STT EVENT STREAM FEATURE TO ASSIST SOFTWARE TESTING OF
IMPLANTABLE DEVICES IN ST. JUDE MEDICAL

A Thesis

Presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Engineering,

with Specializations in Biomedical Engineering

by

Yong Jin Park

February 2009

© 2009
Yong Jin Park
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: STT Event Stream Feature to Assist Software Testing
of Implantable Devices in St. Jude Medical

AUTHOR: Yong Jin Park

DATE SUBMITTED: February 2009

COMMITTEE CHAIR: Dr. Lanny Griffin, Dept Chair-Biomed&Gen Eng

COMMITTEE MEMBER: Dr. Dan Walsh, Int Assoc Dean Grad Prgm/Rsrch

COMMITTEE MEMBER: Dr. Robert Crockett, Int Dir Biomedical Eng

ABSTRACT

STT Event Stream Feature to Assist Software Testing of Implantable Devices in St. Jude Medical, Yong Jin Park

During development and testing of the pacemaker and defibrillator device functionality, engineers in the cardiac rhythm management industry use a patient simulator to ensure device functionality properly before device is tested with an animal or a human. The patient simulator is also used in the formal device product testing. In St. Jude Medical, a patient simulator called Simulation Test Tool (STT) has been developed and used by engineers in the company. While the Heart Simulator (HS) feature based on physiological heart model in the STT has been served as a main cardiac rhythm simulation feature, there has been an increasing need of a new feature in the STT for engineers to create heart rhythm scenarios more easily and effectively. This thesis covers the design and implementation of the new STT feature, called Event Stream, which allows users to create heart rhythm scenarios using simple text string based syntax for testing device functionality.

ACKNOWLEDGMENTS

I would like to thank Cal Poly, College of Engineering, and Dr. Lanny Griffin for his input on the technology content for this thesis and also for helping me remotely with the administrative aspects of submitting this thesis so close to the deadline. And I would like to thank Dr. Dan Walsh and Dr. Robert Crockett for emphasizing the learning by doing principle.

I would like to thank Siew Er, Software Tools Systems Director at St. Jude Medical for giving me the opportunity to develop and manage the Simulation Test Tool (STT) project and for supporting me when needed.

Table of Contents

List of Tables	vii
List of Figures	vii
Chapter 1: Background	1
The Basic Understanding of Heart	1
The Heart and Cardiac Rhythm	2
St. Jude Medical Cardiac Rhythm Management Devices	4
Simulation Test Tool (STT)	5
Heart Simulator (HS)	9
Lead Impedance	10
Event Legend	11
Chapter 2: Objective	13
Chapter 3: Methods	14
Chapter 4: Event Stream (ES)	17
Overview	17
Event Stream Software Application in PC	19
Event Stream Software Application in SIM	21
Detailed Event Stream Language Description	22
Events	22
Event property	22
Intrinsic Event properties	23
Pacing Event properties	24
Evoked Response properties	25
Timeout	26
Sequence	27
Repeat Count	27
Scenario	28
Operation	28
Set Lead Impedance Operation	28
Value Variability	30
Variant	31
Increment Variant	32
Decrement Variant	32
Random Variant	33
Key Words	35
Chapter 5: Discussion	37
Chapter 6: Conclusion Remarks	40
Acronyms and Definitions	41

Bibliography	42
--------------------	----

List of Tables

Table 1 - Event Stream Event Symbols	22
Table 2 - Intrinsic Event Property.....	23
Table 3 - Pacing Event property	25
Table 4 - Lead Impedance Operation Token	29
Table 5 - Lead Operation Property	29
Table 6 - Variant Signature Token.....	32
Table 7 - Increment Variant.....	32
Table 8 - Decrement Variant.....	33
Table 9 - Event Stream Language Key Words	35

List of Figures

Figure 1 - Anatomy of Heart.....	1
Figure 2 - Conduction System of the Heart	4
Figure 3 - Heart Rhythm Disorder.....	5
Figure 4 - Simulation Test Tool (STT) System	6
Figure 5 - STT PC Application.....	7
Figure 6 - Simulation Interface Module (SIM) Software Components	8
Figure 7 - STT Heart Simulator.....	9
Figure 8 - STT Lead Impedance Simulator	10
Figure 9 - Events in STT.....	11
Figure 10 - STT Event Legend	12
Figure 11 - STT Event Stream PC Application	19

