

MERLIN.NET AUTOMATION OF EXTERNAL REPORTS  
VERIFICATION PROCESS

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

Gabriel John Wettlaufer

January 2010

© 2010

Gabriel John Wettlaufer

ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: MERLIN.NET AUTOMATION OF EXTERNAL  
REPORTS VERIFICATION PROCESS

AUTHOR: Gabriel John Wettlaufer

DATE SUBMITTED: January 2010

COMMITTEE CHAIR: Lily Laiho, Dr. and Professor

COMMITTEE MEMBER: Robert Crockett, Dr. and Professor

COMMITTEE MEMBER: Kristen O'Halloran Cardinal, Dr. and Professor

## ABSTRACT

### MERLIN.NET AUTOMATION OF EXTERNAL REPORTS

#### VERIFICATION PROCESS

Gabriel John Wettlaufer

Merlin.net™ Patient Care Network is a St. Jude Medical product that is used for remote patient management. The basic concept of Merlin.net is to allow the physician to view patient device follow-up information as well as general patient and device information on a web application. The Merlin.net system also interfaces with the patient and will send them notification if they miss a follow-up. All device information will be collected automatically while the patient is sleeping. This information is sent through a telephone line to a Merlin.net server to process a report package and display the collected information on the Merlin.net web application. The Merlin.net verification team ensures that all reports generated by the Merlin.net servers are processed and outputted correctly. There are currently 296 device parameters supported by Merlin.net, and the manual extraction and comparison of the expected parameter values takes several hours for each patient follow-up session. Currently there are 250 patient follow-up sessions used for verification testing. Each new release will continue to create additional patient follow-up sessions. Merlin.net releases are approximately 6 months apart, and each new release adds approximately 30-50 new patient follow-up records to support the new devices. In order to meet aggressive project deadlines, while ensuring that the Merlin.net system is processing and outputting patient follow-up data correctly, it is necessary to come up with an automated process to verify the contents of the processed data is correct. This will save a tremendous amount of time as well as improve on the quality of the

verification process by eliminating human error and rework. It is critical for patient safety that the patient device follow-up information is processed and outputted correctly.

In this thesis an automated process was developed to verify the correct content of the Merlin.net server generated reports for each patient follow-up session. This process leveraged different tools and scripting languages to achieve automation. TDE (Test Development Environment) tool was used to extract the device parameters from the patient follow-up sessions. The TDE script was written to extract the desired parameter values from the patient follow-up session and automatically populates parameters in a device parameters spreadsheet. Once all the device parameter values are extracted in the spreadsheet, they are passed through a set of mapping rules, which form the expected values. The mapping rules were implemented as VBA (Visual Basic for Application) macros, one macro for each report. The VBA macros write the expected values back to the spreadsheet to form an “expected values spreadsheet”. The patient follow-up session is then sent to the Merlin.net server to process, which generates a processed patient follow-up session that contains a reports package in .zip format. A perl script was then written to compare the parameter values in the Merlinet.net generated reports with the corresponding expected values from the expected values spreadsheet. The perl script generates a comparison report displaying the discrepancies between the actual and the expected values.

Keywords: Merlin.net, St. Jude Medical, Cardiac Device, Remote Follow-up, Test Automation, ICD, pacemaker

# TABLE OF CONTENTS

	Page
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
I. INTRODUCTION .....	1
St. Jude Medical Implantable cardiac devices .....	1
History of Manual Remote Follow-up with Housecall <sup>TM</sup> Plus System .....	5
Automated Remote Follow-up with Merlin@home <sup>TM</sup> .....	10
Merlin.net <sup>TM</sup> Patient Care Network .....	12
Merlin.net <sup>TM</sup> Patient Data Processing .....	15
Merlin.net Verification .....	17
II. OBJECTIVE .....	21
III. METHODS AND MATERIALS.....	23
TDE Implementation .....	23
Expected values spreadsheet.....	25
VBA Implementation.....	28
Perl comparison script .....	29
IV. RESULTS .....	33
V. CONCLUSION AND FUTURE WORK .....	37
REFERENCES .....	40
APPENDIX A: TDE SCRIPT .....	43
APPENDIX B: VBA MACROS.....	44
APPENDIX C: PERL COMPARISON SCRIPT .....	45

## LIST OF TABLES

Table	Page
<i>Table 1: Example Parameters supported in XPSM generated reports.....</i>	19
<i>Table 2: Average execution time for each implemented script.....</i>	33

## LIST OF FIGURES

Figure	Page
<i>Figure 1: Implanted dual-chamber ICD device with leads connected to the right atrium and ventricle.....</i>	4
<i>Figure 2: The figure on the left is a Housecall Plus Receiver used by the physician. The figure on the right is a Housecall Plus Transmitter located at the patient's home. ....</i>	8
<i>Figure 3: Remote follow-up with the Housecall Plus System.....</i>	9
<i>Figure 4: Example of FastPath Summary screen from Housecall Plus follow-up.....</i>	9
<i>Figure 5: Example of a Merlin@home Unit.....</i>	11
<i>Figure 6: Example Patient enrollment page on the Merlin.net website.....</i>	13
<i>Figure 7: Diagram of the automated external report verification process.....</i>	24
<i>Figure 8: Example of expected values spreadsheet of external files.....</i>	27
<i>Figure 9: DDT processed file contents.....</i>	29
<i>Figure 10: Perl script execution for VA report.....</i>	31
<i>Figure 11: Example of comparison report for VA report.....</i>	32
<i>Figure 12: Time Comparison of manual and automated process. ....</i>	35

## **I. INTRODUCTION**

### **St. Jude Medical Implantable cardiac devices**

The heart is an organ consisting of four chambers that pump blood. The two upper chambers are divided into the right and left atria, and the two lower chambers divided into the right and left ventricles. The right atrium receives venous blood (oxygen-poor blood) from the body and pumps it into the right ventricle [1]. The right ventricle pumps the blood to the lungs, where gas exchange occurs and blood becomes saturated with oxygen [1]. The oxygen-rich blood from the lungs then travels to the left atrium and into the left ventricle, which delivers the oxygen-rich blood to the rest of the body [1]. In addition to oxygen, the blood carries other nutrients (glucose, electrolytes, etc.) to the organs [1].

In order to keep a body healthy, the heart must deliver a sufficient amount of blood to the body. As a pump, the heart is most efficient in delivering blood when functioning within a certain heart rate range [2]. Normally, the heart's natural pacemaker called the SA node (a special tissue located on the right atria wall), keeps the heartbeat (heart rate) in the normal range [1, 2]. The SA node generates electrical signals along special conduction tissues on the walls of the atria and the ventricles [1, 2]. These electrical signals cause the heart muscles to contract and pump blood in an orderly and efficient manner [1].

Normally, the heart beats between 60 and 100 beater per minute in an adult at rest. There are two types of abnormal heart conditions: Tachycardia and Bradycardia. Tachycardia is condition where the heart beats at an abnormally faster rate greater than

100 beats per minute: Bradycardia is a condition where the heart beats at an abnormally slower rate less than 60 beats per minute [3].

Two common life-threatening tachycardias are ventricular tachycardia and ventricular fibrillation [4]. Ventricular tachycardia is a rapid regular rhythm caused by electrical signals originating in the ventricle [4]. Ventricular tachycardia can decrease blood delivery by the heart and cause low blood pressure [3]. Ventricular tachycardia can also progress to a more serious heart rhythm called ventricular fibrillation [3].

