computation(as in ⟨*XXXR*⟩), then used the mode's associated parameter to accurately calculate the ⟨*pacing-rate*⟩ (see 5.2.1.5 for related parameters)

Next in the colour set **Timing Cycle** are the ⟨*current-pacing-occurrenc*⟩ and ⟨*next-pacing-duration*⟩. These two are employed to identify the start time and the estimated duration of the subsequent *pacing occurrence*. The equation of the ⟨*current-pacing-occurrenc*⟩ is as follows (c.f. Table 5.4)

$$pacing\ ID \times pacing\ rate \tag{5.30}$$

The ⟨*next-pacing-duration*⟩ is computed according to the following rules:

- If the operation mode is set to only pace the atrial, then the duration is the value of the parameter ⟨*atrial pulse width*⟩

- Likewise, if the operation mode is set to only pace the ventricular, then the duration is the value of the parameter ⟨*ventricular pulse width*⟩
- However, if the operation mode is a dual-chamber mode, then both of the two above rules are implemented subsequently, as described in the ⟨*pacing-rate*⟩.

Finally, the colour set of ⟨*timers*⟩ (defined in Table 5.4) represents the computed values of the assigned parameters explained in section 5.2.1.5.

5.2.1.5 The model variables and parameters

Unlike the naming style of the variables in the previous model (i.e. the CCS model as in 4.2.1.4), the variables' names in this model are abbreviated by their initial letters. Nevertheless, the functions of the variables in this model are equivalent to those in the previous model.

CPN variables are utilized in CP-net inscriptions and bound to different values from their colour set. The CPN variables of this model, their abbreviations, and their colour set are presented in Table 5.5.

Table 5.5: The CPN variables of the colour sets

Colour Set	CPN variable(s)	The used abbreviation(s)
REAL	*duration*	*d*
interval ratio	*get_interval, set_interval*	*gi, si*
ECG	*get_ecg, set_ecg*	*ge, se*
pacemaker parameters	*get_pacemaker, set_pacemaker*	*gs, ss*
timing cycle	*get_timing, set_timing*	*gt, st*

The reliable results of any model begin from assigning accurate values to its parameters. In this proposed model, parameters are employed for composing distinguished case studies, verify different requirements of the pacemaker system, and validate the pacemaker system toward complex case studies.

As in the Cardiac Conduction System (CCS) model in the previous chapter, parameters are classified according to their purposes. In this chapter, some parameters are inherited from the CCS model, and thereby the reader is referred to section 4.2.1.4 for more details.

The inherited parameters from the CCS model are as follows:

- Electrocardiogram Setup: Three parameters that control the simulated behaviour of the ECG machine.

- Heartbeat and Rhythmicity Rates: Four parameters define the heartbeat rate and the rhythmicity rates.
- Events Durations: For each cardiac event, a single parameter defines its duration.
- Waves Amplitudes: Also, for each cardiac event, a parameter represents the event's amplitude from 12 different perspectives by 12 leads.
- Frequency of Fluctuated Waves: For two distinct fluctuated waves, two parameters identify the frequency of each wave.
- Event Enablement: Parameters determine the reporting of cardiac events.
- Event Ratio: Parameters control the ratios of cardiac events.
- Dropped Beat: Three parameters allow the representation of dropped beats.
- Heart and Leads Axes: Various parameters enable the implication of estimating the events' amplitudes.
- Activate Leads: Parameters simulate the election of some leads during ECG.
- Precordial Lead Variables: Parameters assist in the further calculation for the precordial leads.

However, this model introduces other parameters as follows:

Pacemaker System

Table 5.6 represents the parameters that regulate the delivered therapy by the pacemaker system. The parameter $\langle mode \rangle$ sets the bradycardia operation mode of the pacemaker system. These modes were interpreted in section 2.4.3 and analyzed in section 5.3.

The parameter $\langle magnet\_test \rangle$ facilitates the implementation of the magnet test during the magnet state. Typically, the magnet test is applied to investigate the status of the pacemaker's battery. As in the actual magnet test, this parameter forces the model to adapt to specific settings concerning the pacemaker's operation mode and its rate as explained in Boston Scientific [2007].

Moreover, the value of the parameter $\langle battery\_status \rangle$ also influences the settings of the pacemaker automatically. The legal status levels of the battery include Begining of Life (BOL), Elective Replacement Near (ERN), Elective Replacement Time (ERT), and Elective Replacement Past (ERP). As defined in Boston Scientific [2007], the functional capabilities of the pacemaker are determined in accordance with each status level.

Table 5.6: Parameters for Pacemaker System

Parameter	Valid Value	Annotations
mode	*Off, DDDR, VDDR, DDIR, DOOR, VOOR, AOOR, VVIR, AAIR, DDD , VDD , DDI , DOO , VOO , AOO , VVI , AAI , VVT , AAT , OVO , OAO , ODO , OOO*	Bradycardia Operation modes are defined in Table 5.1
magnet_test	*true , false*	For Magnet State
battery_status	*BOL , ERN , ERT , ERP*	For Therapy Availability

Paced Waves

The parameters in Table 5.7 and 5.8 show the events that can be redefined as paced events (c.f. 4.2.1.4). In this model, paced events occur when a regular event is interrupted or occurred right after a pacemaker's pacing. While these parameters allow for different definitions of events, they can also be defined identically as the regular definitions if desired under the case study specifications.

Table 5.7: Parameters for Paced Waves Amplitude

Parameter	Valid Value	Annotations
PPa	{defult= $\mathbb{R}$, I= $\mathbb{R}$, II= $\mathbb{R}$, III= $\mathbb{R}$, aVR= $\mathbb{R}$, aVL= $\mathbb{R}$, aVF= $\mathbb{R}$, V1= $\mathbb{R}$, V2= $\mathbb{R}$, V3= $\mathbb{R}$, V4=$\mathbb{R}$, V5= $\mathbb{R}$, V6= $\mathbb{R}$ }	Paced P wave
PQa		Paced Q wave
PRa		Paced R wave
PSa		Paced S wave
PTa		Paced T wave
PUa		Paced U wave

Table 5.8: Parameters for Paced Waves Durations

Parameter	Valid Value	Annotations
PPwave	$(duration, range) \mid duration, range \in \mathbb{R}_{\geqslant 0}$	Paced P wave
PQwave		Paced Q wave
PRwave		Paced R wave
PSwave		Paced S wave
PTwave		Paced T wave
PUwave		Paced U wave

Pulses Widths and Amplitudes

In the model, pulses generated by the pacemaker are defined by their widths (i.e. durations) and amplitudes. As with cardiac events (c.f. 4.2.1.4), the pulse width (Table 5.9) determines its time period in milliseconds (ms), while the pulse amplitude (Table 5.10) determines its vertical dimension over time in volts (V).

Table 5.9: Parameters for Pulse Width

Parameter	Valid Value	Annotations
A_pulse_width	*(duration, range)* | *duration, range* $\in \mathbb{R}_{\geqslant 0}$	Atrial Pulse Width
V_pulse_width		Ventricular Pulse Width

Table 5.10: Parameters for Pulse Amplitude

Parameter	Valid Value	Annotations
A_pulse_amplitude	{defult= $\mathbb{R}$, I= $\mathbb{R}$, II= $\mathbb{R}$, III= $\mathbb{R}$, aVR= $\mathbb{R}$, aVL= $\mathbb{R}$, aVF= $\mathbb{R}$, V1= $\mathbb{R}$, V2= $\mathbb{R}$, V3= $\mathbb{R}$, V4=$\mathbb{R}$, V5= $\mathbb{R}$, V6= $\mathbb{R}$ }	Atrial Pulse Amplitude
V_pulse_amplitude		Ventricular Pulse Amplitude

Pulse Sensitivity

In addition to the parameters for pulses widths and amplitudes, Table 5.11 presents the parameters for the pulses sensitivities. The pulse sensitivity determines the minimum voltage resulting from the cardiac activity (i.e. atrial depolarization or ventricular depolarization) to be detected by the artificial pacemaker.

Table 5.11: Parameters for Pulse Sensitivity

Parameter	Valid Value	Annotations
A_sensitivity	$\mathbb{R}_{\geqslant 0}$	Atrial Pulse Sensitivity
V_sensitivity		Ventricular Pulse Sensitivity

Rate Limits

These parameters in Table 5.12 represent the rate limits of the artificial pacemaker. The maximum sensor rate (MSR), as the name suggests, defines the allowed pacing rate regulated by a sensor. The lower rate limit (LRL) defines the number of pace pulses generated per minute when cardiac activities are undetectable. In contrast, the upper rate limit (URL) defines the maximum rate coordinating between sensing the atrial and pacing the ventricular. If the hysteresis rate limit (HRL) is enabled, it allows cardiac events and applicable rates to fall below the LRL in order to maintain a spontaneous rate.

Table 5.12: Parameters for Rate Limits

Parameter	Valid Value	Annotations
MSR	$\mathbb{Z}_{\geqslant 0}$	Maximum Sensor Rate
LRL		Lower Rate Limit
URL		Upper Rate Limit
HRL		Hysteresis Rate Limit

Refractory Periods

In Table 5.13 parameters for refractory periods are given. The refractory periods in the pacemaker system ensure the sensed cardiac event is ignored within the designated chamber in order to eliminate false sensing. The ventricular refractory period (VRP) follows the ventricular events to prevent additional operations by the pacemaker system. Similarly, the atrial refractory period (ARP) occurs after the atrial events to prevent other pacemaker's operations. On the other hand, the post ventricular atrial refractory period (PVARP) follows the ventricular events and monitors the atrial events to impact the other pacing operations. The atrial and ventricular blanking periods follow the spontaneous cardiac events to limit possible crosstalk issues among the cardiac events and the pacemaker operations.

Table 5.13: Parameters for Refractory Periods

Parameter	Valid Value	Annotations
A_blank	$\mathbb{Z}_{\geqslant 0}$	Atrial Blanking Period
V_blank		Ventricular Blanking Period
ARP		Atrial Refractory Period
VRP		Ventricular Refractory Period
PVARP		Post Ventricular Atrial Refractory Period
PVARP_extension		Extend PVARP

Atrial-Ventricular (AV) Delay

The atrial-ventricular (AV) delay is the time period from the atrial event till the ventricular pace. In some operation modes, the AV delay leads to ventricular pacing if no ventricular event is sensed. Table 5.14 represents parameters related to AV delay. The fixed AV delay gives an absolute time for all the cardiac cycles, while the dynamic AV delay is defined for each cardiac cycle according to the duration of the prior cardiac cycle multiplied by a factor. The sensed AV delay offset shortens the AV delay following a sensed atrial event.

Table 5.14: Parameters for AV Delay

Parameter	Valid Value	Annotations
fixed_AV_delay	$\mathbb{R}_{\geqslant 0}$	Fixed AV Delay
MD_AV_delay		Minimum Dynamic AV Delay
dynamic_AV_delay_factor		Dynamic AV Delay Factor
sensed_AV_delay_offset		Sensed AV Delay Offset
dynamic_AV_delay	*true* , *false*	Dynamic AV Delay

Atrial Tachycardia Response (ATR)

The Atrial Tachycardia Response (ATR) parameters restrict the pacemaker system from pacing during atrial tachycardia. When the ATR mode is enabled and atrial tachycardia is detected, the number of ventricular cycles is counted during the ATR duration before the ATR fallback is initiated in accordance with the pacemaker system's specifications.

Table 5.15: Parameters for ATR

Parameter	Valid Value	Annotations
ATR_mode	*true* , *false*	ATR Mode
ATR_duration	$\mathbb{Z}_{\geqslant 0}$	ATR Duration
ATR_fallback		ATR Fallback

Rate-Adaptive Pacing

The rate-adaptive pacing parameters allow the pacemaker system to regulate the cardiac cycle following the metabolic need, which is decided through an accelerometer. The activity threshold identifies the level at which it should exceed before adjusting the pacemaker's rate. The response factor defines the pacing rate of different levels of activities. The reaction time defines the required time before considering MSR instead of LRL. In contrast, the recovery time is the required time before considering back the LRL instead of MSR. Finally, the rate smoothing limits the change of the pacing rate causes by abrupt changes in the rate.

Table 5.16: Parameters for Rate-Adaptive Pacing

Parameter	Valid Value	Annotations
activity_threshold	V-Low , Low , Med-Low , Med , Med-High , High , V-High	Activity Threshold
response_factor	$\mathbb{Z}_{\geqslant 0}$	Response Factor
reaction_time		Reaction Time
recovery_time		Recovery Time
rate_smoothing		Rate Smoothing

5.2.2 Atrial-Depolarization Submodel

The atrial-depolarization submodel, as the name suggests, demonstrates the depolarization activity that occurs within the atrial. The atrial depolarization can be observed in the ECG and reported as the P wave.