Chapter 1: Background

The Basic Understanding of Heart

The heart is divided into the right and left side – each of which consists of a receiving chamber (atrium) and ejecting chamber (ventricle). The right atrium receives blood that is low in oxygen from all the veins in the body, and then the right ventricle pumps that blood to the lungs, where it will become re-oxygenated. The left atrium of the heart receives this oxygen-rich blood from the lungs. The left ventricle then pumps the blood to the body. The heart functions to deliver oxygen and nutrients to tissue through the arteries and picks up carbon dioxide. This cycle of contraction and relaxation is what makes the heart such an important organ.¹

Figure 1 - Anatomy of Heart

The Heart and Cardiac Rhythm

All four heart chambers possess cells that are capable of initiating beats. However, the conduction system of the heart is governed by the sinoatrial node (SA node) because it generates impulses with the greatest frequency, typically around 60 to 70 beats per minute at rest and is termed the natural pacemaker of the heart. The atrioventricular node (AV node), also known as the secondary pacemaker, has an intrinsic rate of 40 to 60 beats per minutes. The tertiary pacemaker, which consists of bundle of His, the left and right bundle branches, and the Purkinje fibers, has a much slower intrinsic rate, around 30 to 40 beats per minute. Because the frequencies of these beats are so slow, they are not sufficient to produce the necessary cardiac output required by the human body if the other pacemakers become damaged.

As previously discussed, the SA node is where a normal, healthy heart muscle generates the depolarization that initiates the heart beat. The SA node is located at the junction of the high right atrium, the junction of the crista terminalis. There is a layer of a thick band of atrial muscle at the border of the atrial appendage, and the superior vena cava. Positioned beneath the epicardial surface of the crista terminalis, there is a layer of atrial muscle between the node and endocardium, so the SA node does not occupy the entire thickness of the atrial myocardium.

From the SA node, the cardiac impulse spreads throughout the right atrium along the anterior, middle, and posterior internodal pathways at a rate of about 1 m/s. These tracts are composed of ordinary myocardial cells and specialized conducting fibers; they are the main routes of conduction from the SA node to the atrioventricular

(AV node). Bachmann's bundle is the interatrial tract that is responsible for conducting the impulse directly from the SA node to the left atrium.

The electric impulse generated from the SA node is normally conducted through the AV node. The AV node provides the path of the least resistance for the impulse to proceed to the ventricles. The conduction is delayed in the AV node, which allows for maximal ventricular filling with blood. The bundle of His is located distal to the AV node within the ventricular septum, it continues the electrical impulse from the AV node down to the ventricles. It then divides into the left and right bundle branches, which are responsible for conducting the impulse to their respective ventricles. The left and right bundle branches ultimately subdivide into a complex network of conducting fibers, named the Purkinje fibers. The conduction velocity in the Purkinje fiber is faster than any other conduction velocity system within the heart; the conduction velocity is estimated to be on the range of 1 to 4 m/s. This allows for rapid endocardial ventricular activation.

The first portions of the ventricles that are excited from the electrical impulse are the interventricular septum and the papillary muscles. The early contraction of the septum makes it rigid and acts as an anchor point for the contraction of the rest of the myocardium. The endocardium of the ventricles are activated rapidly, but the epicardium is activated later as the conduction velocity through the myocardium is much slower (approximately 0.3 to 0.4 m/s). Also, the epicardium of the right ventricle is activated prior to the epicardium of the left ventricle, that is because the left ventricular wall is much thicker than that of the right ventricular wall. Apical and

central epicardial regions are activated somewhat earlier than their basal counterparts. Typically, the last regions that are activated are the posterolateral basal epicardial regions and a small location in the basal portion of the interventricular septum.

Figure 2 - Conduction System of the Heart

St. Jude Medical Cardiac Rhythm Management Devices

There are two lines of cardiac rhythm management devices in St. Jude Medical – pacemaker and Implantable Cardiac Defibrillator (ICD). The pacemaker treats bradycardia, which refers to a slow heart rate, usually below 60 beats per minute. The ICD treats tachyarrhythmia, which is a fast heart rate, usually above 100 beats per minute.² The devices are implanted typically in the patient's upper left shoulder just under the skin. They are connected to the heart by means of leads which are flexible

insulated wires running through a vein into the superior vena cava and into the inside of the heart, where they are attached directly to the muscle.³

Figure 3 - Heart Rhythm Disorder

Simulation Test Tool (STT)

The Simulation Test Tool (STT) is an integrated tool set of software and hardware components together that implement a system for testing of pacemakers and ICDs. The STT simulates the electrical artifacts of a patient's heart, producing signals that appear to the device under test to be coming from a real heart. It also detects and measures the outputs of the device, and is able to respond to them in real-time as a patient would or in a way that a tester specifies. Other features include the ability to program Lead Impedance and various Waveforms for heart events.