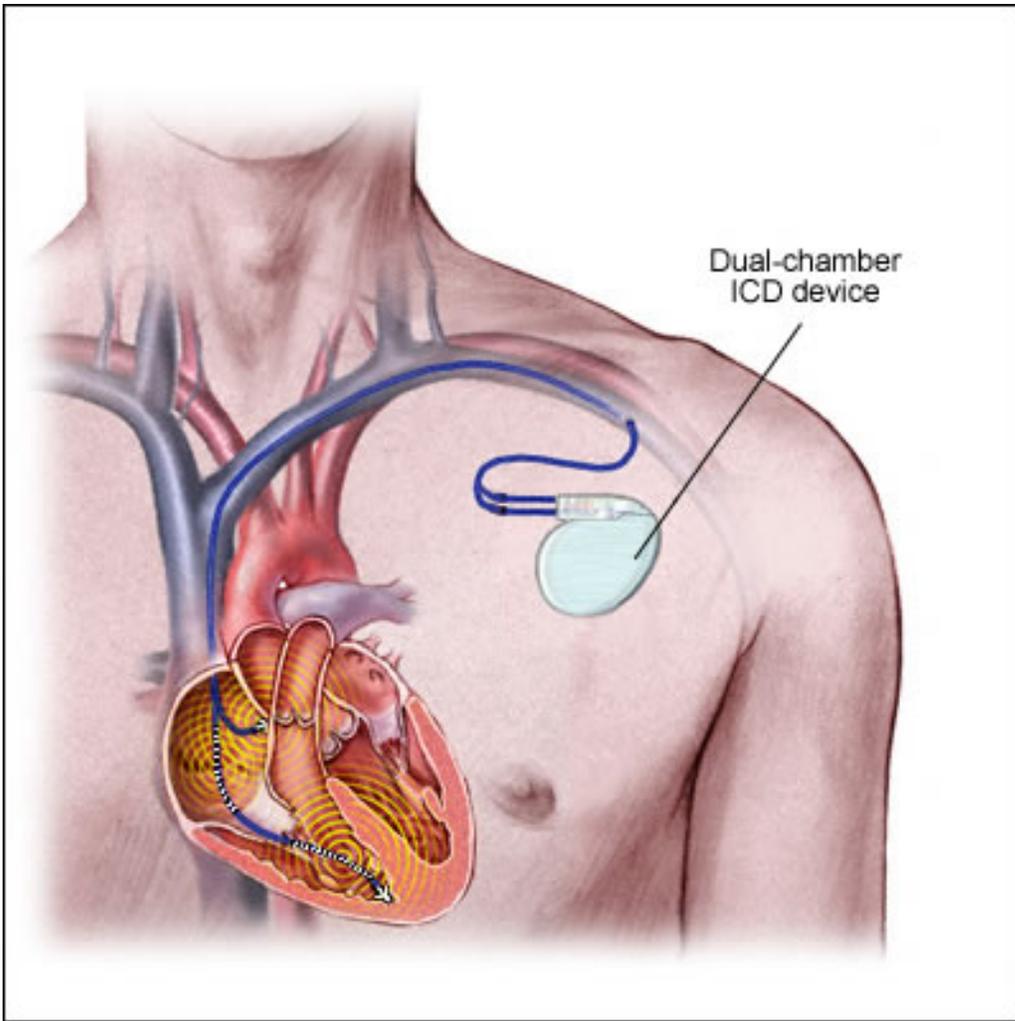
Ventricular fibrillation is an irregular rhythm, which is a result of multiple rapid and chaotic electrical signals firing from many different areas in the ventricles [3]. A heart undergoing ventricular fibrillation is in a state of standstill called cardiac arrest [3]. The heart muscles quiver and cease pumping [3]. The result is a halt in the delivery of blood to the body. Irreversible brain damage occurs within minutes of the onset of ventricular fibrillation and leads to death.

Bradycardia is a slower than normal heart rate. Bradycardia is a sign of a problem with the heart's electrical signals [2]. It usually indicates either that the heart's natural pacemaker is not working appropriately or that the electrical pathways of the heart are disrupted [1]. In severe forms of bradycardia, the heart beats so slowly that it does not pump enough blood to meet the body's needs [1]. This can lead to a list of symptoms including, chest pain, fainting, light-headedness, shortness of breathe, fatigue, and can be life threatening if not treated [1].

The Cardiac Rhythm Management Division (CRMD) of St. Jude Medical designs and manufactures implantable cardiac devices to treat bradycardia (debilitating slow heartbeat), tachycardia (life-threatening fast heartbeat), and other cardiac rhythm

disorders. A pacemaker is an implanted medical device that uses electrical impulses, delivered by leads contacting the heart muscles, to regulate the beating of the heart [2]. The primary role of a pacemaker is to maintain an adequate heart rate, either because the heart's native pacemaker is not fast enough, or there is a block in the heart's electrical conduction system [1]. An implantable cardioverter-defibrillator (ICD) is an electrical impulse generator which is implanted in patients who are at risk of sudden cardiac death due to ventricular fibrillation [5]. The device is programmed to detect cardiac arrhythmia and correct it by delivering a jolt of electricity [1].

Implantation of an ICD and pacemaker are very similar. The 2 to 3 hour procedure is considered minor as it does not involve major heart surgery [1]. Patients are typically sedated during the procedure [1]. A local anesthetic is injected under the skin over the area where the ICD or pacemaker will be placed, usually in the right or left upper chest near the collarbone [1]. The lead is then inserted into a vein located in the upper chest near the collarbone [1]. The tip of the lead is placed on the inner wall of the heart with the visual guidance of x-rays. If there is more than one lead, the process is repeated [1]. The other end of the lead (or leads) is connected to the defibrillator or pacemaker unit, which is then inserted under the skin at the incision site [1]. Because there are no nerve endings inside the blood vessels and the heart, the patient usually does not feel the placement of the lead(s) [1]. See Figure 1 for an example of an implanted dual-chamber ICD device with leads connected to the right atrium and ventricle.



*Figure 1: Implanted dual-chamber ICD device with leads connected to the right atrium and ventricle.*

The patient's heart rhythms, pulse, and blood pressures are routinely monitored [1]. The doctor may check or adjust the settings on the defibrillator or pacemaker according to the measurements taken during this time [1]. These settings are changed using a Programmer, which is a portable computer that allows the physicians to view the activity collected by the device on a graphical user interface (GUI). The programmer communicates with the device using a telemetry wand. By placing the wand outside of the body and over the implanted device, it can read and display the device information on

the programmer screen. The physician can program different sensing and therapy thresholds on the implanted devices for each patient's individual needs. The nurses also periodically examine the incision over the implantation site for bleeding, redness, or other signs of infection [1]. It is normal to feel some pain over the incision for 1-2 weeks [1]. Medications are usually given to alleviate pain [1]. Patients are typically discharged from the hospital in 1- 2 days after the procedure [1].

### **History of Manual Remote Follow-up with Housecall™ Plus System**

Patients with an implanted cardiac device typically require an office visit follow-up to their electrophysiologist approximately every three months. At each follow-up visit, the following data is acquired using the device programmer: patient history (focusing on arrhythmia-related symptoms, device therapies and device-related problems), device parameters, stored episodes, device interrogation testing, diagnostic data (data collected performing different tests to diagnose a particular heart condition), and measured data. Measured data includes lead impedance test and battery voltage/charge time. The device programmer is a powerful computer with a telemetry wand that reads the information from the device, and displays it on the programmer screen. The programmer screen is an LCD touch-screen that allows the physician to (1) evaluate the device information and (2) adjust the settings on the device if necessary.