Unlike the atrial-depolarization submodel in the previous model (i.e. the CCS model), the atrial-depolarization submodel in this chapter is fairly compacted. As shown in Figure 5.2, this submodel is composed of only one transition connecting four distinguished places.

The processing of atrial depolarization, if enabled, is determined based on the computing of the tokens from all the four connected places explained in the previous submodel. The token of place ⟨*Timing Cycle*⟩ has the timestamps of a real number, which initially equals the value of the parameter ⟨*initial time*⟩, and after that equals the continuous cumulative timestamps of the model. The firing of transition ⟨*stimulate atrial*⟩ is only enabled if its guard, which represents appropriate rules, is holding. The specific rules that must be satisfied are as follows:

- The value of ⟨*next event*⟩ from the colour set **Timing Cycle**, defined in Table 5.4, equals ⟨*P-wave*⟩, ⟨*dropped-Pwave*⟩, or ⟨*paced-Pwave*⟩ defined in 5.2.
- If the value of the parameter ⟨*number of beats*⟩ is greater than zero, then the number of modelled full heartbeats must be less than or equal to the value of this parameter.
- Otherwise, if the value of the parameter ⟨*number of beats*⟩ equals zero, then the number of modelled full heartbeats has no impact on the transition.

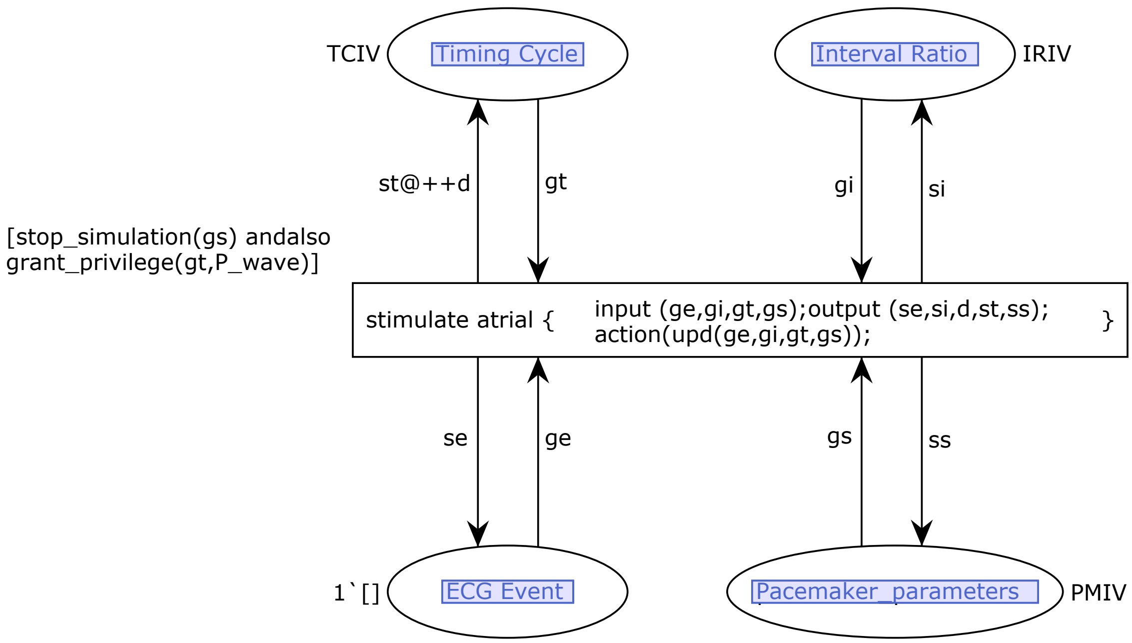


Figure 5.2: The Atrial-Depolarization Submodel

After firing transition ⟨*stimulate atrial*⟩, all colour tokens are negotiated according to the event's parameters, and they are returned to their original places. Consequently, the generated event from the behaviour of this transition is restricted to four possible cases as follows:

- The first case is that the event is not enabled, and thereby, all original colour tokens are returned with no further computations.
- The second case is the ordinary ⟨*P-wave*⟩ event where the colour tokens are updated accordingly.
- The third case is the ⟨*dropped-Pwave*⟩, when the event's interval is due. This event is reported with the duration of a regular ⟨*P-wave*⟩ but holds zero amplitude.

- The fourth case is the ⟨*paced-Pwave*⟩, where the original duration and amplitude of the regular ⟨*P-wave*⟩ are redefined according to the associated parameters of the paced events. This event typically follows the event of ⟨*artificial-pacing*⟩.

5.2.3 AV-Node Submodel

The AV-Node submodel (Figure 5.3) represents the event where the action potentials spread through the AV node towards the ventricular. Like the previous submodel, the transition ⟨*present PR segment*⟩ is enabled if and only if the value of ⟨*next event*⟩ from the colour set **Timing Cycle** equals ⟨*PR-segment*⟩, ⟨*dropped-PRsegment*⟩, or ⟨*paced-PRsegment*⟩.

Upon the firing of transition ⟨*present PR segment*⟩, all tokens from all connected places are concurrently calculated by their defined parameters and then returned to their original places with cases similar to the previous submodel.

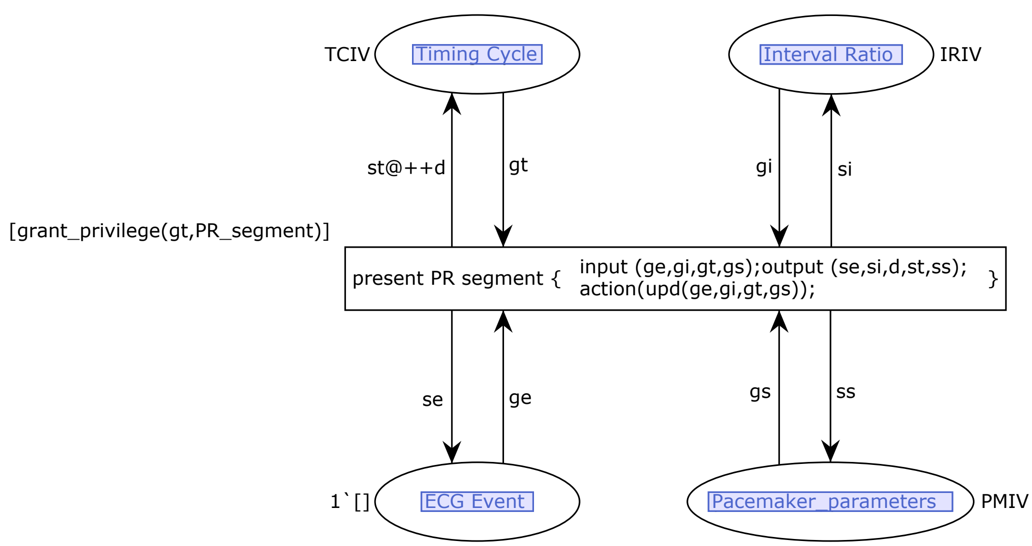


Figure 5.3: The VA-Node Submodel

5.2.4 Ventricular-Depolarization Submodel

As shown in Figure 5.4, the Ventricular-Depolarization submodel, as the name suggests, addresses the ventricular depolarization event. This event, when examined using ECG, can be recognized as the QRS complex. The QRS complex is composed of three waves; ⟨*Q-wave*⟩, ⟨*R-wave*⟩, and ⟨*S-wave*⟩. Each wave reflects a position within the ventricular where the depolarization event occurred.

Accordingly, this submodel is composed of three transitions interpreting the waves of the QRS complex. Each transition is enabled and fired when its guard is fulfilled according to the value of ⟨*next event*⟩ from the colour set **Pacemaker Parameters**. Consequently, the tokens of connected places are appropriately updated.

Similar to other transitions in the previous submodels, each transition in this submodel is enabled if its assigned event is triggered. The event can be defined as regular, dropped or paced, and each event is subject to the values of associated parameters.

The output of each transition is as follows:

- The event is not enabled, resulting in all colour tokens returned unprocessed.
- The event is produced as regular because of the values of the associated parameters.
- The event is dropped since its interval is scheduled, and only the duration of the regular event is delivered with zero amplitude.
- The event is paced as it happened just after the artificial pacing, and the event duration and amplitude may be extended based on the parameters' values.

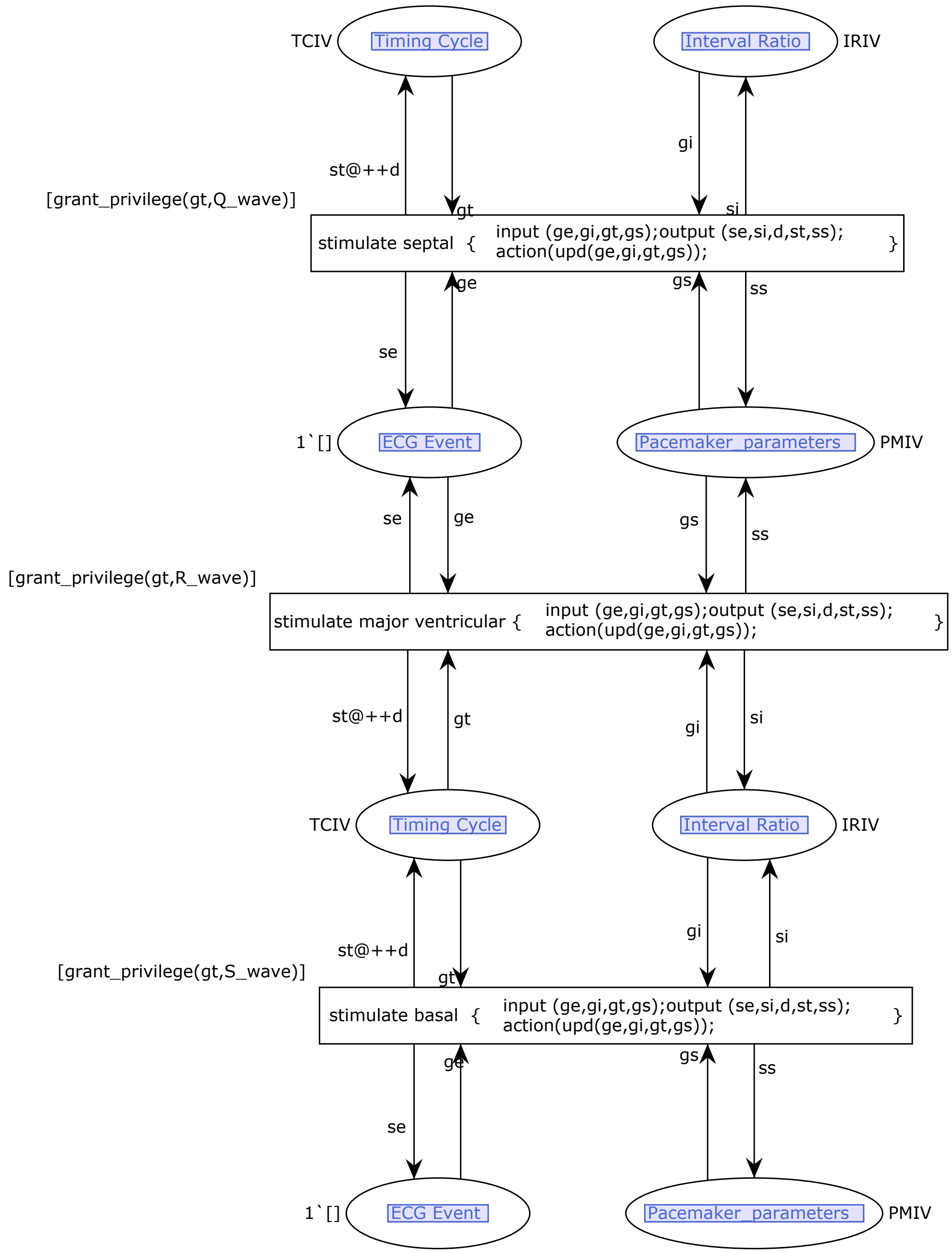


Figure 5.4: The Ventricular-Depolarization Submodel

5.2.5 ST-Segment Submodel

In Figure 5.5, the ST-segment submodel, as the name suggests, denotes the ⟨*ST-segment*⟩ of the ECG event. The ⟨*ST-segment*⟩ starts after completing the ventricular depolarization (i.e. after the ⟨*S-wave*⟩) and ends before the ventricular repolarization (i.e. before the ⟨*T-wave*⟩). This submodel follows the implementation of the previous submodels (i.e. when the transition ⟨*present ST segment*⟩ is enabled and fired, colour tokens are computed as as previously demonstrated.)