The STT has two major hardware components (see Figure 4). The first is a Pentium class PC workstation running the Windows operating system. The second component is the Simulator Interface Module (SIM), which is a freestanding embedded system, based on a PC architecture single board computer (SBC), and running the VxWorks real-time operating system. The SIM also includes the analog

and digital hardware interfaces to the target device. The SIM and the PC communicate via a Local Area Network (LAN) interface, using the TCP/IP protocol.

Figure 4 - Simulation Test Tool (STT) System

The PC hosts the non-real-time application software, including the user interface, data logging, output displays, and test programs.

Figure 5 - STT PC Application

The Simulator Interface Module (SIM) runs the VxWorks real-time operating system. The SIM hosts the real-time application software, including the Tcl Script Interpreter and Heart Simulator.

Figure 6 - Simulation Interface Module (SIM) Software Components

Heart Simulator (HS)

The Heart Simulator feature in the STT is a programmable physiological model of the heart. Parameters are programmable and realistic heart waveforms are emitted under software control. Refractory and other timing elements are user configurable. In cooperation with the hardware capabilities of the SIM, incoming pacer pulses are measured and logged.

The heart simulator also supports triggering of P or R wave relative to an A or V pulse, triggering of evoked response and polarization wave relative to an A or V pulse, display of event time, Atrial/Ventricular Pulse Amplitude and Pulse Width measurement, as well as pulse energy.

Figure 7 - STT Heart Simulator

Lead Impedance

The lead impedance of a pacemaker or ICD is composed of the impedance within the device, the lead, the electrodes, and the electrode-heart tissue interface. The Lead Impedance simulator in the STT provides a programmable DC input to the SIM for dynamic variation of the lead impedance.

The screenshot shows a software window titled "Lead Impedance". It contains four control panels for different heart chambers: "Right Atrium", "Left Atrium", "Right Ventricle", and "Left Ventricle". Each panel has three radio buttons: "Open", "Short", and "Impedance (Ohms)". The "Impedance (Ohms)" option is selected in all four panels, and each has a numeric input field set to "500". Below these panels is a "NOTE" section with an information icon, explaining the behavior of the "Open" and "Short" settings. At the bottom of the window are three buttons: "Program", "Clear Batch", and "Reset".

Right Atrium
☐ Open ☐ Short ☒ Impedance (Ohms) 500

Left Atrium
☐ Open ☐ Short ☒ Impedance (Ohms) 500

Right Ventricle
☐ Open ☐ Short ☒ Impedance (Ohms) 500

Left Ventricle
☐ Open ☐ Short ☒ Impedance (Ohms) 500

NOTE: When 'Open' is selected, the Tip conductor of the lead is disconnected from the simulator I/O circuits, but the Ring and Case remain connected. In hardware version 3.0 and higher the Ring conductor is also disconnected.
When 'Short' is selected, the Tip and Ring conductors of the lead are connected to each other, and remain connected to the simulator I/O circuits.

Program Clear Batch Reset

Figure 8 - STT Lead Impedance Simulator

Event Legend

Each event (heart or device) is represented by a unique symbol in the STT. For example, an intrinsic heart event in right atrium and ventricle chambers are represented by ‘P’ and ‘R’ character respectively. Likewise, pacing pulse detected in right atrium and ventricle chambers are represented by ‘A’ and ‘V’ character.

Figure 9 - Events in STT

The complete list of symbols used in the STT is available in a separate Legend Window screen.

Figure 10 - STT Event Legend

Chapter 2: Objective

While the STT has successfully provided patient simulation functionalities such as Heart Simulation and Lead Impedance simulation since its introduction in 2002, users have also found areas of improvement and requested a number of enhancements in the STT. In 2006, STT project members and I have taken a task to develop a new feature, Event Stream, based on following requirements from customers:

1. The new STT feature shall provide a separate screen to enable users to perform all necessary simulation set ups for Heart Simulation, Lead Impedance and Waveform instead of multiple screens for each simulation.
2. The new STT feature shall provide more intuitive ways of constructing simulation scenarios consisting of heart rhythms and device interactions.
3. The new STT feature shall allow users to control heart rhythm scenarios and reproduce the scenarios in more precise manner.
4. The new STT feature shall provide randomization of heart rhythm interval in the test scenario.