A remote patient follow-up collects the same information as an office visit follow-up, except the device information is collected from an external device located in the patient's home. This information is transferred through a telephone line to an external server for the electrophysiologist to view. Patients located far from their electrophysiologists office benefits more by remote follow-up. Both routine "visits" and

emergency "post-shock visits" are accomplished faster with no loss of accuracy via telephone. In cases where the patient must go to the Emergency Room after an event, the data gathered via remote transmission can be sent to the patient's electrophysiologist, which can suggest proposed therapy to the ER or local physicians caring for the patient. Telephone monitoring can certainly substitute for some office visits. The benefit of telephone monitoring enables the physician to examine the patient and perform pacing threshold measurements with increased speed and convenience for both the patient and monitoring doctor.

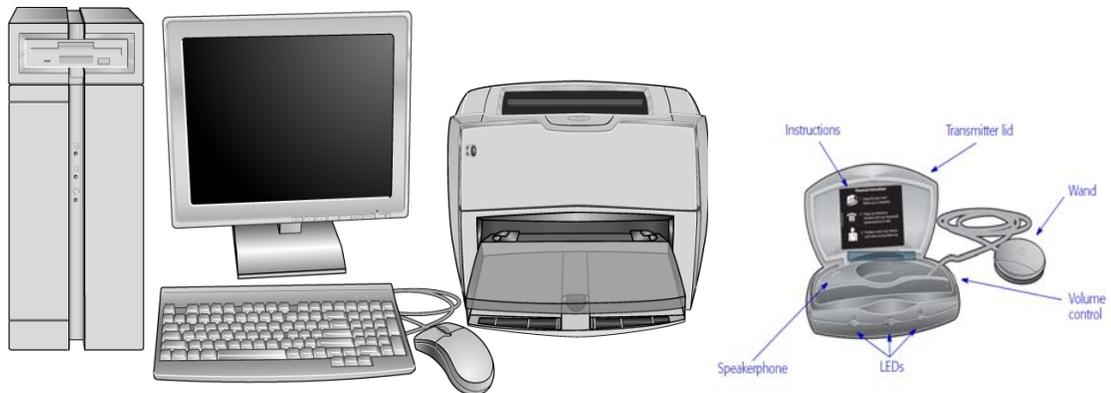
The Housecall™ Plus Remote Follow-Up System is an aid to patient follow-up, allowing remote follow-up sessions with patients having a St. Jude Medical ICD. Pacemakers are not supported by the Housecall Plus system, because the Housecall Plus system was originally developed by Ventritex, Inc, a company acquired by St. Jude Medical in 1996 that designed and manufactured ICDs and their remote follow-up system. Because the transmitter is used remotely, this can help decrease the number of office visits for patients and may also reduce the amount of time that passes before an episode and consequent therapy are evaluated by a physician.

The Housecall Plus Remote Follow-Up System conducts a patient follow-up session through the remote interrogation of a patient's device. A physician can:

- Interrogate a device
- View the interrogation data, including:
  - bradycardia parameters
  - tachycardia parameters
  - diagnostic data

- stored episodes
  - measured data
- Print follow-up session reports

The Housecall Plus system consists of a receiver and a transmitter. The receiver is a stand-alone, PC-based system that works in conjunction with the transmitter to allow follow-up monitoring of ICDs. The Receiver can be located at the monitoring center, follow-up clinic, physician's office, or hospital, and allows the physician to conduct a patient follow-up session through the remote interrogation of a patient's device. The user can interrogate the device, view the interrogation data, print follow-up session reports, and send the reports to the server. The Receiver's user interface has three components: the mouse, keyboard, and the monitor. The Housecall Plus Transmitter is about the size of a telephone answering machine and consists of a telemetry wand, which reads information from the patient's ICD, and a data model which transmits the patient data to a server. The transmitter telemetry wand uses inductive telemetry to retrieve information from the device. Inductive telemetry uses two coils, one in the device and the other located on the transmitter wand, with mutual inductance between these coils to communicate information from the device to the transmitter. The transmitter retrieves data from the patient's ICD and sends it via modem to the Receiver. The transmitter cannot be used alone. All commands to operate the transmitter come from the Receiver. See Figure 2 for a diagram of the Housecall Plus system.



*Figure 2: The figure on the left is a Housecall Plus Receiver used by the physician. The figure on the right is a Housecall Plus Transmitter located at the patient's home.*

A follow up requires the physician to call the patient's home where their transmitter is located. The patient speaks directly to the physician and is instructed to open the lid on their transmitter and place the telemetry wand over their implanted device. The physician will press the located device button on the receiver screen to ensure that the telemetry wand is located in the correct position to retrieve data from the device. The result of a successful locate device command is a green checkmark on the receiver screen for the physician, and the transmitter model will chirp to notify patient. After the device is located the physician will press the interrogation button on the receiver screen, at which point the transmitter retrieves the stored data on the device and displays it on the receiver screen. See Figure 3 for a diagram of a Remote follow-up with the Housecall Plus System. The figure shows the simple flow of a Housecall Plus Remote follow-up. In this flow the operator uses the Housecall Receiver to retrieve the device information via the Housecall Transmitter. This information is sent to the St. Jude Medical (SJM) server where the information is processed and stored for the physician to view at his or her convenience.

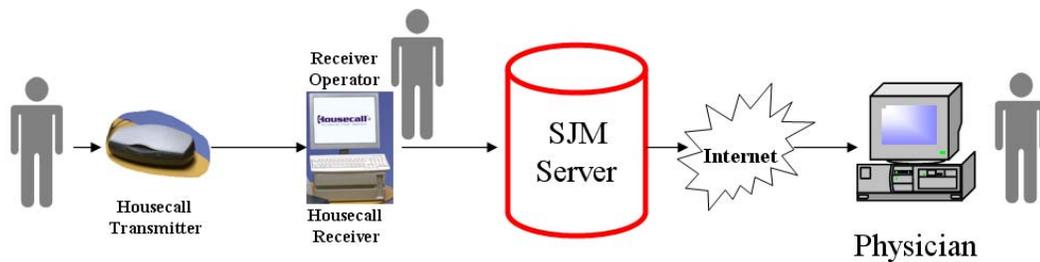


Figure 3: Remote follow-up with the Housecall Plus System

The physician can conduct a standard follow-up, retrieve episode and diagnostic data, view information about the stored device data, print PDF reports, and upload the reports to a server to view at a later date. The standard remote follow-up takes approximately ten minutes. Please see Figure 4 for an example of the FastPast Summary screen viewed upon initial interrogation of the device on the Housecall Plus receiver screen.