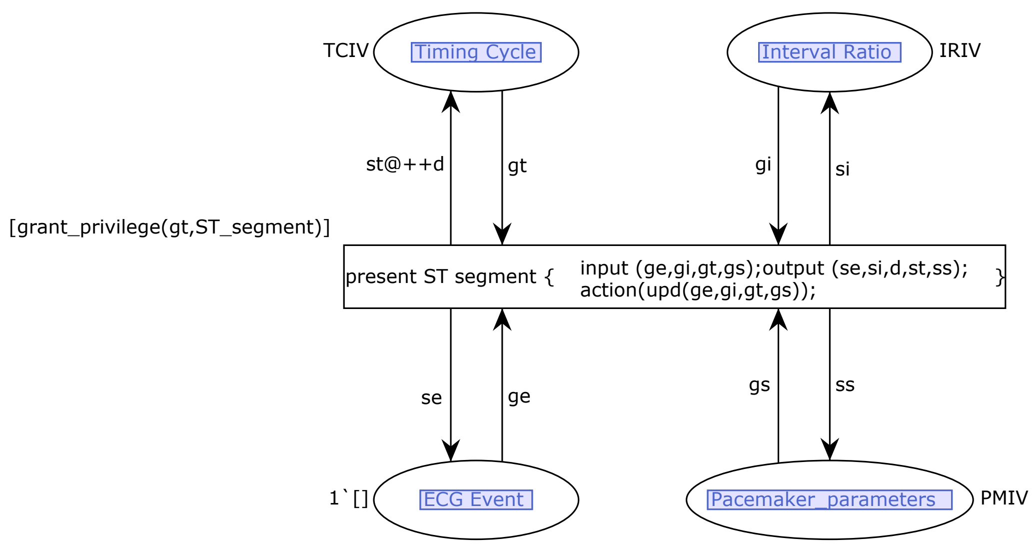


Figure 5.5: The ST-Segment Submodel

5.2.6 Ventricular-Repolarization Submodel

This submodel (Figure 5.6) demonstrates the event of the ventricular repolarization, which can be typically seen through the ECG as the ⟨*T-wave*⟩, and some cases include ⟨*TU-segment*⟩ and ⟨*U-wave*⟩. The segment connecting the end of the ventricular repolarization to the atrial depolarization is called the ⟨*TP-segment*⟩.

The behaviours of the transitions of $\langle$*T-wave*$\rangle$, $\langle$*TU-segment*$\rangle$ and $\langle$*U-wave*$\rangle$ are consistent with the previously explained transitions in previous submodels. The colour tokens are computed via firing fulfilled transitions in accordance with the related parameters' values, as discussed in the core-elements submodel.

However, the transition $\langle$*full heartbeat*$\rangle$ is enabled if and only if the next event is $\langle$*TP-segment*$\rangle$ or $\langle$*paced-TPsegment*$\rangle$ (c.f the $\langle$*TP segment*$\rangle$ does not have a definition as a dropped event). Therefore, the potential two cases produced as a result of firing this transition are as follows:

- The first case is a regular $\langle$*TP-segment*$\rangle$ sponsored by the values of the associated parameters.
- The second case is a $\langle$*paced-TPsegment*$\rangle$ also sponsored by the related parameters, and it occurred as a result of following the event of $\langle$*artificial-pacing*$\rangle$.

While the $\langle$*TP-segment*$\rangle$ holds no interval settings, this event controls the interval of the complete heartbeat cycle (i.e. during the firing of its transition $\langle$*full heartbeats*$\rangle$, the schedules of heartbeats' intervals are checked and applied if due as explained in sections 4.2.1.4). In principle, the cases of the cardiac cycle after firing the transition $\langle$*full heartbeat*$\rangle$ are as follows:

- The next cardiac cycle is processed in a similar implementation as in the preceding cycle.
- The next cardiac cycle is dropped with a defined duration that equals the preceding cycle
- The next cardiac cycle is also dropped but with a defined duration that declares to a different duration.

- The next cardiac cycle is halt since the model was terminated by reaching the desired state.

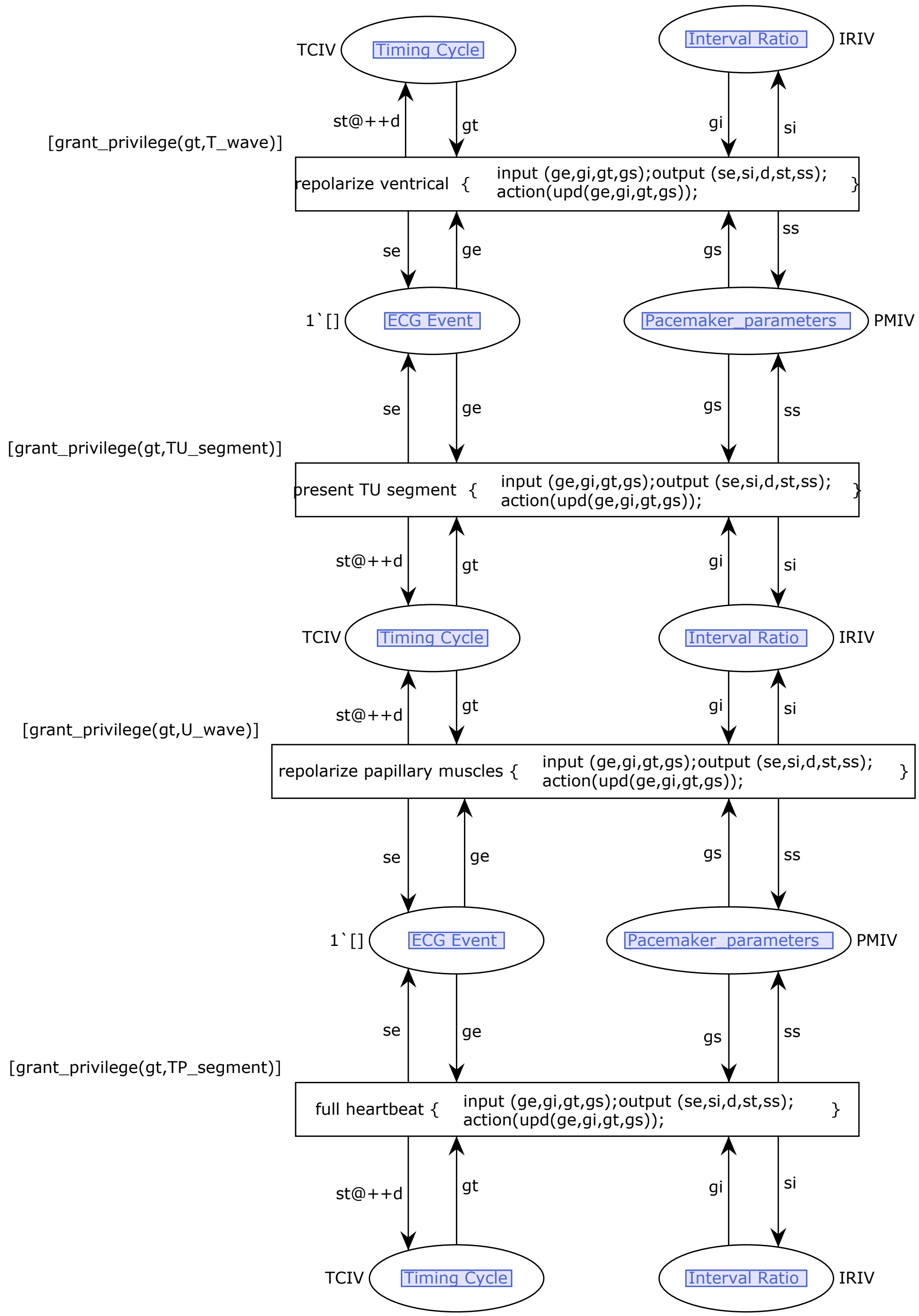


Figure 5.6: The Ventricular-Repolarization Submodel

5.2.7 Artificial-Pacemaker Submodel

This last submodel (Figure 5.7) signifies the artificial pacemaker's functions and operations. While this submodel's structure is comparable to the previous submodels (i.e. four places connected by a transition), this submodel's transition and rules are different.

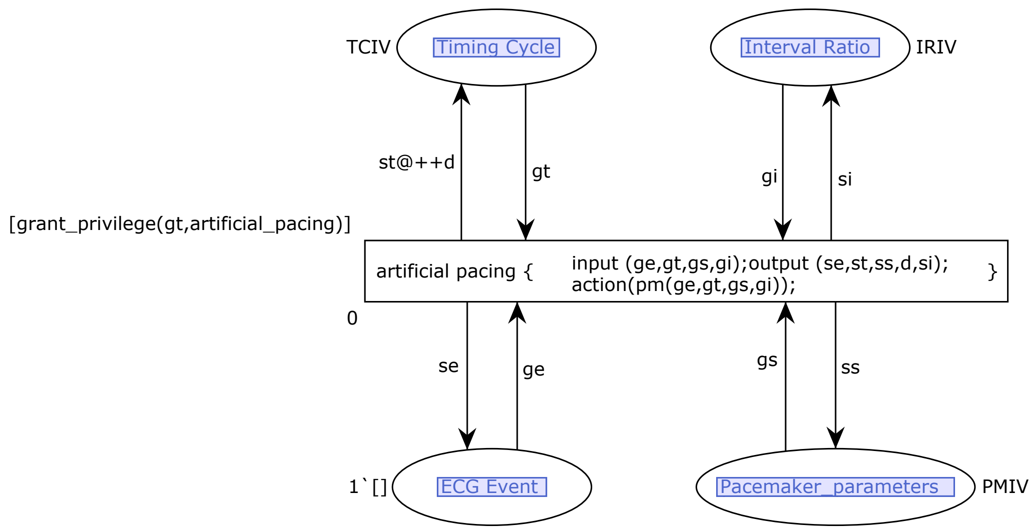


Figure 5.7: The Artificial-Pacemaker Submodel

The firing of transition ⟨*artificial pacing*⟩ is only enabled if its guard, which considers relevant rules, is holding. The specific rules that must be satisfied are as follows:

- The value of parameter ⟨*mode*⟩, defined in Table 5.6, indicates chamber pacing as in *AXXX, VXXX,* or *DXXX.*

- The value of ⟨*next event*⟩ from the colour set **Timing Cycle**, defined in Table 5.4, equals ⟨*artifical-pacing*⟩.

- The current time of the model equals the calculation of ⟨*pacing rate*⟩, defined in Table 5.4.
- The associated ⟨*timers*⟩ of the applied ⟨*operation mode*⟩ equal their assigned values.

After firing transition ⟨*artificial pacing*⟩, colour tokens are computed accordingly, and then associated places are updated with the processed colour tokens. The defined cases carried by the colour tokens of the connected places are as follows:

- The colour token of the place ⟨*ECG Events*⟩ (see 5.2.1.2) is updated in accordance with the applied operation mode, as follows:
  - If the operation mode is set to pace only *atrial*, then the colour token is updated with the definition of the set ⟨*atrial-pacing*⟩ according to the values of associated parameters.
  - Similarly, if the operation mode is set to pace only *ventricular*, then the colour token is updated with the definition of the set ⟨*ventricular-pacing*⟩ also according to the values of associated parameters.
  - If the operation mode is set to pace both of the *atrial* and *ventricular*, and both chambers are due concurrently based on the parameters, then the colour token is updated with both sets. Otherwise, if chambers are due sequentially, then the colour token is updated sequentially with the currently scheduled event set.
- The colour token of the place ⟨*Interval Ratio*⟩ (see 5.2.1.3) is returned as its original state with no further processing since the current specifications of the

pacemaker system demand no intervals. Nevertheless, this place was included in this submodel for four main reasons:

- This colour token is required by other functions of other colour sets.
- Avoiding likely redundant definitions of colour sets in order to satisfy two systems (i.e. CCS and CPS) that actually have a high degree of similarity.
- Keeping the structures and definitions of submodels consistent with each other.
- Establishing and facilitating the necessary definitions of the pacemaker system if new operations or case studies required the theories of intervals.

- The colour token of the place ⟨*Timing Cycle*⟩ (see 5.2.1.4) is calculated as follows:
  - The ⟨*pacing ID*⟩ is increased by one.
  - The ⟨*current pacing occurrence*⟩ is calculated by multiplying the new ⟨*pacing ID*⟩ by the proper ⟨*pacing rate*⟩.
  - The ⟨*last pacing occurrence*⟩ is set as the currently consumed ⟨*current pacing occurrence*⟩.
  - The ⟨*next event*⟩ is assigned to the previous event if it was interrupted or to the next event of the cardiac activities.
  - The other components are preserved or retained from their initial parameters' values.
- The colour token of the place ⟨*Pacemaker parameters*⟩ (see 5.2.1.1) is updated as follows:

– In the single-chamber modes, the $\langle$*last paced lead*$\rangle$ and $\langle$*last sensed lead*$\rangle$ maintain their values.