Chapter 3: Methods

From user requirements, the STT project team decided to create a new simple language called Event Stream, which allows users to specify desired heart rhythm scenarios using text strings on the Event Stream GUI edit window. Following describes how each user requirement is met in the Event Stream feature:

1. The new STT feature shall provide a separate screen to enable users to perform all necessary simulation set ups for Heart Simulation, Lead Impedance and Waveform instead of multiple screens for each simulation.

Solution:

- A separate Event Stream GUI edit widow is developed for user to specify the desired scenarios.
2. The new STT feature shall provide more intuitive ways of constructing simulation scenarios consisting of heart rhythms and device interactions.

Solution:

- The Event Stream language syntax is constructed with the existing event symbols to specify heart rhythm, lead impedance and waveform characteristics. For example, P and R letters denote atrial and ventricle intrinsic heart event respectively in the Event Stream.
3. The new STT feature shall allow users to control heart rhythm scenarios and reproduce the scenarios in more precise manner.

Solution:

- The Event Stream language syntax allows users to specify the exact time interval between two events. For example, the Event Stream scenario “P 120 R” generates an atrial event followed by a ventricular event after 120 ms delay.
- As each specified event and interval in the Event Stream scenario is executed exactly same manner, the following scenario with a loop count of 10 generates the same set of P and R events with interval of 120 ms between them 10 times.

“10 (P 120 R);”

4. The new STT feature shall provide randomization of heart rhythm interval in the test scenario.

Solution:

- The “+-“syntax in the Event Stream language allows user to randomize the interval between two events in each iteration. For example, in the scenario “10 (P 120+-10 R), the interval between P and R events randomly varies between 120 and 130 ms in each iteration of execution.

Chapter 4: Event Stream (ES)

Overview

The Event Stream language allows heart rhythms to be defined in text strings, where mnemonics are used to represent events, and integers are used to define the time intervals between those events. An event is either an input to the device from the test system, such as a P or an R wave, or an output from the device to the test system, such as an A pulse or a V pulse. Timing intervals are expressed in milliseconds.

For example, an Event Stream to generate a P wave, followed 150 milliseconds later by an R wave, would take the following form:

P 150 R

As heart rhythms often follow the same pattern for many cycles, a means is provided to repeat a pattern for a defined number of cycles. A basic intrinsic rhythm that is to run for 10 cycles would look like this:

10 (P 150 R 850);

In this example 'P' and 'R' represent intrinsic heart events. The first event (a P wave) will be generated as soon as the Event Stream begins execution, then after 150 milliseconds an R wave will be generated, followed by another P wave after 850 milliseconds. The sequence within the parentheses will repeat 10 times.

STT Events trace for this scenario:

1610 B	1 P	152 R	850 P	150 R	850 P	150 R	850 P	150 R	850 P
150 R	850 P	150 R	850 P	150 R	850 P	150 R	850 P	150 R	850 P
150 R	850 E								

“B” marks the beginning of the Event Stream execution

“E” marks the end of the Event Stream execution

A means is also provided to wait for an event from the device to occur, and then create the desired response. Additional refinements allow event characteristics such wave shape or pulse width and amplitude to be specified on an event by event basis. Random and pseudo-random intervals can be specified, as well as intervals that change progressively by a fixed amount. These simple mechanisms provide sufficient flexibility to define the majority of the heart rhythms required for device functional testing.

Event Stream Software Application in PC

Figure 11 - STT Event Stream PC Application

- In the Event Stream feature, Event Stream text sequences may be entered into the edit window.
- Existing files containing Event Stream sequences may be opened using the “Open” button. Once opened, the text from the file displays in the edit window.
- Text in the edit window may be saved to a file by using the “Save” button.

The “.ess” file extension will be automatically appended to the file name.

- The “New” button allows the edit window to be cleared, ready for new input.
- The “Skip Sequence” button will cause the execution of the current sequence to be terminated and processing will continue with the next sequence, if any.
- When the “Continuous” check box under “Run Mode” is checked and the Run button is pushed, the displayed Event Stream scenario will run continuously until the Stop button is selected. During an Event Stream scenario execution, the Continuous check box is disabled.
- When the “Random Seed” “Enable” check box is checked, a seed for the Random Number generator may be entered. Using a seed will cause the same random sequence to be repeated each time the Event Stream scenario is run. The Random Seed cannot be changed during event stream execution. If “Random Seed” is disabled the random numbers generated will be truly random. That means that the same sequence will not be reproduced each time the Event Stream scenario is run.
- Pressing the “Run” button causes the Event Stream scenario to be executed.
- Pressing the “Stop” button when an event stream is executing will cause the Event Stream execution to terminate.