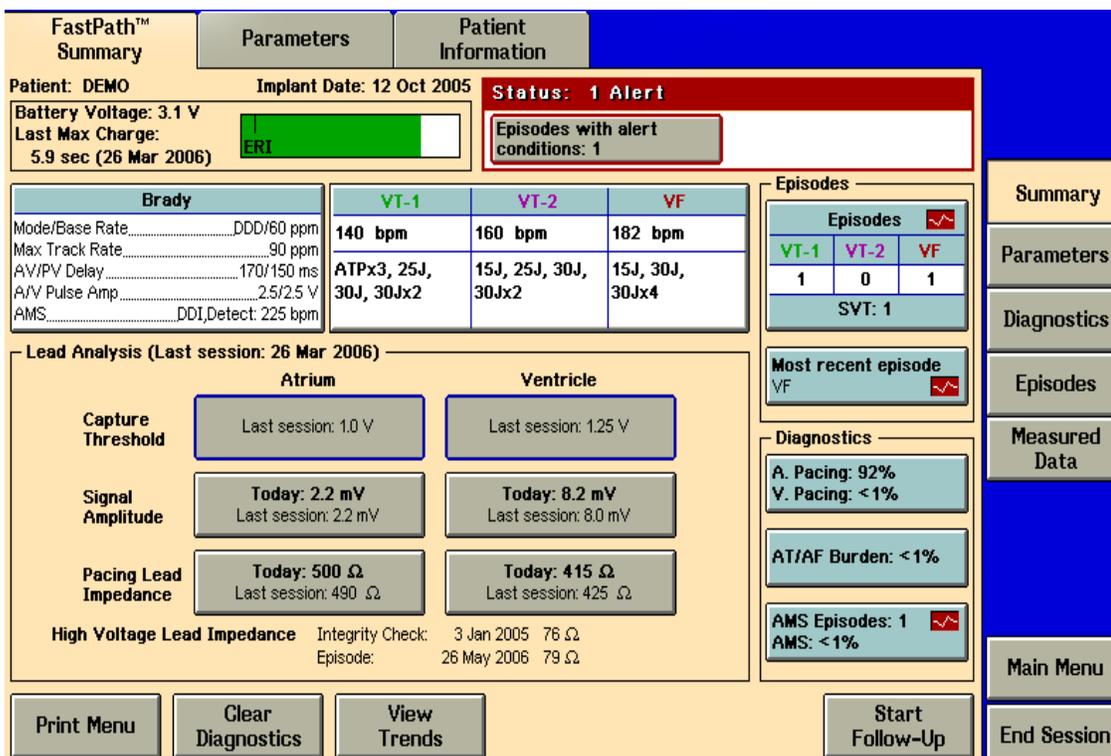


Figure 4: Example of FastPath Summary screen from Housecall Plus follow-up

## **Automated Remote Follow-up with Merlin@home™**

The Merlin@home™ transmitter is the next generation remote patient follow-up system for St. Jude Medical. The Merlin@home transmitter uses RF wireless technology that remotely monitors patients' implanted cardiac devices. This system also addresses the situation where patients with implanted cardiac devices must visit doctors' offices several times per year to have their device performance checked and has expanded the support to both ICDs and pacemakers. The Merlin@home transmitter's wireless technology allows the devices to be automatically checked, and conduct follow-up procedures without direct patient involvement. The patient need only remain within range of the transmitter while it reads his or her device. Patients also may initiate data transmissions as instructed by their physicians. The Merlin@home transmitter is transportable and can be set-up wherever a standard phone line is available, typically by the bedside for data transmission while the patient sleeps. During a Merlin@home session, the transmitter reads the data from the implanted cardiac device and sends it to a server where the physician can view it. The Merlin@home transmitter has removed any patient or physician manual interaction from the follow-up process. The information is automatically collected, processed, and uploaded to a server for the physician to view at their convenience. After the information from the follow-up is reviewed, the physician can decide if the patient needs to come in for an office visit to modify the settings on the device, or if no modifications are required they simply wait for the next follow-up. The follow-up information is archived and stored in the server and saved under the follow-up date and timestamp, so all follow-ups are available to view at a later date. The

Merlin@home replaces the Housecall Plus system for newer St. Jude Medical ICDs and pacemakers. See Figure 5 for an example Merlin@home unit.



*Figure 5: Example of a Merlin@home Unit*

The collected data includes:

- The model and serial number of implanted device
- Device settings
- Activity since last follow-up session
- Battery status
- The transmitter can also conduct an implant check, which checks the implant's status

## **Merlin.net™ Patient Care Network**

The Merlin.net™ Patient Care Network (PCN) system is an advanced online patient management system designed to improve patient care through enhanced cardiac monitoring, and to optimize clinic resources by providing timely device and disease management data through a single point of access. The Merlin@home transmitter works in conjunction with the St. Jude Medical data management system, Merlin.net, to provide complete remote care service for patients and their physicians. Each patient is registered to a clinic's web page where the patient's device model, serial number, and personal information are entered and stored. Each clinic is given a username and password to the Merlin.net website where all clinic and patient information is stored. These log-in credentials allow the clinic to enroll new patients, view existing patients, and view follow-up information for all patients enrolled in the clinic. See Figure 6 for an example of the patient enrollment page on the Merlin.net website.

Recent Transmissions | **Patient List** | Clinic Administration | Reports & Tools | My Preferences | Help | Exit

Here's Dora Explorer's Patient Profile. What would you like to do?

**Patient List**  
Patient profile

See  
▶ Patient profile

Do  
▶ Enroll a patient  
▶ Edit a patient  
▶ Release a patient  
▶ Re-activate a patient

**Arrhythmia & Device Mgmt**

Total	0
Alerts	0
Unviewed	0
Messages	0
Import progress	0%

R. Bhatia  
ST. JUDE MEDICAL

**Patient & Device Data** | Baseline Clinical Data | Follow-up Schedule | DirectAlerts™ Notification | Medical Team

**Patient device data**

Serial #: 6543 | Device name: Current@+ DR RF, 22 | Implant Date: 01-26-2009

**Patient address & phone**

First name: Dora | Middle name: the | Last name: Explorer  
 Gender: Female | Date of birth: 09-10-1966 | Race:  
 Patient ID: | Clinic location: Newfoundland | Patient prefers: English (US)  
 Address 1: | Country: USA | City: | State/Province: | Zip/Postal code: |  
 Address 2: | City: | State/Province: | Zip/Postal code: |  
 Address 3: | State/Province: | Zip/Postal code: |  
 Merlin.net™ number: 756396990 | Zip/Postal code: |  
 Primary phone: 1 | Phone Number: |  
 DirectCall™ method: |

**Notify Patient** | **Transmitter** | **Patient consent**

Between: 09:00 AM | And: 04:00 PM | Order transmitter: No | Patient data collection? No  
 Patient has transmitter: Yes

Transmitter Setup | Emergency Contact | View Leads | Cancel | Close | Print Profile | Next

Figure 6: Example Patient enrollment page on the Merlin.net website

Each patient is identified by his or her unique device model and serial number combination. From the web application the physician has the capabilities to:

- Set up patient follow-up schedule. The physician can adjust the Merlin@home follow-up schedule from the Merlin.net web application. There are two follow-ups each day and they are typically selected at 10pm and 5am when the patient is sleeping. This schedule can be adjusted to accommodate the patients sleep pattern. The Merlin@home transmitter will conduct follow-ups according to this schedule.

- Set critical alerts. The physician can select these critical alerts from a checklist on the Merlin.net web application. These alerts are deemed critical by the physician and the web application will notify the physician if the selected alerts are encountered during a follow-up. Examples of some critical alerts are: Tachy Therapy Disabled, Charge Time Limit Reached, Device Reset, Device at End of Service, Lead Impedance out of Range, and Episodes with Alert Conditions.
- Set alert notification destinations. The physician can choose from a several different notification destinations if a critical alert is encountered in a follow-up. Email, phone, text message, and fax are the different ways the physician can choose to be notified.

After the follow-up is complete the physician can view the different PDF reports and the collection of information under the clinic's website login. The Merlin.net system also interfaces with the patient and will send them notification if they miss a follow-up. All device information will be collected automatically while the patient is sleeping (the follow-up schedule is configurable by the clinic) with a Merlin@home transmitter. This information will be sent through plain old telephone lines (POTS) to a SJM Device Data Translator (DDT) server to process a report package, and display the collected information on a web application. This will allow the physician to view the collected data at their convenience. Device information can also be collected, transmitted, and processed through the Merlin.net system with an In-Clinic follow-up. During an In-Clinic follow-up the physician will interrogate the device using the Merlin Programmer. The Merlin programmer is a powerful, portable computer that is designed to help physicians efficiently conduct cardiac tests, analyze therapeutic and diagnostic data, and

program implanted devices for optimal patient care. Physicians can quickly make changes to device therapy through the system's LCD touch-screen to decrease follow-up time. The physician can manually conduct a follow-up using the Merlin programmer and send it to the Merlin.net system through an import feature on the web application. The web application sends the follow-up to the DDT server, where the reports are processed similarly to the Merlin@home follow-up, and the processed transmission is available to view on the web application. A typical Merlin@home follow-up takes approximately five minutes to transmit and process the data and view it on the Merlin.net website. The Merlin.net system will show the history of all In-Clinic and Merlin@home automated follow-up in the same location for the convenience of the physician.