– However, in dual-chambers modes, the values of the $\langle$*last paced lead*$\rangle$ and $\langle$*last sensed lead*$\rangle$ are updated to the value of the other chamber (i.e. from atrial to ventricular and vice versa).

– The other components are preserved or retained from their initial parameters' values.

5.3 Model Analysis

The analysis of this proposed model is conducted by both simulation-based performance analysis and State Space analysis (aka reachability analysis). These two analyses were introduced in section 3.4. In principle, in the simulation-based performance analysis, the model is frequently executed to conduct various simulations. At the same time, its data is being collected and compared. Typically, the simulation-based performance analysis provides valuable insights into the model. However, it cannot prove the execution of all desirable states of the model. The reachability analysis is thereby conducted. By utilizing a computer tool, the outcome of the reachability analysis includes a graph and a report. The graph shows nodes rendering reachable markings and arcs showing enabled transitions of those markings. On the other hand, the report presents standard properties of the model, such as boundedness properties, home properties, liveness properties, and fairness properties.

In the analysis of this proposed model, the pacemaker system is verified and validated. During the verification, definite components of the system are evaluated to determine satisfaction with certain specifications. In comparison, system validation is the evaluation of the complete system in accordance with its requirements and obligations [IEEE Guide [2011] and CMMI Product Team [2002]].

In section 5.3.1, the bradycardia operation modes of the pacemaker are verified with their related parameters and functions, while in section 5.3.2, the pacemaker system is validated with different cardiac rhythms.

5.3.1 Bradycardia-Operation-Mode Analysis

This analysis represents the verification of the cardiac pacemaker system. The bradycardia operating modes (BOMs) affect the functions of the pacemaker system. Different BOMs interact with specific parameters (see 5.2.1.5, 6.5 and 6.5). Each BOM was analyzed for *500 heartbeats* by both simulation-based performance analysis and State Space analysis. Typically, the conducted analysis methods result in detailed technical reports that can count for numerous pages. However, the essential functions of the DOMs and most significant insights are summarized as follows:

Table 5.17: Bradycardia-Operation-Mode Analysis

Operation Mode	Essential Functions and Important Insights
Off	As the name suggests, the pacemaker is disabled in this mode. As expected, in this mode, the transition $\langle$*artificial pacing*$\rangle$ in the artificial-pacemaker submodel is reported as a $\langle$*dead transition*$\rangle$, which indicates the pacemaker never paced through the complete analysis.
OOO	This mode is temporary, and it indicates no pacing nor sensing capabilities. Similar to the previous mode, the transition $\langle$*artificial pacing*$\rangle$ in the artificial-pacemaker submodel is a $\langle$*dead transition*$\rangle$.
OAO	This mode is also temporary, and it signifies that only the atrium is sensed at a consistent time following the *low rate limit (LRL)* without any pacing capabilities. Accordingly, the conducted analysis reported steady sensing in the atrium at the defined rate.

. . . continued

. . . continued

Operation Mode	Essential Functions and Important Insights
OVO	Similar to the previous mode, this mode implies that only the ventricle is sensed consistently per the *low rate limit (LRL)* without any pacing capabilities. Subsequently, the conducted analysis proves steady sensing in the ventricle following the specified rate.
ODO	This mode combines the functions of the two above modes as both the atrium and ventricle are sensed following the *LRL* and the *AV delay* without any pacing capabilities. Consequently, the reported results show consecutive sensing in both the atrium and ventricle, as declared by the applicable rates.
AOO	This mode is asynchronous as only the atrium receives a pacing stimulus from the pulse generator at a fixed rate without any sensing capabilities. The reported results matched the expected behaviour. The pacemaker system in the mode paces following the defined rate.

. . . continued

. . . continued

Operation Mode	Essential Functions and Important Insights
VOO	Like the previous mode, this mode is asynchronous as only the ventricle receives a pacing stimulus at a fixed rate without any sensing capabilities. The analysis results show that the pacemaker system paces following the defined rate.
DOO	This mode joins the functions of the above two modes. Hence, each of the atrium and ventricle receives a pacing stimulus independently at different fixed rates without any pacing capabilities. The recorded results demonstrate the consistent pacing at the fixed rates.
AAI	In this mode, the atrium is sensed consistently and only paced whenever a spontaneous atrial depolarization is undetected. Throughout distinct ratios for the ⟨*P wave*⟩, which indicates the atrial depolarization, the reported results show artificial stimulus by the pacemaker following each ⟨*dropped Pwave*⟩ (i.e., the absence of the atrial depolarization).

. . . continued

. . . continued

Operation Mode	Essential Functions and Important Insights
AAT	In contrast to the previous mode, in this mode, the atrium is sensed and consequently paced if and only if a spontaneous atrial depolarization is detected. The related results reveal the pacemaker system consistently paced the atrium whenever an atrial depolarization occurred, indicated by the event $\langle P\ wave \rangle$.
VVI	This mode represents the sensing of the ventricle if and only if pacing during the absence of spontaneous ventricular depolarization. The results confirm the demand pacing of the ventricle when the *QRS complex* is undetected.
VVT	In comparision to the above mode, this mode indicates that the ventricle is sensed and therefore paced if a spontaneous ventricular depolarization is discovered. The results report steady pacing in the ventricle following the *QRS complex*.

. . . continued

. . . continued

Operation Mode	Essential Functions and Important Insights
VDD	This mode is synchronous. It implies that both the atrium and ventricle are sensed, but only the ventricle is paced if a spontaneous ventricular depolarization is undetected after a defined delay. By setting various ratios for the associated events, the results confirm that the ventricular pacing occurred after following specific timers if and only if the spontaneous ventricular depolarization dropped.
DDD	This mode is also synchronous. It indicates that both the atrium and ventricle are sensed. Therefore, if a spontaneous atrial depolarization is undetected within a defined timer, the atrium is paced. Likewise, if a spontaneous ventricular depolarization is undetected within a defined timer, the ventricle is paced. The reported results satisfy the described functions of this mode (i.e. the pacing occurs only during the absence of the desired deplorations)

. . . continued

. . . continued

Operation Mode	Essential Functions and Important Insights
DDI	In this synchronous mode, the atrium and ventricle are sensed. The detection of a spontaneous atrial depolarization inhibits the atrium's pacing without influence on the ventricle's pacing. However, the detection of a spontaneous ventricular depolarization inhibits the pacing capabilities for both the atrium and ventricle. The reported results of various event ratios show that the model correctly fulfills the functions of this mode, as explained earlier.
AOOR, VOOR, DOOR, AAIR, VVIR, VDDR, DDIR, DDDR	The functions of these modes are related to the above modes. Suppose a mode holds the assigned rate-adaptive pacing, i.e. XXXR. In that case, it indicates the pacing capabilities are subject to a connected sensor controlling the heart rate according to the stakeholder's activities. The analysis of these modes demanded more settings for the related parameters. Ultimately, the model satisfies the functions of these modes in accordance with the associated timers and parameters.

5.3.2 Cardiac-Rhythm Analysis

This analysis represents the validation of the cardiac pacemaker system via evaluating various operation modes against selected cardiac rhythms. Moreover, this analysis can assist in predicting the results of different new settings.

The analysis of four scenarios is presented. Each scenario has a defined cardiac rhythm and a specific operation mode. The parameters are set to the norm values (see 5.2.1.5, 6.5 and 6.5) of the defined cardiac rhythm and operation mode.

The calculation of timed State Space can be complex and laborious in part since the size of the reachability graph may be infinite as various timed markings with global clocks and timestamps are distinctive. Therefore, each scenario was analyzed for a defined number of heartbeats to restrict the infinite markings due to the global clock.

Table 5.18 presents these selected scenarios as follow:

Table 5.18: Scenarios for Cardiac-Rhythm Analysis

Operation Mode	Total of Heartbeats	Cardiac Rhythm
OAO	1000	Normal Sinus Rhythm
DOO	2000	Sinus Bradycardia
VDD	3000	First-Degree Heart Block
DDDR	6500	Junctional Rhythm

For all scenarios, the reported statistical results hold similar features. As shown in Table 5.19, the numbers of nodes and arcs in the SCC graph are always identical to State Space's corresponding numbers. As expected, this implies that the model has no cyclical behaviour. Even though the numbers of nodes and arcs are equal for both State Space and SCC graph in this proposed model, these numbers may not always agree based on the model's specifications. When the number of SCC-graph nodes is fewer than the number of State-Space nodes, there are non-trivial SCCs and cycles in the State Space of the model (i.e. may not terminate).

Table 5.19: CPN Tools state space report

Statistics					
Operation Mode		OAO	DOO	VDD	DDDR
Number of beats		1000	2000	3000	6500
Occurrence Graph	Arcs	1000	2004	3000	6504
	Nodes	1001	2005	3001	6505
	Status	Full	Full	Full	Full
Scc Graph	Nodes	1001	2005	3001	6505
	Arcs	1000	2004	3000	6504
Dead Markings		[1001]	[2005]	[3001]	[6505]
Dead Transition Instances		[1][a]	None	[1][b]	None
Live Transition Instances		None	None	None	None

[a]Artificial Pacemaker(artificial pacing)
[b]Artificial Pacemaker(artificial pacing)

The reachability properties' results indicate that an occurrence sequence exists in a sequential direction. By performing standard query functions, it was observed that the occurrence sequence, in all scenarios, returns true only between nodes in ascending order. This behaviour is anticipated since the model is time-based. The boundedness properties report the number of tokens for places after executing all reachable markings. In all scenarios, the quantity of one token was reported as the maximal and minimal number of tokens that can remain, in any reachable marking, on each place. Similar to reachability properties' results, the boundedness properties' results matched the assumed numbers in accordance with the timed characteristics of this model. The home properties' results, by design, indicate no home marking that is reachable from any other marking.

The liveness properties' results address *dead markings*, *dead transition instances*, and *live transition instances*. *Dead markings* are those with unenabled binding elements, which result from nodes holding tokens with no outgoing arcs or disabled transitions. For all scenarios, a single dead marking was reported. This is typically expected with models that should terminate at a certain point by design.

The results of *dead transition instances* indicate the transitions that were never enabled (i.e. fired) during the model's execution. When the results of *dead transition instances* are reported as *None*, this implies that all the transitions in the model have the opportunity of firing (occurring) at least once. According to the adopted operation modes and parameters, two scenarios hold no dead transition instances. In comparison, each of the other two scenarios registers a single dead transition instance. This result is interpreted as indicating that the pacemaker did not deliver any electrical impulses following the designated parameters.

On the other hand, the results of *live transition instances* state the transitions that are reachable in the occurrence sequence of any reachable marking. All four scenarios contain no *live transition instances*, thereby determining that each transition is not always found in the occurrence sequence of any reachable marking. Models with dead markings regularly have no *live transition instances* because the model was terminated. Dead transitions are not the opposite of live transitions since a non-dead transition must be enabled at least once while a live transition should continue to be enabled.

Finally, the fairness properties' results represent a list of all impartial transitions which occur infinitely. When such transitions are removed or restricted, all infinite occurrence sequences of the model will be subsequently eliminated if desired. As expected, in accordance with the parameters and specifications, there are no infinite occurrence sequences in any scenario.

Appendix D presents the analysis report for the second scenario. Since the analysis reports of the other scenarios, as interpreted in this section, are similar but lengthy, no more appendixes were appended.

CHAPTER

6

CONCLUSIONS

This chapter concludes this research. Section 6.1 discusses the chosen approach of this research. Section 6.2 demonstrates the contributions of the thesis, Section 6.3 lists related publications, and section 6.4 addresses the future work.

6.1 Discussion

The utilize of TPCNs to model a complex system, such as the pacemaker system, and the adopted approach in this study reveals some merits and demerits that deserve further discussion. Besides the already addressed merits in the previous chapters, the other merits of selecting TCPNs with the adopted approach are as follows:

- The employing of TCPNs, as a modelling language in this approach, offer intuitive graphical formalisms that empower various disciplines, such as medical research, to observe and study the behaviour of the CPS safely and efficiently. Unlike some other graphical modelling languages, in this TCPN-based approach, only three notations were utilized; places for the models' states, transitions for the models' functions, and arcs that guide the flow of the models.

- While this approach offers uncomplicated representations of its models, all the TCPN-based models are, by default, based on mathematical definitions, which maintain the formal analysis of the structural and behavioural properties of the developed models.

- Since this approach adopted the TCPNs, the proposed models properly support the scalability of the system. For example, the cardiac conduction system can be further expanded to include the functions of the cardiac tissues and other cardiac functions.