Event Stream Software Application in SIM

The software component of Event Stream feature in the SIM is:

- Responsible for parsing each event token from the scenario string passed down from the PC application via LAN, and taking an appropriate action such as generating a heart event or waiting for a pacing pulse.
- Implemented in Tcl script language and executed in the Tcl script interpreter.

Detailed Event Stream Language Description

Events

An event is either an output from the device (A or V pulse), or an output from the simulator (P or R wave). Simulated heart events can be intrinsic, or in response to a pacing pulse (evoked response). In the Event Stream language a set of text tokens is used to denote the events, (see **Table 1**).

An event can have associated properties, which can be optionally specified in Event Stream inline syntax. The event property is used to control the characteristic and behavior of an associated individual event.

Table 1 - Event Stream Event Symbols

Event Token	Definition
A, RA	right atrial pacing pulse
V, RV	right ventricle pacing pulse
LA	left atrial pacing pulse
LV	left ventricle pacing pulse
P, RP	right atrial intrinsic wave
R, RR	right ventricle intrinsic wave
LP	left atrial intrinsic wave
LR	left ventricle intrinsic wave
REA	evoked response on right atrium
REV	evoked response on right ventricle
LEA	evoked response on left atrium
LEV	Evoked response on left ventricle

Event property

Within an Event Stream scenario, the properties for any event can be modified by specifying a new value within a pair of square brackets ("[]") immediately following

the event token. An arbitrary number of qualified property values can be specified inside the brackets, and they can be in any order. The property parameters must be separated by a comma (“,”), if more than one parameter is specified.

Example: P [C=Haversine, H=3.0, W=20] or P[W=20, C=Haversine, H=3.0]

Intrinsic Event properties

An intrinsic event has three properties; amplitude, width, and waveform type. These properties and the values that can be assigned to them are defined in **Table 2** below.

Table 2 - Intrinsic Event Property

Property Name	Definition	Initial System Value	Property Value Range
H	Waveform amplitude value, negative amplitude generates wave with negative polarity	4.0 for P, LP; 8.0 for R, LR; 5.0 for REA, LEA; 8.0 for REV, LEV	Real , -40.0 to +40.0, in mV
W	Waveform width	20, for all P, LP, R, LR; 30 for REA, LEA; 40 for REV; 50 for LEV	Integer, 1 to 1000, in ms
C	Waveform type, shape, or morphology	“Haversine”	“Haversine” “Square” “Cenelec” “PDI” “Var_Cenelec” <string-constant>

Example:

P [H=3.0, W=30, C= Cenelec] 200 P[H=-5.0] 1000 P

- This example generates one P wave with amplitude of 3.0 mV, width of 30 ms, and configuration of Cenelec. The polarity of the wave is positive.
- After a 200 ms delay, the second P wave is generated with negative polarity, amplitude of -5.0 mV. The second P wave inherits the other properties from the first P wave, with width of 30 ms and shape of Cenelec.
- After a 1000 ms delay, the third P wave is generated with the properties inherited from the second P wave, amplitude of -5.0 mV with negative polarity, width of 30 ms and shape of Cenelec.

Pacing Event properties

Four pulse event properties are available; threshold, delay before evoked response event, evoked response event, and timeout period. These four properties are defined in **Table 3** below. The property values of a pulse event may be changed on each occurrence of the event. A value will remain in effect until a new value is specified.

Table 3 - Pacing Event property

Property Name	Definition	Initial System Value	Property Value Range
T	Threshold value	0.2 for A; 1.0 for LA; 0.8 for V, LV	real, 0.2 to 10, in V
D	Delay before evoked event	1	integer, 1 to 1000000, in ms
E	Evoked response event	NONE	<intrinsic-event> <intrinsic-event> [< intrinsic-event- property>+] NONE
TO	Timeout time	2000	integer, 0 to 1000000, in ms

Evoked Response properties

An evoked response is a heart event that is generated in response to a pulse event.

The pulse event properties, “T”, “D”, and “E”, are used to specify if an evoked response should be generated, and if so, when and what type of event it should be.

The Evoked Response event property can be updated pulse by pulse, if needed. Once the evoked response event property is set, subsequent events of the same kind inherit the same values until new values are set.

Example: RA [T=1.0, D=10, E=REA]

Sets the capture threshold of right the atrial chamber to 1.0 V, and the chamber’s evoked response is enabled. If an A pulse with amplitude greater than the threshold is detected, an atrial evoked response will be generated after a delay of 10 ms.

Timeout

The Timeout property optionally sets the timeout period in milliseconds to wait for a pulse event. The valid range is 1 – 1,000,000 ms (1 ms - 16 min 40 sec). The default value, if the property is not specified, is 2000 ms, or 2 seconds. A timeout marker “~” will be sent back to the STT (PC) application if the paced event is not detected within the timeout period.