The combination of the Merlin@home transmitter and Merlin.net system provide the benefits of a complete remote patient care and patient management system. The Merlin@home has removed any patient or physician manual interaction from the follow-up process. The Merlin.net system expands remote patient management features that allow the physician to set up patient follow-up schedules, set critical alerts, and select critical alert notification destination. This alleviates most of the manual interaction between the physician and the patient with the exception of required office visits.

### **Merlin.net™ Patient Data Processing**

The Device Data Translator (DDT) server creates the report package that is viewed on the Merlin.net web application and consists of two components: A Virtual Unity Programmer (VUP) software component and an External Parsing Service Manager (XPSM) component. The VUP component reads the patient record similar to the Merlin programmer. It interprets the incoming data from the Merlin@home and the In-Clinic

flow-ups and generates a variety of different PDF reports. These are the same PDF reports that are generated by the Merlin programmer and are available to view, print, and export. The VUP component is directly branched off the Merlin programmer code. The display rules are stripped from the programmer code in order to make the processing faster. The display rules are a bulky code that slows the processing speed of the follow-up and is not needed for the VUP since it is simply processing the data and exporting it. The DDT receives approximately 4000 follow-ups a day from the field, so the processing speed is very important in order to ensure that the information is processed and displayed on the web application in a timely manor. The VUP and programmer code branched from the same code base is desired by Merlin.net because consistency between the products is crucial for patient safety and customer perception. This ensures that all programmer bug fixes are incorporated into each VUP release and the PDF report from both components match. After the VUP component has interpreted the follow-up data and generated the required PDF reports the follow-up data is transferred to the XPSM component. The XPSM component generates the following three external reports:

- HL7/IEEE – This .xml file is a report that is used as a transport mechanism for delivering medical information between different organizations. This is used for EMR (Electronic Medical Records) transferred between different clinics. The parameters in this report are required for EMR compliance.
- GDF (GUI Data File) – This .xml file contains crucial device information that is inserted into the Merlin.net database. This data is displayed on the Merlin.net web application. Parameters contained in the GUI Data File include activity stored on the device from the patient follow-up session.

- VA Report (Follow-up Summary Data File) – A .txt file that has parameters and a format specific to the VA (Veterans Affairs) clinics requirements. The format and parameters contained in this report are specific requirements of the Veterans Affairs clinic.

After XPSM generates these external reports, the transaction is then re-zipped into a processed transmission that includes the VUP generated PDF reports, external reports, and original follow-up data. The transmission is then forwarded to an outgoing directory where it is populated in the Merlin.net database and then viewable on the web application. The Merlin.net system currently has no programming capabilities. It's solely used for processing and display patient follow-up information for the physician or clinic representative to view.

### **Merlin.net Verification**

The St. Jude Medical software verification department ensures that the product software meets the requirements documented in the software requirement specifications. Software verification is a planned and systematic pattern of all actions necessary to provide a high level of confidence that the item or product conforms to established technical requirements. It ensures that all software requirements are successfully tested and meets the specifications before it is released for clinical use. Requirements are analyzed for correctness, completeness, and testability. Any implied requirements discovered during the analysis are identified. Formal verification testing consists of developing and executing a set of verification test designs and procedures that provide full requirements coverage. The Merlin.net verification team has two primary

responsibilities consisting of testing the DDT functionality and verifying the information in the three XPSM generated external reports is correct.

DDT functional testing covers the basic, alternate, and exception flows of the DDT. The basic flow tests the follow-ups processing and priority handling. It ensures that the directory structure of the processed file is correct and all mandatory reports are present in the processed file. Priority handling testing ensures that follow-ups sent at high priority, such as a patient initiated follow-up, are processed with a higher priority than the scheduled follow-up with normal priority. The alternate and exception flows of DDT functional testing covers different error handling conditions and error logging of the DDT. For example, if a corrupt follow-up is sent to the DDT, it will fail the follow-up, store it in a failed directory, and send an email notification to the IT administrator indicating the reason for the failure. A log file is also generated with more detailed information of the specifics of the failure. The administrator then checks the log file to determine the reason the follow-up is failing and contacts the appropriate team to investigate the issue. The issue could be due to a processing error and the follow-up session needs to be resent through the DDT to reprocess, or it could be a software bug that the software development team needs to address.

The more challenging aspect of verification testing is to ensure that all the information in the processed report package is outputted correctly. This is specifically referring to the XPSM generated reports since the VUP generated PDF reports are verified in the Merlin programmer software verification. These reports include the above mentioned HL7/IEEE, GDF, and VA reports. This testing checks the format of the reports, ensures the parameter names and the parameter values in the reports are correct.

See Table 1 below for an example of some of the information and parameters that are generated in each of the XPSM reports. Note: This is a high level example of parameters included in each report, but does not list all supported parameters.

*Table 1: Example Parameters supported in XPSM generated reports*

	<b>HL7/IEEE</b>	<b>GDF</b>	<b>VA Report</b>
Example Parameters	Device Manufacturer	Device Model and Serial#	Episodal Information
	Implant Date	Clinic Preferences	Device Status
	Device Serial Number	Episodal information	Lead Information
	Lead Information	Alerts	Program parameters
	Battery Information	Device Status	VT/VT Episodes
	Episodal information	Lead Information	Episode Details
	Programmable parameters	Program parameters	

The XPSM generated reports have information that will be viewed by medical professionals, so it's crucial that this information is displayed accurately. There are currently 190 parameters in the HL7/IEEE report, 50 parameters in the GDF report, and 56 parameters in the VA report and support for new parameters can be added for each new Merlin.net release. Additionally each parameter can support several different values and some values are dependent on the values that other parameters are programmed to. Considering the large number of parameters to be tested and the variation of ranges and scenarios possible for the patient follow-up data it is necessary for the Merlin.net verification testing team to create as many different patients' In-Clinic and remote follow-up scenarios as possible to fully test that the DDT software implementation is outputting the XMPS reports correctly. The goal is to cover all ranges of parameter values for each parameter to ensure that they are displayed with the correct names and values. Some of the parameter values are dependent on each other, so it's necessary to create a large number of patient follow-up sessions to cover the full range of values. For

example, if a device is programmed to atrial sensing and pacing only, no parameter values are present for ventricular sensing and pacing. The Merlin.net project currently supports 52 device models and will continue to support new devices models for each release. The last Merlin.net project (Merlin.net 4.0), 250 patient follow-up sessions were used for testing. The Merlin.net releases are approximately 6 months apart, and we add approximately 20-30 new patient follow-up records each release for newly supported device models, so the number of sessions will continue to increase with each release. Given that the scope of this effort could be extremely time consuming and the team's resources are limited it is desired that an automation process be developed to test the content of these reports.