- In this approach, besides examining all the operating modes with various parameters, it also examines various realistic synthetic events of the cardiac electrical activities, which sponsors the analysis of the verification and validation of the pacemaker system. This approach is also valuable for industrial usage to guide the early-stage development of the pacemaker system.

- The analysis of this approach includes the technique of the State Space analysis, which can be carried automatically upon the request of the users. This analysis reports the standard behavioural properties of the models, which empower different stakeholders, with minimum training on the tools and reading the report, to work with the model with few obstacles.

- In addition, TCPNs are extended from the Petri Nets (PNs), which are supported by a large community and several tools. While TCPNs maintain valuable properties of PNs, TCPNs combine the capabilities of PNs with the capabilities of a high-level programming language (i.e. The Standard ML (SML/NJ implementation)). This combination allows the adopted approach to adequately introduce and analyze the pacemaker system and its operational environment.

- While several code-generation tools were proposed for different PNs extensions, the adopted CPN Tools in this approach offer no direct tool for the code-generation process. Nevertheless, since the coloured sets and the CPN ML language are founded on the functional programming Language (i.e. The Standard ML), the models can be redefined applying the set theory and the Standard ML. Then, the output may allow for further translation into another programming language.

- During the development of the proposed models, it can be concluded that the adjustment of TCPN-based models is straightforward. In the adopted CPN Tools, syntax errors are identified instantly, which assists in limiting syntax errors during the simulation analysis and efficiently adjusting the model as required.

- The approach of this research employed the capability of the adopted CPN Tools to import and export different file formats during the execution of the models. This not only facilitates the integration of the proposed models with other applications but also allows for the construction of customizable datasets from the executed models.

On the other hand, the adopted approach in this study and TCPNs hold some demerits, which require further investigation and development. These demerits include the following:

- In order to reduce the complexity of the proposed models, one approach was to limit the number of places and transitions without a negative impact on the models. Consequently, some practical resolutions include transition guards and functions. However, these transition guards and functions usually compact some interval process of the models. While this appears to be elegant, some intervals of the models may then only reveal the beginnings and endings of events following their rules and parameters.

- In the approach, the time was represented using the holding duration technique in TCPNs, where only available coloured tokens enable transitions. This technique turns the modelling of the pacemaker system into a highly challenging task for two reasons; the entire pacemaker depends on time and timers, and timers are interconnected among several operating modes. Nevertheless, one method to defeat this challenge was to pre-process the events of the pacemaker system as explained in section 5.2.

- While this approach depends on the graphical notations by TCPNs, visual inspection of tests, such as the electrocardiogram (ECG), is common when examining the pacemaker system. The adopted CPN tools, however, support Java-based applications to be integrated with the TCPN-based model, which then can offer a more traditional visualization of the models. Nevertheless, the adoption of other applications must be independently verified to ensure accurate visualization of the model's data before assessing final results.

- CPN Tools, as an elected modelling tool for this approach, offer instant syntax checking, but the provided error messages are usually poor. Consequently, debugging in CPN Tools can be time-consuming and labour-intensive. For example, if a function is missing a comma in one of its lines, the editor of CPN Tools will flag the entire function without specifying the line of the missing comma. Some recommendations here include the improvement of the current editor of CPN Tools or the development of a new editor of CPN Tools.

- Another concern with the adopted approach of breaking models into several submodels is that many submodels look alike. However, each submodel represents an independent function, and therefore, an independent submodel is justifiable. Nevertheless, while it is possible to compact all submodels into a single model, the overall readability and legibility of the models will extremely suffer as a result of this compressed design. On one hand, the models must reflect the specifications of the desired system. On the other hand, the complexity of the models should not exceed their real system.

- Finally, while TCPNs is ordinarily suitable for modelling biological systems, the number of available TCPN-based biological models is limited. Since many biological systems hold precise established specifications, a leading project to sponsor and provide open-source TCPN-based models for common biological systems will empower and promote more studies to adopt TCPNs for related systems.

6.2 Highlights of the Contributions

The contributions of this study include the following:

- **Implementing a formalization of the pacemaker system and its environment.** Formal methods provide a practical and effective approach to verify and validate complex systems. This thesis formalizes the pacemaker system and the cardiac conduction system, which presents realistic synthetic events of the heart's cardiac electrical activities to assist in validating the pacemaker system. While the proposed models in this work are formal and comprehensive, they are also flexible enough to adapt and analyze different algorithms and rules. These models may eventually evolve into standard formal models of the pacemaker system.

- **Proposing a mathematical approach to model and analyze the pacemaker system.** The modelling of the pacemaker system is, by nature, complex and challenging. Designing the pacemaker with TCPN-based models indicates the representation of the real behaviour of the pacemaker in mathematical terms. Consequently, these proposed models empower mathematical reasoning of both the consequences of current behaviours and the accurately predicted outcomes of new behaviours with new settings. These new settings can then be initially modelled and analyzed with these proposed models, which contributes to safety efficiency and cost reduction.

- **Providing an approach of adopting principles of software design when modelling complex systems.** Like other modelling languages, Timed Coloured Petri Nets (TCPNs) combines the expressive power in describing systems' behaviours and the mathematical capabilities to prove models' properties. Nevertheless, complex systems require a proper approach in order to reach the desired results. This study demonstrates a proper approach to adopt TCPNs when modelling a complex system, such as the pacemaker system. It also shows an appropriate implementation of some design principles, such as modularity and abstraction, within TCPNs.

6.3 Related Publications

1. Mohammed Assiri and Ryszard Janicki. Modelling and analyzing electrocardiogram events using timed coloured petri nets. In Michael Köhler-Bußmeier and Ekkart Kindler and Heiko Rölke, editors, *Proceedings of the International Workshop on Petri Nets and Software Engineering co-located with 41st International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2020), Paris, France, June 24, 2020 (due to COVID-19: virtual conference),* volume 2651 of CEUR Workshop Proceedings, pages 222–223. CEUR-WS.org, 2020.

2. Mohammed Assiri and Ryszard Janicki. Towards Modelling of Cardiac Pacemakers with Timed Coloured Petri Nets and Related Tools. In Michael Köhler-Bußmeier and Ekkart Kindler and Heiko Rölke, editors, *Proceedings of the International Workshop on Petri Nets and Software Engineering co-located with 42st International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2021), Paris, France, June 25, 2021 (due to COVID-19: virtual conference),* volume 2907 of CEUR Workshop Proceedings, pages 1–20. CEUR-WS.org, 2021.

6.4 Future Work

In this thesis, the presented work can be further extended in several directions, including theory, applications, and tools (among others).

Theory

While the proposed model is based on the predefined modelling language, Timed Coloured Petri Nets (TCPNs), the definition of TCPNs can be extended to include proper definitions for interval orders. The current model involves the beginnings and endings of events following their rules and parameters. However, as presented in [Alqarni [2016]], the definitions of TCPNs can be further extended with interval processes.

Applications

While this study covers the representation of many cardiac rhythms, a few highly abnormal rhythms limit the model capabilities of comprehensive representation of all possible rhythms. Therefore, some extensions to the model parameters are recommended.

In addition, the model supports the random distribution of event durations between two declared values. Nevertheless, the values of event amplitude are limitedly defined as either presented, dropped, or paced. Like the event durations, the event amplitudes can also be redefined to promote the random distribution between two declared values.

Tools and Automation

While some external tools can visualize the produced cardiac events by this study, a visualization tool can be developed and embedded into the CPN Tools [AIS Group [2013]], which support Java-based applications. This tool will allow real-time and visual observation of the model execution.

In addition, this proposed model can be further developed into a web-based model, which CPN Tools [AIS Group [2013]] partially support, to provide more accessibility to a wide range of stakeholders. As well, tools such as [Simonsen [2014]], [Carlsson [2018]], or [Gomes and Barros [2001]] can be properly adopted with the development of the new web-based version to empower code generation for simulated cases.

6.5 Closing Remarks

Software verification is an ongoing challenge. More studies and efforts in this area will eventually assist in the development of more reliable software. As a pilot problem for the Verified Software Initiative, modelling the pacemaker system through different formalisms will not only lead to a more concrete understanding of current tools but will also drive more comprehensive comparison among them. This study demonstrates the expressive power and reliability of TCPNs towards modelling safety-critical systems such as the pacemaker system. Nevertheless, the number of CPN-based models concerning the verification of the pacemaker system is limited and insufficient. Consequently, this study aimed to contribute to the verification and validation of the pacemaker system and to present a practical methodology when approaching similar systems. The model is formed of several submodels to augment its legibility and adaptability. In addition, the model can assist not only in facilitating a better understanding of pacemakers and relevant cardiac events but also in producing customizable data to judge and optimize different relevant algorithms and techniques sufficiently.

APPENDIX A: PROGRAMMABLE PARAMETERS

These programmable parameters in Table [1] are considered the boundaries when operating the pacemaker in a specific mode.

The programmable parameters requirements

Parameter	Programmable Values	Increment	Nominal	Tolerance
Mode	Off DDD VDD DDI DOO AOO AAI VOO VVI AAT VVT DDDR VDDR DDIR DOOR AOOR AAIR VOOR VVIR	—	DDD	—
Upper Rate Limit	50-175 ppm	5 ppm	120 ppm	±8 8 ms
Lower Rate Limit	30-50 ppm 50-90 ppm 90-175 ppm	5 ppm 1 ppm 5 ppm	60 ppm	±8 ms
Maximum Sensor Rate	50-175 ppm	5 ppm	120 ppm	±4 ms
Fixed AV Delay	70-300 ms	10 ms	150 ms	±8 ms

. . . continued

. . . continued

Parameter	Programmable Values	Increment	Nominal	Tolerance
Dynamic AV Delay	Off, On	—	Off	—
Minimum Dynamic AV Delay	30-100 ms	10 ms	50 ms	
Sensed AV Delay Offset	Off, -10 to -100 ms	-10 ms	Off	±1 ms
A or V Pulse Amplitude Regulated	Off, 0.5-3.2V 3.5-7.0 V	0.1V 0.5V	3.5V	±12%
A or V Pulse Amplitude Unregulated	Off, 1.25, 2.5, 3.75, 5.0V	—	3.75V	—
A or V Pulse Width	0.05 ms 0.1-1.9 ms	- 0.1 ms	0.4 ms	0.2 ms
Ventricular Refractory Period	150-500 ms	10 ms	320 ms	±8 ms
A or V Sensitivity	0.25, 0.5, 0.75 1.0-10 mV	- 0.5 mV	A-0.75 mV V-2.5 mV	±20%

. . . continued

. . . continued

Parameter	Programmable Values	Increment	Nominal	Tolerance
Atrial Refractory Period	150-500 ms	10 ms	250 ms	±8 ms
PVARP	150-500 ms	10 ms	250 ms	±8 ms
PVARP Extension	Off, 50-400 ms	50 ms	Off	±8 ms
Hysteresis Rate Limit	Off or same choices as LRL	—	Off	±8 ms
Rate Smoothing	Off, 3, 6, 9, 12, 15, 18, 21, 25%	—	Off	±1%
ATR Mode	On, Off	—	Off	—
ATR Fallback Time	1-5 min	1 min	1 min	±1 cc
ATR Duration	10 cardiac cycles 20-80 cc 100-2000 cc	- 20 cc 100 cc	20 cc	±1 cc
Ventricular Blanking	30-60 ms	10 ms	40 ms	—

. . . continued

. . . continued

Parameter	Programmable Values	Increment	Nominal	Tolerance
Activity Threshold	V-Low, Low, Med-Low, Med, Med-High, High, V-High	—	Med	—
Reaction Time	10-50 sec	10 sec	30 sec	±3 sec
Response Factor	1-16	1	8	—
Recovery Time	2-16 min	1 min	5 min	±30 sec

APPENDIX B: PROGRAMMABLE PARAMETERS FOR BRADYCARDIA THERAPY MODES

Bradycardia Therapy

User programmable parameters are provided for controlling the delivery of patient-tailored, bradycardia therapy. These parameters are described in the following subsections; which parameters are meaningful with which pacing mode are listed in Table 2, Programmable Parameters for Bradycardia Therapy Modes.