Example: RA [TO=10000]

The number on the right side of the equal sign “=” specifies the timeout time to wait for the event. In this example, the time timeout value is set to 10000 ms, or 10 sec. If waiting on an A pulse times out after 10 sec, then a timeout marker “~” will be sent back to the application program, and the event sequence execution will continue.

Example: 10 (A [TO=9000, D=25, E=REA] V[D= 30, E=REV])

The scenario in this example waits up to 9000 milliseconds (9 seconds) for an A pulse (in the right atrium). When detected delay 25 milliseconds and generate Right Atrial Evoked Response. If no pace pulse is detected within the timeout period, send the timeout marker and don't generate an evoked response. Wait for a V pulse (in the right ventricle, and with the default timeout), then delay 30 milliseconds before generating a Right Ventricle Evoked Response. If no V pulse is detected within the timeout period, don't generate an evoked response. Repeat the whole sequence a total of 10 times.

STT Events trace, with pacing pulses:

319 B	338 A	27 p	144 V	31 r	626 A	26 p	146 V	31 r	625 A
26 p	146 V	31 r	625 A	26 p	146 V	31 r	625 A	26 p	146 V
31 r	626 A	26 p	146 V	30 r	626 A	26 p	146 V	31 r	625 A
26 p	146 V	31 r	653 A	26 p	145 V	31 r	653 A	26 p	146 V
31 r	0 E	653 A	172 V	683 A	172 V	684 A	172 V	484 S	

Sequence

A sequence in Event Stream is an autonomous session of one or more events, and/or operations. It is composed of one or more string tokens, enclosed by parenthesis and delimited with the semicolon character ‘;’. A sequence specifies the sequential order of events, intervals and operations to be executed. At least one blank space must be used to separate consecutive events, operations, and a delay within a sequence.

Example: (P 120 R 880);

Repeat Count

The <repeat-count> of a sequence is a number specifying the number of time to repeat the sequence, in the range of 1 – 1,000,000. The default value is 1. Events in a sequence must be included in a pair of parentheses “()”, with the sequence repeat count preceding the left parenthesis “(”. The repeat count and the left parenthesis “(” should be separated with a blank space.

Example: 5 (RV 500 RP);

Scenario

A scenario is the logical concatenation of all sequences added together. The complete logical sequence is called an Event Stream scenario.

- An Event Stream scenario can be composed of one or more sequences.
- When a scenario is executed, it starts with the first sequence and ends with the last sequence in the scenario, and begins with a marker “B” and ends with a marker “E”.

Example: 5 (RV 700 RP); 2 (RV 500 RP);

Operation

An operation performs an action or activity that can alter some of the heart or the device system properties, or change the status of the STT. An operation can happen at any time in the Event Stream sequence.

Set Lead Impedance Operation

One operation is defined for lead impedance programming. An operation performs an action or activity that can alter some of the heart or the device system properties, or change the status of the STT. An operation can happen at any time in the Event Stream sequence.

The “I” is a single character (uppercase “i”) string specifying a set lead impedance operation, as defined in **Table 4**. The set lead impedance operation will set the impedance value for a specific lead.

Table 4 - Lead Impedance Operation Token

Operation Token	Definition
I	Set lead impedance

Four operation properties, for the lead impedance associated with each of the four heart chambers, are the valid properties which can be updated.

The four operation properties are defined in **Table 5**.

Table 5 - Lead Operation Property

Property Name	Definition	Initial System Value	Property Value Range
RA	Right Atrial chamber lead impedance	500	Integer, 150 to 6500 in 50 Ohms steps; O – open; S - short
RV	Right Ventricle chamber lead impedance	500	Integer, 150 to 6500 in 50 Ohms steps; O – open; S - short
LA	Left Atrial chamber lead impedance	1000	Integer, 150 to 6500 in 50 Ohms steps; O – open;

LV	Left Ventricle chamber lead impedance	500	S - short Integer, 150 to 6500 in 50 Ohms steps; O – open; S - short
----	---	-----	---

The keyword tokens “RA, RV, LA, LV” are used to specify the property name (chamber) associated with the operation, not ‘V’ or ‘A’ pulses in this context.

For the set lead impedance operation, the property value is the integer value specifying the lead impedance in Ohms, in the range of 150 – 6500, if it is set to a discrete value, with increment and decrement step value of 50 Ohms.