## **II. OBJECTIVE**

The objective of this project is to fully or partially automate the verification effort of verifying the content of the XPSM generated reports. The manual process requires an individual to load the patient session record onto the Archive Viewer of the Merlin programmer, with a device containing certain scenario criteria needed for full parameter coverage, navigate through the different programmer screens until the parameter is found, and then manually enter these parameters in an expected values spreadsheet. These values then have to be manually compared to the values in the XPSM reports generated by the DDT. The manual process is tedious, prone to human error, and requires a lot of time and rework. Any formatting errors, mistypes, or misreads will cause the comparison discrepancies and therefore have to be investigated. It is estimated that manually verifying one patient follow-up session takes approximately 5 hours for a well trained individual. This means that for each Merlin.net project, if 30 new patient follow-up sessions are added, it will take about 150 man hours (3.75 weeks) to complete the verification of these new XPSM generated reports. There are several software interim releases for each Merlin.net project, for feature and code enhancement, so there could easily be 5-10 interim software releases for each Merlin.net release. This means that the manual verification of 150 hours could have to be conducted several times within the timeframe of the project. This time estimate does not include the existing 250 patient follow-up records that must be re-compared for each release. Manually verifying the content of the XPSM generated reports, for all of these patient follow-up sessions, for each Merlin.net release would take a significant amount of time to verify, so it is critical that an automation process be developed. An automated process to verify these “external

reports” will allow the verification team to do a more thorough job of testing more patient follow-up scenarios, while still meeting the deadlines. The Merlin.net deadlines are very aggressive and automating this process is crucial to meeting these deadlines and producing a high quality system.

### **III. METHODS AND MATERIALS**

#### **TDE Implementation**

In order to extract the device parameters from the patient follow-up session, the TDE (Test Development Environment) tool is used. The TDE tool extracts the device parameters from the patient follow-up session on the archive viewer of the programmer. The archive viewer is simply a way for the test engineer to view a patient follow-up session on the Merlin programmer. Loading the patient follow-up session (.zip file) onto the programmer is a manual process that involves saving the follow-up session to a USB drive and opening it in the Merlin programmer archive mode. This operation will display all the follow-up session information on the programmer exactly how it displayed during real time device interrogation. The TDE tool is an Integrated Development Environment in the sense that you can write scripts, edit, debug, execute, schedule runs, and generate reports. TDE will navigate through the programmer screens and look for the object IDs of the parameters that are used in the external reports verification. The script is written to extract the desired values and automatically populates the device parameter in a Device parameters spreadsheet in a workbook named Master Data File (MDF). There are about fifteen parameters that are not implemented because they do not have object ID's on the programmer screen. These are mainly values from graphs and charts that still have to be manually entered by navigating through the different programmer screens until the parameters are found, and then manually enter these parameters in the device parameters Excel spreadsheet. The TDE parameter extraction process only needs to be executed one time, when there is a new patient follow-up session. The device parameters for each

patient follow-up session are saved to the worksheet and do not need to be extracted again for any reason. See Figure 7 for a diagram of the external report automation process. See Appendix A for an example of TDE script.

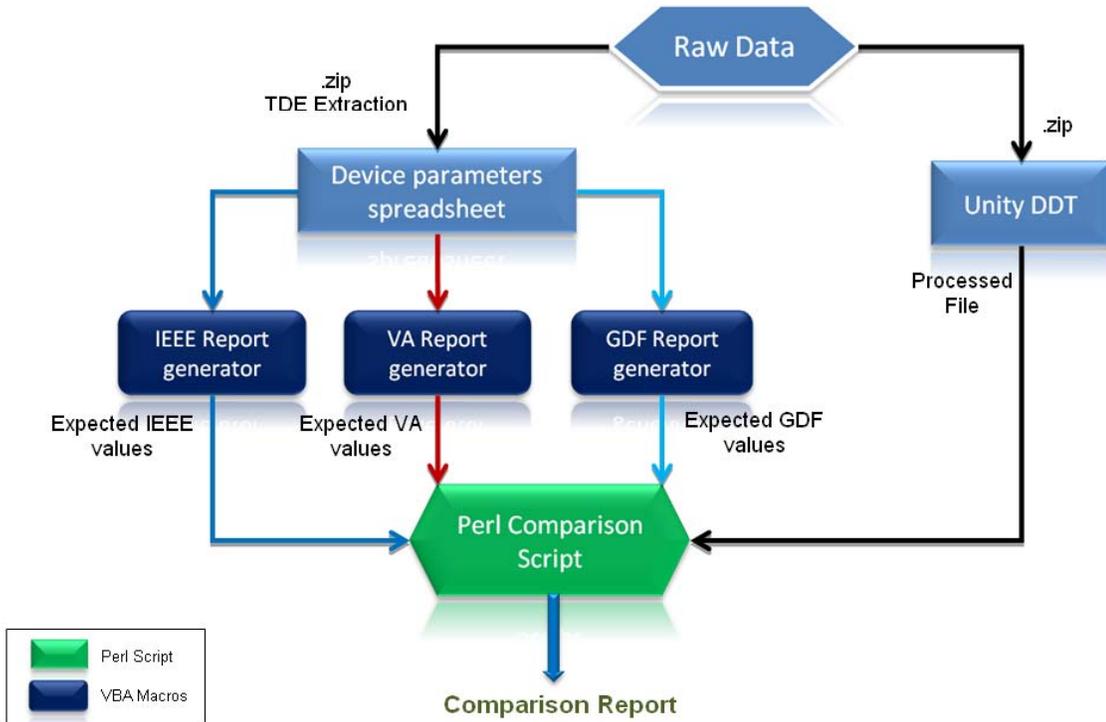


Figure 7: Diagram of the automated external report verification process

Step by step explanation of the block diagram above.

1. Raw Data (session record) is a set of patient follow-up sessions that are generated by the Merlin Programmer and Merlin@home follow-ups, and compressed in a suitable format (.zip or .tar.gz). This session record contains all the device information.
2. TDE will load this patient follow-up session on the Merlin programmer, navigate through various screens on the programmer, read the device

parameters and export the values in an excel spreadsheet. This forms the “Device parameters spreadsheet”.

3. Once all the device parameter values are extracted in the spreadsheet, they are passed through a set of mapping rules (Report generators), and these form the expected values. The mapping rules are implemented as VBA (Visual Basic for Application) macros, one for each kind of report viz. IEEE, VA and GDF. The expected values are written back to the spreadsheet to form an “expected values spreadsheet”.
4. The patient follow-up session (same from step 1) is sent to the DDT for processing.
5. The DDT generates the processed file that contains the three reports, IEEE, VA and GDF.
6. A perl script is then used to compare the parameter values in the DDT generated reports with the corresponding expected values from the expected values spreadsheet (generated in step 3). The perl script generates a comparison report displaying the discrepancies between the actual and the expected parameter values, if any.

### **Expected values spreadsheet**

The expected values spreadsheet is comprised of several worksheets that are listed below. This spreadsheet is used to compare the expected values for the parameter values for each external report with the XPSM generated external reports processed by the DDT. See Figure 8 for an example of the expected values spreadsheet. The first column, column A, is a list of all the device parameters. Each column thereafter is the patient

follow-up sessions and the corresponding values for each parameter. If a cell is blank it means that there is no value for that particular parameter for the patient follow-up session. This could mean that this feature is not supported for the device model or the particular activity to create this parameter did not occur on the device for the patient follow-up session. This will not cause a failure since the expected values and the processed reports will both have no values for these parameters.