Programmable Parameters for Bradycardia Therapy Modes

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Low Rate Limit	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Upper Rate Limit	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Maximum Sensor Rate											x	x	x	x	x	x	x	x
Fixed AV Delay							x	x	x	x					x	x	x	x

. . . continued

. . . continued

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Dynamic AV Delay							x			x					x			x
Sensed AV Delay Offset										x								x
Atrial Amplitude	x		x	x				x	x	x	x	x				x	x	x
Ventricu-lar Amplitude		x			x	x	x	x	x	x			x	x	x	x	x	x
Atrial Pulse Width	x		x	x				x	x	x	x	x				x	x	x

. . . continued

. . . continued

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Ventricular Pulse Width		x			x	x	x	x	x	x			x	x	x	x	x	x
Atrial Sensitivity	x			x					x	x		x					x	x
Ventricular Sensitivity		x				x	x		x	x				x	x		x	x
VRP		x				x	x		x	x				x	x		x	x
ARP	x			x					x	x		x					x	x
PVARP	x			x					x	x		x					x	x
PVARP Extension							x			x					x			x

. . . continued

. . . continued

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Hysteresis				x		x				x		x		x				x
Rate Smoothing				x		x	x			x		x		x	x			x
ATR Duration							x			x					x			x
ATR Fallback Mode							x			x					x			x
ATR Fallback Time							x			x					x			x
Activity Threshold											x	x	x	x	x	x	x	x

. . . continued

. . . continued

Parameter	AAT	VVT	AOO	AAI	VOO	VVI	VDD	DOO	DDI	DDD	AOOR	AAIR	VOOR	VVIR	VDDR	DOOR	DDIR	DDDR
Reaction Time											x	x	x	x	x	x	x	x
Response Factor											x	x	x	x	x	x	x	x
Recovery Time											x	x	x	x	x	x	x	x

Lower Rate Limit (LRL)

The Lower Rate Limit (LRL) is the number of generator pace pulses delivered per minute (atrium or ventricle) in the absence of

- Sensed intrinsic activity.
- Sensor-controlled pacing at a higher rate.

The LRL is affected in the following ways:

1. When Rate Hysteresis is disabled, the LRL shall define the longest allowable pacing interval.

2. In DXX or VXX modes, the LRL interval starts at a ventricular sensed or paced event.

3. In AXX modes, the LRL interval starts at an atrial sensed or paced event.

Upper Rate Limit (URL)

The Upper Rate Limit (URL) is the maximum rate at which the paced ventricular rate will track sensed atrial events. The URL interval is the minimum time between a ventricular event and the next ventricular pace.

Atrial-Ventricular (AV) Delay

The AV delay shall be the programmable time period from an atrial event (either intrinsic or paced) to a ventricular pace.

In atrial tracking modes, ventricular pacing shall occur in the absence of a sensed ventricular event within the programmed AV delay when the sensed atrial rate is between the programmed LRL and URL.

AV delay shall either be

1. Fixed (absolute time)

2. Dynamic

Paced AV Delay

A paced AV (PAV) delay shall occur when the AV delay is initiated by an atrial pace.

Sensed AV Delay

A sensed AV (SAV) delay shall occur when the AV delay is initiated by an atrial sense.

Dynamic AV Delay

If dynamic, the AV delay shall be determined individually for each new cardiac cycle based on the duration of previous cardiac cycles. The previous cardiac cycle length is multiplied by a factor stored in device memory to create the dynamic AV delay.

The AV delay shall vary between

1. A programmable maximum paced AV delay

2. A programmable minimum paced AV delay

Sensed AV Delay Offset

The Sensed AV Delay Offset option shall shorten the AV delay following a tracked atrial sense.

Depending on which option is functioning, the sensed AV delay offset shall be applied to the following:

1. The fixed AV delay

2. The dynamic AV delay

Refractory Periods

To avoid false sensing, refractory periods follow events during which senses in the affected chamber are ignored. To show that a sense was ignored due to refractory, its marker is displayed in parentheses.

Ventricular Refractory Period (VRP)

The Ventricular Refractory Period shall be the programmed time interval following a ventricular event during which time ventricular senses shall not inhibit nor trigger pacing.

Atrial Refractory Period (ARP)

For single chamber atrial modes, the Atrial Refractory Period (ARP) shall be the programmed time interval following an atrial event during which time atrial events shall not inhibit nor trigger pacing.

Post Ventricular Atrial Refractory Period (PVARP)

The Post Ventricular Atrial Refractory Period shall be available in modes with ventricular pacing and atrial sensing. The Post Ventricular Atrial Refractory Period shall be the programmable time interval following a ventricular event when an atrial cardiac event shall not 1. Inhibit an atrial pace. 2. Trigger a ventricular pace.

Extended PVARP

The Extended PVARP works as follows:

1. When Extended PVARP is enabled, an occurrence of a premature ventricular contraction (PVC) shall cause the pulse generator to use the Extended PVARP value for the post-ventricular atrial refractory period following the PVC.

2. The PVARP shall always return to its normal programmed value on the subsequent cardiac cycle regardless of PVC and other events. At most one PVARP extension shall occur every two cardiac cycles.

Refractory During AV Interval

The PG shall also be in refractory to atrial senses during the AV interval. In this context, refractory means the pacemaker does not track or inhibit based on the sensed activity.

Noise Response

In the presence of continuous noise the device response shall be asynchronous pacing.

Atrial Tachycardia Response (ATR)

The Atrial Tachycardia Response prevents long term pacing of a patient at unacceptably high rates during atrial tachycardia. When Atrial Tachycardia Response is enabled, the pulse generator shall declare an atrial tachycardia if the intrinsic atrial rate exceeds the URL for a sufficient amount of time.

Atrial Tachycardia Detection

The atrial tachycardia (AT) detection algorithm determines onset and cessation of atrial tachycardia.

1. AT onset shall be detected when the intervals between atrial senses are predominately, but not exclusively, faster than URL.

2. AT cessation shall be detected when the intervals between atrial senses are mostly, but not exclusively, faster than URL.

3. The detection period shall be short enough so ATR therapy is not unnecessarily delayed nor continued.

4. The detection period shall be long enough that occasional premature atrial contractions do not cause unnecessary ATR therapy, nor cease necessary ATR therapy upon occasional slow beats.

ATR Duration

ATR Duration works as follows:

1. When atrial tachycardia is detected, the ATR algorithm shall enter an ATR Duration state.

2. When in ATR Duration, the PG shall delay a programmed number of cardiac cycles before entering Fallback.

3. The Duration delay shall be terminated immediately and Fallback shall be avoided if, during the Duration delay, the ATR detection algorithm determines that atrial tachycardia is over.

ATR Fallback

If the atrial tachycardia condition exists after the ATR Duration delay is over, the following shall occur:

1. The PG enters a Fallback state and switches to a VVIR Fallback Mode.

2. The pacing rate is dropped to the lower rate limit. The fallback time is the total time required to drop the rate to the LRL.

3. During Fallback, if the ATR detection algorithm determines that atrial tachycardia is over, the following shall occur:

   - Fallback is terminated immediately
   - The mode is switched back to normal

4. ATR-related mode switches shall always be synchronized to a ventricular paced or sensed event.

Rate-Adaptive Pacing

The device shall have the ability to adjust the cardiac cycle in response to metabolic need as measured from body motion using an accelerometer.

Maximum Sensor Rate (MSR)

The Maximum Sensor Rate is the maximum pacing rate allowed as a result of sensor control.

The Maximum Sensor Rate shall be

1. Required for rate adaptive modes

2. Independently programmable from the URL

Activity Threshold

The activity threshold is the value the accelerometer sensor output shall exceed before the pacemaker's rate is affected by activity data.

Response Factor

The accelerometer shall determine the pacing rate that occurs at various levels of steady state patient activity.

Based on equivalent patient activity:

1. The highest response factor setting (16) shall allow the greatest incremental change in rate.

2. The lowest response factor setting (1) shall allow a smaller change in rate.

Reaction Time

The accelerometer shall determine the rate of increase of the pacing rate. The reaction time is the time required for an activity to drive the rate from LRL to MSR.

Recovery Time

The accelerometer shall determine the rate of decrease of the pacing rate. The recovery time shall be the time required for the rate to fall from MSR to LRL when activity falls below the activity threshold.

Hysteresis Pacing

When enabled, hysteresis pacing shall result in a longer period following a sensed event before pacing. This encourages self-pacing during exercise by waiting a little longer to pace after senses, hoping that another sense will inhibit the pace.

To use hysteresis pacing:

1. Hysteresis pacing must be enabled (not Off).

2. The pacing mode must be inhibiting or tracking.

3. The current pacing rate must be faster than the Hysteresis Rate Limit (HRL), which may be slower than the Lower Rate Limit (LRL).

4. When in AAI mode, a single, non-refractory sensed atrial event shall activate hysteresis pacing.

5. When in an inhibiting or tracking mode with ventricular pacing, a single, non-refractory sensed ventricular event shall activate hysteresis pacing.

Rate Smoothing

Rate Smoothing shall limit the pacing rate change that occurs due to precipitous changes in the intrinsic rate.

Two programmable rate smoothing parameters shall be available to allow the cardiac cycle interval change to be a percentage of the previous cardiac cycle interval:

1. Rate Smoothing Up

2. Rate Smoothing Down

The increase in pacing rate shall not exceed the Rate Smoothing Up percentage.

The decrease in pacing rate shall not exceed the Rate Smoothing Down percentage.

APPENDIX C: THE RESULTS OF SPACE STATE ANALYSIS OF THE CCS MODEL

Boundedness Properties

The boundness properties, which report the number of tokens for places after executing all reachable markings.

Best Integer Bounds

The *Best Integer Bounds* show the *upper* and the *lower* columns, which respectively indicate the maximal and minimal numbers of colour tokens remain in each place in any reachable marking.

In the proposed model in this chapter, the places of the colour set **ECG** or **interval ration** always have a single token for both best upper and lower integer bounds meaning these places always hold a constant number of tokens. On the other hand, the other places hold at most one colour token when a colour token is available in the place, and also no colour token when an enabled transition transferred the colour token. All these reported numbers are accurate according to the specification of the system.

```
Best Integer Bounds
                           Upper       Lower
   M0_Core_Elements'Event 1
                           1           1
   M0_Core_Elements'Pacemaker 1
                           1           0
   M0_Core_Elements'interval 1
                           1           1
   M1_Atrial_Depolarization'PP_interval 1
```

```
                         1          1
M1_Atrial_Depolarization'P_wave 1
                         1          1
M1_Atrial_Depolarization'SA_node 1
                         1          0
M1_Atrial_Depolarization'atria_pre 1
                         1          0
M1_Atrial_Depolarization'atrial_post 1
                         1          0
M2_VA_Node'PR_seg 1      1          1
M2_VA_Node'VA_post 1     1          0
M2_VA_Node'VA_pre 1      1          0
M2_VA_Node'VA_ratio 1    1          1
M2_VA_Node'atrial_post 1
                         1          0
M3_Ventricular_Depolarization'QRS_complex_post 1
                         1          0
M3_Ventricular_Depolarization'QRS_complex_pre 1
                         1          0
M3_Ventricular_Depolarization'Q_ratio 1
                         1          1
M3_Ventricular_Depolarization'Q_wave 1
                         1          1
M3_Ventricular_Depolarization'RR_interval 1
                         1          1
M3_Ventricular_Depolarization'R_wave 1
```

```
                              1          1
     M3_Ventricular_Depolarization'S_ratio 1
                              1          1
     M3_Ventricular_Depolarization'S_wave 1
                              1          1
     M3_Ventricular_Depolarization'VA_post 1
                              1          0
     M3_Ventricular_Depolarization'post_Q_wave_pre_R_wave 1
                              1          0
     M3_Ventricular_Depolarization'post_R_wave_pre_S_wave 1
                              1          0
     M4_ST_Segment'QRS_complex_post 1
                              1          0
     M4_ST_Segment'ST_ratio 1
                              1          1
     M4_ST_Segment'ST_seg 1   1          1
     M4_ST_Segment'end_of_ST_segment 1
                              1          0
     M4_ST_Segment'start_of_ST_seg 1
                              1          0
     M5_Ventricular_Repolarization'End_of_ST_segment 1
                              1          0
     M5_Ventricular_Repolarization'Pacemaker 1
                              1          0
     M5_Ventricular_Repolarization'TP_seg 1
                              1          1
```

```
    M5_Ventricular_Repolarization'TU_ratio 1
                          1          1
    M5_Ventricular_Repolarization'TU_seg 1
                          1          1
    M5_Ventricular_Repolarization'T_ratio 1
                          1          1
    M5_Ventricular_Repolarization'T_wave 1
                          1          1
    M5_Ventricular_Repolarization'U_ratio 1
                          1          1
    M5_Ventricular_Repolarization'U_wave 1
                          1          1
    M5_Ventricular_Repolarization'heartbeat_ratio 1
                          1          1
    M5_Ventricular_Repolarization'post_R_wave_pre_TU_seg 1
                          1          0
    M5_Ventricular_Repolarization'post_TU_seg_pre_U_wave 1
                          1          0
    M5_Ventricular_Repolarization'ventrical_post 1
                          1          0
    M5_Ventricular_Repolarization'ventrical_pre 1
                          1          0
```