If the lead impedance is set to open or short, use value of “O” or “S”. Setting the lead impedance of a chamber to short or open will disable the capability of the heart/SIM to detect the pace activities on the chamber.

Example: I [RA=1000, RV=3000, LA=O, LV=S]

Sets the lead impedance of Right Atrium Chamber to 1000 Ohm, the impedance of Right Ventricle Chamber to 3000 Ohm, the lead impedance of Left Atrium Chamber to Open, and the impedance of Left Ventricle Chamber to Short.

Value Variability

Some numeric Event and Event Property values in a sequence can be incremented or

decremented by fixed amount or random value on each sequence cycle.

Events that can be varied:

Delay Interval between events

Event properties that can be varied:

Heart Event Property – Amplitude, Width

Device Event Property – Evoked Response Delay Interval

When a qualified numeric Event or Event property value is specified with a variant in a sequence, an actual value is computed according to the variant type, and applied to the Event or Event Property on each sequence cycle. There are three different types of variant: Increment by a fixed amount, Decrement by a fixed amount, Change by a random value within the range.

Variant

The variant <variant> can denote a random variation range (“+”), or an increment amount (“+”), or a decrement amount (“-”) for the preceding <base-value>. The variation is applied consecutively to each cycle of a sequence, excluding the initial cycle.

Table 6 - Variant Signature Token

Signature Token	Designation
+	Sequential increment
-	Sequential decrement
+-	Sequential random in a range

Some numeric Event and Event Property values in a sequence can be incremented or decremented by fixed amount or random value on each sequence cycle.

Increment Variant

The $\langle \text{base-value} \rangle + \langle \text{variant} \rangle$ specifies that the value in a sequence start with the base value of $\langle \text{base-value} \rangle$, and increases by $\langle \text{variant} \rangle$ amount each time the sequence is executed.

Table 7 - Increment Variant

Sequence Repeat #	Actual Value
1st	$\langle \text{base-value} \rangle$
2nd	$\langle \text{base-value} \rangle + (\langle \text{variant} \rangle * 1)$
.....	
the N-th	$\langle \text{base-value} \rangle + (\langle \text{variant} \rangle * (N-1))$

Example: /* Increment P to R interval by 10 ms on each cycle */

3(P 400+10 R);

The resulting output event sequence is:

P 120 R 880 P 130 R 880 P 140 R 880

Decrement Variant

The $\langle \text{base-value} \rangle - \langle \text{variant} \rangle$ specifies that the value in a sequence start with the base

value of <base-value>, and decreases by <variant> amount each time the sequence is executed.

Table 8 - Decrement Variant

Sequence Repeat #	Actual Value
1st	<base-value>
2nd	<base-value> + (<variant> * 1)
.....	
the N-th	<base-value> + (<variant> * (N-1))

Example:

```
/* Increment P waveform amplitude by 0.2 mv on each cycle */
10 ( P[H=11.1 + 0.2]);
```

Random Variant

The <base-value>+-<variant> specifies that the actual value in a sequence is a random numbers with the variation range between the values of (<base-value>-<variant>) and (<base-value>+<variant>), each time the sequence is executed. The value of the <variant> must be less than that of the (<base-value> – <variant>). By default, the random delays are truly randomized (not repeatable). A pseudorandom (repeatable) series can be generated by specifying the random number seed.

Example: Increment and Decrement time interval by fixed amount

```
10 ( P 150-10 R 850-20); 10 (P 50+10 R 850+20);
```

Rate ramps up and down. For 10 cycles, generate a P wave followed by a 150

millisecond PR interval that reduces by 10 milliseconds on each cycle, and then generate an R wave followed by an RP interval of 850 milliseconds that reduces by 20 milliseconds each cycle. After completing the first 10 cycle sequence, repeat the second sequence 10 times, increasing the PR and RP intervals on each cycle.

STT Events trace:

>9999 B	1 P	151 R	851 P	140 R	830 P	130 R	810 P	120 R	790 P
110 R	770 P	100 R	750 P	90 R	730 P	80 R	710 P	70 R	690 P
60 R	670 P	51 R	650 P	60 R	630 P	70 R	610 P	80 R	910 P
90 R	930 P	100 R	950 P	110 R	970 P	120 R	990 P	130 R	1010 P
140 R	1030 E								

Key Words

Certain characters, words, and symbols are reserved in the Event Stream Language.

These are described in Table 9, below. In addition, Haversine, Square, Cenelec, PDI,

Var_Cenelec are defined names of the embedded intrinsic wave forms. NONE is

defined as a keyword for the purpose of disabling evoked response of a heart chamber

when used as a property value assigned to pulse event property name E.