- **MDF** – This worksheet contains the device parameter names and their values. Most of these values are extracted using TDE and a few have to be entered manually. In this sheet, one column represents one patient session followed by the values for each parameter. When parameters have multiple values they are written in the same cell separated by a delimiter (such as '|'). This worksheet forms the input to VBA macros.
- **VA Report** – This worksheet is where the expected result for testing the VA report parameters resides. Initially it is blank, with only the VA parameter names in the first column, and then gets populated with the expected values when the Visual Basic for Applications (VBA) macros are executed.
- **GDF Report** – This worksheet contains the expected result for testing the GUI Data File (GDF) generated by the DDT. Most of the values in this sheet are manually entered since there are no mapping rules for generating the GDF. There is, however, some VBA implementation to read the parameter values from the UnityMDF sheet, format them as per requirements, and then write them back to this worksheet.

- **HL7/IEEE Report** – There are seven remaining worksheets where the expected result for testing the HL7/IEEE report parameters resides. These HL7/IEEE parameters are divided among the sheets based on the parameter groups that are output in the HL7/IEEE xml report. The names of these sheets are Parameter Groups, Device Observations ICD, Patient Observations, Device Therapy Settings, LV-Sensing-Pacing Lead, RA-Sensing-Pacing Lead and RV-Sensing-Pacing Lead. Similar to the VA Report sheet, these worksheets are also initially blank until the VBA macros are executed to populate the expected values.

	A	B	C	D	E	F	G
1		2566_3213-36_2007-11-06_19-38-43.zip	2566_3213-36_2007-11-06_15-03-04.zip	10011_3211-36_2008-09-27_15-11-39.zip	6749_3211-36_2008-11-19_16-33-44.zip		
2	UP_VT1Therapy1RampStep		15 msec				
3	UP_VT1Therapy2RampStep		20 msec				
4	UP_VT1Therapy1ScanMode		On (Dec)		On (Dec)		
5	UP_VT1Therapy2ScanMode		On (Dec)				
6	UP_VT1Therapy1NoBursts		5		5		
7	UP_VT1Therapy2NoBursts		6				
8	UP_AtrialSensitivity						
9	UP_AutoModeSwitch	DDIR	DDI		Off		
10	UP_VT2Therapy1RampStep		20 msec				
11	UP_VT2Therapy2RampStep		30 msec				
12	UP_VT2Therapy1ScanMode		On (Dec)				
13	UP_VT2Therapy2ScanMode		On (Dec)				
14	UP_VT2Therapy1NoBursts		6				
15	UP_VT2Therapy2NoBursts		7				
16	UP_AtrialPulseAmplitude	2.5 volts	3.0 volts		2.5 volts		
17	UP_AtrialPulseWidth	0.5 msec	0.3 msec		0.5 msec		
18	UP_BaseRate	60 bpm	70 bpm	60 bpm	60 bpm		
19	UP_LVPulseAmplitude	3.5 volts	3.0 volts		5.0 volts		
20	UP_LVPulseWidth	0.5 msec	0.3 msec		0.5 msec		
21	UP_MaximumTrackingRate	95 bpm	105 bpm		110 bpm		
22	UP_Mode	DDD	DDD	VVT	DDD		
23	UP_PacedAVDelay	200 msec	180 msec		200 msec		
24	UP_SensedAVDelay	150 msec	150 msec		150 msec		
25	UP_RVPulseAmplitude	2.5 volts	2.75 volts	2.0 volts	5.0 volts		
26	UP_RVPulseWidth	0.5 msec	0.6 msec	0.5 msec	0.5 msec		
27	UP_EOSVoltage	2.1 volts	2.1 volts	2.1 volts	2.10 volts		
28	UP_ERIVoltage	2.45 volts	2.45 volts	2.45 volts	2.45 volts		
29	UP_VFDetectionInterval	400 msec	340 msec	250 msec	330 msec		
30	UP_VFEnergy1	36.0 joules	25.0 joules	32.0 joules	36.0 joules		
31	UP_VFEnergy2	36.0 joules	27.5 joules	32.0 joules	36.0 joules		
32	UP_VFEnergy3	36.0 joules	36.0 joules	32.0 joules	36.0 joules		
33	UP_InterventricularPaceDelay						
34	UP_LVPulseConfiguration	LV tip to RV coil	Bipolar		LV tip to RV coil		
35	UP_MaximumSensorRate	95 bpm	105 bpm	110 bpm	110 bpm		
36	UP_ShortestAVDelay	100 msec	90 msec		100 msec		
37	UP_AtrialTachycardiaDetectionRate	180 bpm	190 bpm	130 bpm	180 bpm		
38	UP_NegativeAVHysteresisSearch	Off	-100 msec		Off		
39	UP_VentricularPacemakerRefractory	250 msec	200 msec	250 msec	250 msec		
40	UP_ShortestPVARPREF	225 msec	250 msec	225 msec	225 msec		
41	UP_VT1Energy1	36.0 joules		10.0 joules			
42	UP_VT1Therapy1Type	CVRT	ATP		ATP		
43	UP_VT1Energy2	36.0 joules		22.5 joules	25.0 joules		
44	UP_VT1Therapy2Type	CVRT	ATP		CVRT		
45	UP_VT1Energy3	36.0 joules		22.5 joules	36.0 joules		
46	UP_VT1Therapy3Type	CVRT	CVRT		CVRT		
47	UP_VT1Energy4	36.0 joules		27.5 joules	36.0 joules		
48	UP_VT1Therapy4Type	CVRT	CVRT x 2		CVRT x 2		
49	UP_VT1DetectionInterval	590 msec	425 msec	405 msec	430 msec		
50	UP_VT2Energy1	36.0 joules		15.0 joules	15.0 joules		
51	UP_VT2Therapy1Type	CVRT	ATP		CVRT		
52	UP_VT2Energy2	36.0 joules		20.0 joules	25.0 joules		
53	UP_VT2Therapy2Type	CVRT	ATP		CVRT		
54	UP_VT2Energy3	36.0 joules		34.0 joules	36.0 joules		
55	UP_VT2Therapy3Type	CVRT	CVRT		CVRT		
56	UP_VT2Energy4	36.0 joules		25.0 joules	25.0 joules		
57	UP_VT2Therapy4Type	CVRT	CVRT		CVRT		

Figure 8: Example of expected values spreadsheet of external files

## **VBA Implementation**

Excel VBA is used to implement the mapping rules for the VA and HL7/IEEE report expected values generation. These mapping rules convert the device parameters to the expected parameters in the VA and HL7/IEEE reports. The following are the steps implemented in VBA to generate the expected values:

- i. The columns in the UnityMDF sheet are read one at a time. This corresponds to reading one patient follow-up session.
- ii. The parameter values are stored in variables.
- iii. The script then reads the VA & HL7/IEEE parameters from the respective sheets, one at a time.
- iv. The control is then passed on to a series of switch-cases. These cases contain the implementation of the mapping rules that is used to generate the expected values. Based on the parameter name read in step (iii) above, one of the cases is satisfied and the expected value is calculated.
- v. This value is then written back to the VA & HL7/IEEE worksheets.
- vi. The above steps are executed in a loop until all the columns are processed. After the macro is finished executing, the expected value spreadsheet is completely generated.

There are two macros that need to be executed, one for VA and one for HL7/IEEE, in order to generate the expected values. These can be run directly from the Excel spreadsheet by going to Tools → Macro → Macros, select the macro, and click Run. Once the macro is finished executing, a pop-up appears to indicate the completion

of the process. The work must then be saved and the expected result spreadsheet is generated. See Appendix B for an example of VBA macros.

### Perl comparison script

A Perl script was written that is used to perform the comparison of the reports processed by the DDT to the values in the expected values spreadsheet. The patient follow-up session is passed through the DDT which produces the processed files. These processed files contain a GUI Data File (GDF) report, VA report, and HL7/IEEE report. The additional information included in the processed file are the VUP generated PDF reports and the original raw data that was sent to the DDT. See Figure 9 for an example of DDT processed file contents.