Best Integer Bounds

Best Upper Multi-set Bounds

The *Best Upper Multi-set Bounds* represent the produced colour tokens for each place in accordance with its defined colour set in any reachable marking. Due to the massive number of produced colour tokens, this section contains just a few samples.

```
  Best Upper Multi-set Bounds
     M0_Core_Elements'Event 1
                                1`[]++
1`[{id=1,title=P_wave,start_time=0.0,end_time=80.5837173018,
   assigned_duration=90.0,calculated_duration=80.5837173018,
   amplitude={I=1.25,II=2.5,III=1.25,aVR=~1.875,aVL=0.0,aVF
   =1.875,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
1`[{id=2,title=PR_segment,start_time=80.5837173018,end_time
   =171.120407978,assigned_duration=90.0,calculated_duration
   =90.5366906759,amplitude={I=0.0,II=0.0,III=0.0,aVR=0.0,aVL
   =0.0,aVF=0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
1`[{id=3,title=Q_wave,start_time=90.5366906759,end_time
   =100.61678563,assigned_duration=10.0,calculated_duration
   =10.0800949545,amplitude={I=1.0,II=2.0,III=1.0,aVR=~1.5,aVL
   =0.0,aVF=1.5,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
1`[{id=4,title=R_wave,start_time=90.5366906759,end_time
   =102.848491302,assigned_duration=20.0,calculated_duration
   =12.3118006263,amplitude={I=3.5,II=7.0,III=3.5,aVR=~5.25,
   aVL=0.0,aVF=5.25,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6
   =0.0}}]++
.
```

```
.
.
     M5_Ventricular_Repolarization'heartbeat_ratio 1
                         1`{P_previous_interval=0,
                            P_last_interval=0,
                            R_previous_interval=0,
                            R_last_interval=0,
                            heartbeat_counter=0,
                            heartbeat_ratio=0,P_ratio=0,
                            Q_ratio=0,R_ratio=0,S_ratio=0,
                            T_ratio=0,U_ratio=0,PR_ratio=0,
                            ST_ratio=0,TU_ratio=0}++
1`{P_previous_interval=0,P_last_interval=40,
   R_previous_interval=0,R_last_interval=0,heartbeat_counter
   =0,heartbeat_ratio=0,P_ratio=0,Q_ratio=0,R_ratio=0,S_ratio
   =0,T_ratio=0,U_ratio=0,PR_ratio=0,ST_ratio=0,TU_ratio=0}++
1`{P_previous_interval=0,P_last_interval=40,
   R_previous_interval=0,R_last_interval=97,heartbeat_counter
   =0,heartbeat_ratio=0,P_ratio=0,Q_ratio=0,R_ratio=0,S_ratio
   =0,T_ratio=0,U_ratio=0,PR_ratio=0,ST_ratio=0,TU_ratio=0}++
.
.
.
     M5_Ventricular_Repolarization'post_R_wave_pre_TU_seg 1
                         1`electronegative
     M5_Ventricular_Repolarization'post_TU_seg_pre_U_wave 1
```

```
                    1‘electronegative
    M5_Ventricular_Repolarization’ventrical_post 1
                    1‘electronegative
    M5_Ventricular_Repolarization’ventrical_pre 1
                    1‘electronegative
```

Best Upper Multi-set Bounds

Best Lower Multi-set Bounds

The *empty* multiset shows the absence of colour tokens by the specified colour set.

```
 Best Lower Multi-set Bounds
    M0_Core_Elements’Event 1
                    empty
    M0_Core_Elements’Pacemaker 1
                    empty
    M0_Core_Elements’interval 1
                    empty
    M1_Atrial_Depolarization’PP_interval 1
                    empty
    M1_Atrial_Depolarization’P_wave 1
                    empty
    M1_Atrial_Depolarization’SA_node 1
                    empty
    M1_Atrial_Depolarization’atria_pre 1
                    empty
```

```
    M1_Atrial_Depolarization'atrial_post 1
                    empty
    M2_VA_Node'PR_seg 1 empty
    M2_VA_Node'VA_post 1
                    empty
    M2_VA_Node'VA_pre 1 empty
    M2_VA_Node'VA_ratio 1
                    empty
    M2_VA_Node'atrial_post 1
                    empty
    M3_Ventricular_Depolarization'QRS_complex_post 1
                    empty
    M3_Ventricular_Depolarization'QRS_complex_pre 1
                    empty
    M3_Ventricular_Depolarization'Q_ratio 1
                    empty
    M3_Ventricular_Depolarization'Q_wave 1
                    empty
    M3_Ventricular_Depolarization'RR_interval 1
                    empty
    M3_Ventricular_Depolarization'R_wave 1
                    empty
    M3_Ventricular_Depolarization'S_ratio 1
                    empty
    M3_Ventricular_Depolarization'S_wave 1
                    empty
```

```
      M3_Ventricular_Depolarization'VA_post 1
                          empty
      M3_Ventricular_Depolarization'post_Q_wave_pre_R_wave 1
                          empty
      M3_Ventricular_Depolarization'post_R_wave_pre_S_wave 1
                          empty
      M4_ST_Segment'QRS_complex_post 1
                          empty
      M4_ST_Segment'ST_ratio 1
                          empty
      M4_ST_Segment'ST_seg 1
                          empty
      M4_ST_Segment'end_of_ST_segment 1
                          empty
      M4_ST_Segment'start_of_ST_seg 1
                          empty
      M5_Ventricular_Repolarization'End_of_ST_segment 1
                          empty
      M5_Ventricular_Repolarization'Pacemaker 1
                          empty
      M5_Ventricular_Repolarization'TP_seg 1
                          empty
      M5_Ventricular_Repolarization'TU_ratio 1
                          empty
      M5_Ventricular_Repolarization'TU_seg 1
                          empty
```

```
    M5_Ventricular_Repolarization'T_ratio 1
                        empty
    M5_Ventricular_Repolarization'T_wave 1
                        empty
    M5_Ventricular_Repolarization'U_ratio 1
                        empty
    M5_Ventricular_Repolarization'U_wave 1
                        empty
    M5_Ventricular_Repolarization'heartbeat_ratio 1
                        empty
    M5_Ventricular_Repolarization'post_R_wave_pre_TU_seg 1
                        empty
    M5_Ventricular_Repolarization'post_TU_seg_pre_U_wave 1
                        empty
    M5_Ventricular_Repolarization'ventrical_post 1
                        empty
    M5_Ventricular_Repolarization'ventrical_pre 1
                        empty
```

Best Lower Multi-set Bounds

APPENDIX D: THE RESULTS OF SPACE STATE ANALYSIS OF THE CPS MODEL

Boundedness Properties

The boundness properties, which report the number of tokens for places after executing all reachable markings.

Best Integer Bounds

The *Best Integer Bounds* show the *upper* and the *lower* columns, which respectively indicate the maximal and minimal numbers of colour tokens remain in each place in any reachable marking.

```
Best Integer Bounds
                           Upper      Lower
   A0_Artificial_Pacemaker'e7 1
                           1          1
   A0_Artificial_Pacemaker'i1 1
                           1          1
   A0_Artificial_Pacemaker'p7 1
                           1          1
   A0_Artificial_Pacemaker't7 1
                           1          1
   M0_Core_Elements'Event 1
                           1          1
   M0_Core_Elements'Pacemaker 1
                           1          1
   M0_Core_Elements'Timing_Cycle 1
                           1          1
```

```
    M0_Core_Elements'interval 1
                            1           1
    M1_Atrial_Depolarization'e1 1
                            1           1
    M1_Atrial_Depolarization'i1 1
                            1           1
    M1_Atrial_Depolarization'p1 1
                            1           1
    M1_Atrial_Depolarization't1 1
                            1           1
    M2_VA_Node'e2 1         1           1
    M2_VA_Node'i2 1         1           1
    M2_VA_Node'p2 1         1           1
    M2_VA_Node't2 1         1           1
    M3_Ventricular_Depolarization'e4 1
                            1           1
    M3_Ventricular_Depolarization'e42 1
                            1           1
    M3_Ventricular_Depolarization'i4 1
                            1           1
    M3_Ventricular_Depolarization'i42 1
                            1           1
    M3_Ventricular_Depolarization'p4 1
                            1           1
    M3_Ventricular_Depolarization'p42 1
                            1           1
```

```
    M3_Ventricular_Depolarization't4 1
                            1          1
    M3_Ventricular_Depolarization't42 1
                            1          1
    M4_ST_Segment'e5 1      1          1
    M4_ST_Segment'i5 1      1          1
    M4_ST_Segment'p5 1      1          1
    M4_ST_Segment't5 1      1          1
    M5_Ventricular_Repolarization'e6 1
                            1          1
    M5_Ventricular_Repolarization'e62 1
                            1          1
    M5_Ventricular_Repolarization'i6 1
                            1          1
    M5_Ventricular_Repolarization'i62 1
                            1          1
    M5_Ventricular_Repolarization'i63 1
                            1          1
    M5_Ventricular_Repolarization'p6 1
                            1          1
    M5_Ventricular_Repolarization'p62 1
                            1          1
    M5_Ventricular_Repolarization't6 1
                            1          1
    M5_Ventricular_Repolarization't62 1
                            1          1
```

```
    M5_Ventricular_Repolarization't63 1
                                    1          1
```

Best Integer Bounds

Best Upper Multi-set Bounds

The *Best Upper Multi-set Bounds* represent the produced colour tokens for each place in accordance with its defined colour set in any reachable marking. Due to the massive number of produced colour tokens, this section contains just a few samples.

```
 Best Upper Multi-set Bounds
    A0_Artificial_Pacemaker'e7 1
                           1`[]++
1`[{id=1,title=P_wave,start_time=0.0,end_time=80.0,duration=80.0,amplitude={I=1.25,II=2.5,III=1.25,aVR=~1.875,aVL=0.0,aVF=1.875,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
1`[{id=2,title=PR_segment,start_time=80.0,end_time=170.0,duration=90.0,amplitude={I=0.0,II=0.0,III=0.0,aVR=0.0,aVL=0.0,aVF=0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
1`[{id=3,title=Q_wave,start_time=90.0,end_time=120.0,duration=30.0,amplitude={I=1.0,II=2.0,III=1.0,aVR=~1.5,aVL=0.0,aVF=1.5,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
.
.
.
1`[{id=21,title=ventricular_pacing,start_time=525.0,end_time=526.001,duration=1.001,amplitude={I=2.75,II=5.5,III=2.75,aVR=~4.125,aVL=0.0,aVF=4.125,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6=0.0}}]++
```

```
1`[{id=22,title=TP_segment,start_time=525.0,end_time=651.923,
    duration=126.923,amplitude={I=0.0,II=0.0,III=0.0,aVR=0.0,
    aVL=0.0,aVF=0.0,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6
    =0.0}}]++
1`[{id=23,title=P_wave,start_time=525.0,end_time=605.0,
    duration=80.0,amplitude={I=1.25,II=2.5,III=1.25,aVR=~1.875,
    aVL=0.0,aVF=1.875,V1=0.0,V2=0.0,V3=0.0,V4=0.0,V5=0.0,V6
    =0.0}}]++
.
.
.
    A0_Artificial_Pacemaker'i1 1
                                1`{P_ratio=0,P_previous_interval=0,
                                    P_last_interval=0,
                                    R_previous_interval=0,
                                    R_last_interval=0,heartbeat_ratio
                                    =0,Q_ratio=0,R_ratio=0,S_ratio=0,
                                    T_ratio=0,U_ratio=0,PR_ratio=0,
                                    ST_ratio=0,TU_ratio=0}++
1`{P_ratio=0,P_previous_interval=0,P_last_interval=40,
    R_previous_interval=0,R_last_interval=0,heartbeat_ratio=0,
    Q_ratio=0,R_ratio=0,S_ratio=0,T_ratio=0,U_ratio=0,PR_ratio
    =0,ST_ratio=0,TU_ratio=0}++
.
.
```

```
.
```