Table 9 - Event Stream Language Key Words

Key Word Token	Definition
;	Sequence separator
,	Property separator
()	Enclose an event sequence
[]	Enclose properties for the preceding event or operation
=	Assign property value on the right to the property named on the left
+, -, +- P, RP, R, RR, LP, LR, REA, REV, LEA, LEV A, V RA, RV, LA, LV I H, W, C T, D, E, TO Haversine, Square, Cenelec, PDI, Var_Cenelec NONE	Delay variation signatures Intrinsic (heart) events Pulse (device) events Pulse events, or property names Set lead impedance operator Intrinsic event property names Pulse event property names Names of built-in waveforms Usage E=NONE (disable heart chamber evoked response)

The key words have a specific meaning in the language, and should not be used for

any other purpose. All key word characters are upper case, except the embedded

waveform names. All key word characters are also case sensitive including embedded waveform names.

Chapter 5: Discussion

When the Event Stream feature is delivered to customers in 2007, after six months of development, the new feature was instantly adopted as the primary heart rhythm scenario generation feature in the STT by pacemaker or ICD firmware testing group. The following key factors are identified as strengths in the Event Stream by customers:

1. Simple and Intuitive – The Event Stream allows users to utilize the existing event symbols (e.g. P and R) to create a desired scenario.
2. Precise and Repeatable – The Event Stream allows users to specify the precise interval between events in the scenario. Once developed, a scenario can be saved in the file and executed over and over again in predictable manner.
3. Flexible and Powerful – The versatility of text based Event Stream language allows users to produce simple to very complex scenarios in a few key strokes.

As of today, the heart rhythm scenarios in the pacemaker and ICD firmware test scripts are exclusively developed using the Event Stream feature. The following is an example of a firmware test script containing Event Stream scenarios:

While the Event Stream feature provides many powerful tools to satisfy needs of majority of users, one weakness in the feature was recognized among advanced users – ability to handle asynchronous heart events. Based on the current design of the Event Stream feature, each event in the Event Stream scenario is executed in order as specified in the scenario. For example, the scenario “RP 120 RR” processes each

event in order – generate a right atrial event, wait for 120 ms and generates a right ventricular event. While above category of scenarios covers the need of most users, some advanced users require left atrial and ventricular events to be simulated asynchronously as right atrial and ventricular events are generated.

Chapter 6: Conclusion Remarks

The Event Stream feature in the STT was initially created upon request of the device firmware test group in St. Jude Medical to meet their need of a simpler and more powerful way than the Heart Simulation (HS) feature to create heart rhythm scenarios interacting with a device. The simple text string based Event Stream language syntax utilizing existing event symbols allows users to quickly learn the new language, and create and run complex scenarios in matter of few key strokes. Because of its simplicity and power, the Event Stream feature was immediately adopted as the main STT feature to generate heart rhythm scenarios not only in device firmware test group but also in firmware development group for every new pacemaker and ICD device projects.

There are two main enhancement areas in the Event Stream feature in the upcoming years:

1. Add new syntax – Users ask new syntax for rest of simulations in the STT such as Noise Signal simulation, high voltage shock event detection event to be added in the Event Stream language.
2. Asynchronous event simulation – Advanced users ask ways to simulate asynchronous events – e.g. simulation of right and left atrial events independently and concurrently.

Acronyms and Definitions

Bradyarrhythmia/ Bradycardia	Slow heart rate, which is usually below 60 beats per minute
Defibrillator	Medical device implanted to the heart via defibrillator leads which treats tachyarrhythmia
ES	Event Stream
HS	Heart Simulator
ICD	Implantable Cardioverter Defibrillator
LAN	Local Area Network
Pacemaker	Medical device implanted to the heart via pacemaker leads which treats bradyarrhythmia
PC	Personal Computer
SBC	Single Board Computer
SIM	Simulation Interface Module
SJM	St. Jude Medical
STT	Simulation Test Tool
Tachyarrhythmia/ Tachycardia	Fast heart rate, usually above 100 beats per minute
Tcl	Tool Command language

Bibliography

¹ Arthur C. Guyton, M.d., and John E. Hall, M.D. TEXTBOOK OF MEDICAL PHYSIOLOGY TENTH EDITION. W.B. Saunders Company, 2000.

² St. Jude Medical Website. <<http://www.sjm.com>>

³ Seymour Furman, M.D., David L. Hayes, M.D., David R. Holmes, Jr., M.D., A PRACTICE OF CARDIAC PACING Third Edition. Futura Publishing Company, Inc., 1993