Best Upper Multi-set Bounds

Best Lower Multi-set Bounds

The *empty* multiset shows the absence of colour tokens by the specified colour set.

```
 Best Lower Multi-set Bounds
    A0_Artificial_Pacemaker'e7 1
                        empty
    A0_Artificial_Pacemaker'i1 1
                        empty
    A0_Artificial_Pacemaker'p7 1
                        empty
    A0_Artificial_Pacemaker't7 1
                        empty
    M0_Core_Elements'Event 1
                        empty
    M0_Core_Elements'Pacemaker 1
                        empty
    M0_Core_Elements'Timing_Cycle 1
                        empty
    M0_Core_Elements'interval 1
                        empty
    M1_Atrial_Depolarization'e1 1
                        empty
```

```
    M1_Atrial_Depolarization'i1 1
                        empty
    M1_Atrial_Depolarization'p1 1
                        empty
    M1_Atrial_Depolarization't1 1
                        empty
    M2_VA_Node'e2 1     empty
    M2_VA_Node'i2 1     empty
    M2_VA_Node'p2 1     empty
    M2_VA_Node't2 1     empty
    M3_Ventricular_Depolarization'e4 1
                        empty
    M3_Ventricular_Depolarization'e42 1
                        empty
    M3_Ventricular_Depolarization'i4 1
                        empty
    M3_Ventricular_Depolarization'i42 1
                        empty
    M3_Ventricular_Depolarization'p4 1
                        empty
    M3_Ventricular_Depolarization'p42 1
                        empty
    M3_Ventricular_Depolarization't4 1
                        empty
    M3_Ventricular_Depolarization't42 1
                        empty
```

```
      M4_ST_Segment'e5 1  empty
      M4_ST_Segment'i5 1  empty
      M4_ST_Segment'p5 1  empty
      M4_ST_Segment't5 1  empty
      M5_Ventricular_Repolarization'e6 1
                          empty
      M5_Ventricular_Repolarization'e62 1
                          empty
      M5_Ventricular_Repolarization'i6 1
                          empty
      M5_Ventricular_Repolarization'i62 1
                          empty
      M5_Ventricular_Repolarization'i63 1
                          empty
      M5_Ventricular_Repolarization'p6 1
                          empty
      M5_Ventricular_Repolarization'p62 1
                          empty
      M5_Ventricular_Repolarization't6 1
                          empty
      M5_Ventricular_Repolarization't62 1
                          empty
      M5_Ventricular_Repolarization't63 1
                          empty
```

Best Lower Multi-set Bounds

Other Properties

These properties include Home Properties, Liveness Properties, and Fairness Properties.

```
Home Properties
------------------------------------------

 Home Markings
    Initial Marking is not a home marking


Liveness Properties
------------------------------------------

 Dead Markings
    [2005]

 Dead Transition Instances
    None

 Live Transition Instances
    None


Fairness Properties
------------------------------------------
```

```
    No infinite occurrence sequences.
```

Other Properties

BIBLIOGRAPHY

Barbara Aehlert. *ECGs made easy.* Mosby Incorporated, 2001.

AIS Group. CPN Tools by The University of Technology, Eindhoven, The Netherlands. *www.cpntools.org*, 2013.

Homa Alemzadeh, Ravishankar K Iyer, Zbigniew Kalbarczyk, and Jai Raman. Analysis of safety-critical computer failures in medical devices. *IEEE Security & Privacy*, 11(4):14–26, 2013.

Mohammad Alqarni. *Modelling Concurrent Systems with Interval Processes.* PhD thesis, 2016.

Paulo ES Barbosa, Misael Morais, Katia Galdino, Melquisedec Andrade, Luís Gomes, Filipe Moutinho, and Jorge CA de Figueiredo. Towards medical device behavioural validation using petri nets. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*, pages 4–10. IEEE, 2013.

Boston Scientific. Pacemaker system specification. *Boston Scientific*, 2007.

Boston Scientific. ALTRUA™60 - System Guide. *Boston Scientific*, (357921-003), 2011.

Boston Scientific. Pacemakers Patient Manual. *Boston Scientific*, (356429-010), 2014.

Djamila Boukredera, Ramdane Maamri, and Samir Aknine. A timed colored petri-net-based modeling for contract net protocol with temporal aspects. In *Proceedings of the Seventh International Multi-Conference on Computing in the Global Information Technology (ICCGI 2012)*, pages 40–45, 2012.

Muriel Boulakia, Serge Cazeau, Miguel A Fernández, Jean-Frédéric Gerbeau, and Nejib Zemzemi. Mathematical modeling of electrocardiograms: a numerical study. *Annals of biomedical engineering*, 38(3):1071–1097, 2010.

Fred DJ Bowden. A brief survey and synthesis of the roles of time in petri nets. *Mathematical and Computer Modelling*, 31(10-12):55–68, 2000.

Martin Carlsson. Automatic code generation from a colored petri net specification for game development with unity3d, 2018.

Gareth Carter, Rosemary Monahan, and Joseph M Morris. Software refinement with perfect developer. In *Software Engineering and Formal Methods, 2005. SEFM 2005. Third IEEE International Conference on*, pages 363–372. IEEE, 2005.

Ming Chen and Ralf Hofestädt. Open problems in petri net modeling and simulation of biological systems. *IT–Information Technology*, 56(2):76–81, 2014.

Hsiu-Sen Chiang. Ecg-based mental stress assessment using fuzzy computing and associative petri net. *Journal of Medical and Biological Engineering*, 35(6):833–844, 2015.

Toshio M Chin and Alan S Willsky. Stochastic petri net modeling of wave sequences in cardiac arrhythmias. *Computers and Biomedical Research*, 22(2):136–159, 1989.

CMMI Product Team. Cmmi for software engineering, version 1.1, staged representation (cmmi-sw, v1.1, staged). Technical Report CMU/SEI-2002-TR-029, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2002.

Anne B Curtis. *Fundamentals of Cardiac Pacing.* Jones & Bartlett Publishers, 2010.

Tomas B Garcia. *Introduction to 12-lead ECG: The art of interpretation.* Jones & Bartlett Publishers, 2014.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli, editors. *Fundamentals of Software Engineering.* Pearson Prentice Hall, 2nd edition, 2003.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Artur O Gomes and Marcel VM Oliveira. Formal development of a cardiac pacemaker:

from specification to code. In *Brazilian Symposium on Formal Methods*, pages 210–225. Springer, 2010.

Artur Oliveira Gomes and Marcel Vinicius Medeiros Oliveira. Formal specification of a cardiac pacing system. In *International Symposium on Formal Methods*, pages 692–707. Springer, 2009.

Luis Gomes and Joao Paulo Barros. Automated code generation from petri nets based system specification. pages 355–360, 01 2001.

himanshu orsolutions. ecg-visualization. https://github.com/himanshu-orsolutions/ecg-visualization, 2020.

CAR Hoare, Jayadev Misra, Gary T Leavens, and Natarajan Shankar. The verified software initiative: A manifesto. *ACM Comput. Surv.*, 41(4):22–1, 2009.

IEEE Guide. Ieee draft guide: Adoption of the project management institute (pmi) standard: A guide to the project management body of knowledge (pmbok guide)-2008 (4th edition). *IEEE P1490/D1, May 2011*, pages 1–505, 2011. doi: 10.1109/IEEESTD.2011.5937011.

Shazia Javed and Noor Atinah Ahmad. Mathematical modeling of real time ecg waveform. In *Science and Information Conference*, pages 606–614. Springer, 2018.

K. Jensen and L. M. Kristensen. Coloured Petri Nets Modelling and Validation of Concurrent Systems. Berlin, 2009. Springer.

Kurt Jensen. Coloured Petri Nets and the Invariant Method. *Theoretical Computer Science*, 14(3):317–336, 1981.

Kurt Jensen, Soren Christensen, and Lars M. Kristensen. CPN Tools State Space Manual. *Department of Computer Science, Univerisity of Aarhus*, 2006.

Zhihao Jiang, Miroslav Pajic, Rajeev Alur, and Rahul Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2):191–213, 2014.

K. Jensen. *Coloured Petri Nets*. Springer, 1994. ISBN 3540580433.

Ekkart Kindler and Csaba Páles. 3D-visualization of Petri net models: Concept and realization. In *Applications and Theory of Petri Nets*, pages 464–473. Springer, 2004. ISBN 3540222367.

John C Knight. Safety critical systems: challenges and directions. In *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference on*, pages 547–550. IEEE, 2002.

Xiaoou Li and Felipe Lara-Rosano. Adaptive fuzzy petri nets for dynamic knowledge representation and inference. *Expert Systems with Applications*, 19(3):235–241, 2000.

Hugo Macedo, Peter Larsen, and John Fitzgerald. Incremental development of a distributed real-time model of a cardiac pacing system using vdm. *FM 2008: Formal Methods*, pages 181–197, 2008.

Ken MacLeod. *An essential introduction to cardiac electrophysiology*. World Scientific Publishing Company, 2013.

Negar Majma and Seyed Morteza Babamir. Medical software runtime checking using petri-nets & software agents. In *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*, pages 449–454. IEEE, 2014.

Negar Majma, Seyed Morteza Babamir, and Amirhassan Monadjemi. Runtime verification of pacemaker using fuzzy logic and colored petri-nets. In *Fuzzy and Intelligent Systems (CFIS), 2015 4th Iranian Joint Congress on*, pages 1–5. IEEE, 2015.

Negar Majma, Seyed Morteza Babamir, and Amirhassan Monadjemi. Utilizing fuzzy colored petri-nets to monitor cardiac pacemaker behavior. In *Application of Information and Communication Technologies (AICT), 2016 IEEE 10th International Conference on*, pages 1–5. IEEE, 2016.

Negar Majma, Seyed Morteza Babamir, and Amirhassan Monadjemi. Runtime verification of pacemaker functionality using hierarchical fuzzy colored petri-nets. *Journal of medical systems*, 41(2):27, 2017.

Patrick E McSharry, Gari D Clifford, Lionel Tarassenko, and Leonard A Smith. A dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on biomedical engineering*, 50(3):289–294, 2003.

Dominique Méry, Neeraj Kumar Singh, et al. Functional behavior of a cardiac pacing system. *International Journal of Discrete Event Control Systems*, 1(2):129–149, 2011.

Dominique Méry, Bernhard Schätz, and Alan Wassyng. The Pacemaker Challenge:

Developing Certifiable Medical Devices (Dagstuhl Seminar 14062). *Dagstuhl Reports*, 4(2):17–37, 2014. ISSN 2192-5283. doi: 10.4230/DagRep.4.2.17. URL `http://drops.dagstuhl.de/opus/volltexte/2014/4543`.

Kjeld H. Mortensen. Automatic code generation method based on coloured petri net models applied on an access control system. In Mogens Nielsen and Dan Simpson, editors, *Application and Theory of Petri Nets 2000*, pages 367–386, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. ISBN 978-3-540-44988-1.

Dragutin Novosel, Georg Noll, and Thomas F Lüscher. Corrected formula for the calculation of the electrical heart axis. *Croatian medical journal*, 40:77–79, 1999.

W. Reisig. Petri nets, an introduction, 2nd ed. Berlin, 1991. Springer Berlin Heidelberg.

Boston Scientific. Rhythm Management Product Performance Report - Q3 Edition. *Boston Scientific*, 2017.

Kirma Sembulingam and Prema Sembulingam. *Essentials of medical physiology.* JP Medical Ltd, 2012.

Asankhaya Sharma. Towards a verified cardiac pacemaker. Technical report, Technical Report NUS November, 2010.

Dong-Her Shih, Hsiu-Sen Chiang, and Ming-Hung Shih. Ecg identification of arrhythmias by using an associative petri net. In *2013 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 1–2. IEEE, 2013.

Kent Inge Fagerland Simonsen. Petricode: A tool for template-based code generation from cpn models. In Steve Counsell and Manuel Núñez, editors, *Software Engineering and Formal Methods*, pages 151–163, Cham, 2014. Springer International Publishing. ISBN 978-3-319-05032-4.

Alvaro Sobrinho, Angelo Perkusich, Leandro Dias da Silva, Thiago Cordeiro, Joilson Rêgo, and Paulo Cunha. Towards medical device certification: a colored petri nets model of a surface electrocardiography device. In *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, pages 2645–2651. IEEE, 2014.

Stanford Medical Center. Types of pacemakers. `https://stanfordhealthcare.org/medical-treatments/p/pacemaker/types.html`, n.d.

Luu Anh Tuan, Man Chun Zheng, and Quan Thanh Tho. Modeling and verification of safety critical systems: A case study on pacemaker. In *Secure Software Integration and Reliability Improvement (SSIRI), 2010 Fourth International Conference on*, pages 23–32. IEEE, 2010.

Wil MP van der Aalst and Christian Stahl. *Modeling business processes: a petri net-oriented approach*. MIT press, 2011.

Michael Westergaard. The BRITNeY Suite: A Platform for Experiments. In *7th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools (CPN 2006)*, 2006.

Michael Westergaard. CPN tools 4: multi-formalism and extensibility. In *Application and Theory of Petri Nets and Concurrency*, pages 400–409. Springer, 2013.

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

SK Yang. A petri net approach to remote diagnosis for failures of cardiac pacemakers. *Quality and Reliability Engineering International*, 20(8):761–776, 2004.

zapaaris. Ecggraphviewer. https://github.com/zapaaris/ECGGraphViewer, 2020.