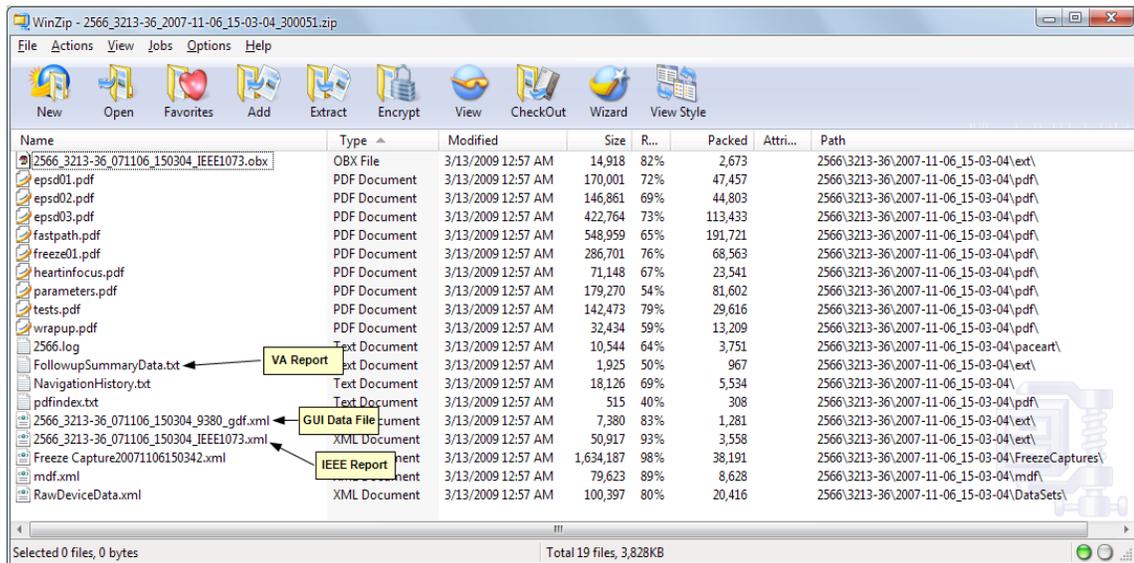


Figure 9: DDT processed file contents

The implementation consists of writing individual Perl modules for VA (VA.pm), HL7/IEEE (IEEE.pm & IEEE3dot0.pm) and GDF (GUI.pm) reports comparison logic. These modules have the logic to parse the respective test reports, arrange the data in a

hash structure, perform the comparison with the expected values and create the comparison report. The functions that are common to all the modules are written in another common module shared by the individual report modules (pcsATS.pm). The common functions include reading the expected result spreadsheet and arranging it in a hash structure, extracting the zip processed files, reading configuration values, and some other functionalities. The following is the complete list of files. See Appendix C for an example of the perl comparison script.

- i. pcs-ats.pl – This is the main Perl script that will need to be run on the command prompt.
- ii. pcsATS.pm – Perl module containing the common sub-routines that are shared by other modules.
- iii. VA.pm – Perl module containing the logic to parse and compare VA report.
- iv. GUI.pm – Perl module containing the logic to parse and compare GUI report.
- v. IEEE.pm & IEEE3dot0.pm – Perl module containing the logic to parse and compare IEEE reports. There are two versions of HL7/IEEE report that are generated by the DDT, v2.3 (with file extension “.obx”) and v3.0 (with file extension “.xml”).
- vi. pcsATS.config – Configuration file containing script settings.

The comparison can be performed by following the steps below:

- i. Create a folder (generally C:\pcsATS\) and copy the scripts and expected values spreadsheet into this folder.
- ii. Copy the DDT processed files in path - C:\pcsATS\srcDir\ProcessedFiles\
- iii. Open up a command prompt.

- iv. Navigate to the folder containing the script (generally C:\pcsATS\)
- v. Type in the command `perl pcs-ats.pl [IEEE/VA/OBX/GUI]`. See Figure 10 for an example of the perl script execution for VA report.
- vi. The script will be executed and the comparison report will be generated once completed.

```

C:\WINDOWS\system32\cmd.exe
C:\>cd pcsATS
C:\pcsATS>perl pcs-ats.pl UA
Set up gcc environment - 3.4.5 <mingw-vista special r3>

----- UA -----
SpreadSheet is loaded!
Extracting: '2566_3213-36_2007-11-06_15-03-04.zip' ...
extractTGZFile: File extract directory: tmpDir/2566/3213-36/2007-11-06_15-03-04/
ext
extractTGZFile: 2566_3213-36_2007-11-06_15-03-04_300051.zip contains the file Fo
llowupSummaryData.txt.
Extracting: '2566_3213-36_2007-11-06_19-38-43.zip' ...
extractTGZFile: File extract directory: tmpDir/2566/3213-36/2007-11-06_19-38-43/
ext
extractTGZFile: 2566_3213-36_2007-11-06_19-38-43_300063.zip contains the file Fo
llowupSummaryData.txt.

Loading UA file: 2566_3213-36_2007-11-06_15-03-04.zip
...[OK]
Cross-checking: 2566_3213-36_2007-11-06_15-03-04.zip ...

Loading UA file: 2566_3213-36_2007-11-06_19-38-43.zip
...[OK]
Cross-checking: 2566_3213-36_2007-11-06_19-38-43.zip ...

----- Done for UA -----
C:\pcsATS>

```

Figure 10: Perl script execution for VA report

The comparison report, generated by the Perl comparison script, is a report that lists the comparison result of each patient follow-up session that was used for testing. This report shows the comparison of the external reports processed by the DDT to the parameter values in the expected values spreadsheet. In the case of no comparison failures, a string, 'Passed with No Errors', is displayed below to the patient follow-up session name. In the case of a failed comparison each patient follow-up session will have a list of failing parameters under the patient follow-up name, and a string, 'Failed', is displayed beside the patient follow-up name. Each parameter, and its corresponding

DDT processed and expected values, are listed under the session name. This is an easy way to display and identify what parameters are showing a discrepancy when comparing the DDT processed values with the expected values. The comparison report can help identify issues in the DDT code between releases. Code changes are due to requirement changes or software enhancements that may affect other parameter values. Processing the patient follow-up sessions for each DDT release and executing the perl comparison script is a good way to check that the implementation didn't affect the values of other parameters. The comparison report basically states what parameters pass or fail for each patient follow-up session. See Figure 11 for an example of comparison report for VA report.

```

VATestResult.txt - Notepad
File Edit Format View Help
===== Completed =====

2566_3213-36_2007-11-06_15-03-04.zip: Failed

- VT/VF Episodes.Groupheader:
  DataFile: '3 Nov 06, 2007 15:00 NSVF 176bpm No Rx N/A 10sec Yes|2 Nov 06, 2007 14:58 SVT
146bpm No Rx N/A 106sec Yes|1 Nov 06, 2007 14:57 VF 181bpm 27.5j Yes 42sec Yes'
  Expected: '3 Nov 06, 2007 15:00 NSVF 176bpm No Rx N/A 10sec Yes|2 Nov 06, 2007 14:58 SVT
146bpm No Rx N/A 106sec Yes|1 Nov 06, 2007 14:57 VF 181bpm N/A Yes 42sec Yes'

===== Completed =====

2566_3213-36_2007-11-06_19-38-43.zip: Failed

- Episode Details.Groupheader:
  DataFile: '4 36.0j 36.0j, At least one shock unsuccessful, Three or more VT/VF episodes in 24
hours|3 No Rx, Non-sustained episode, Three or more VT/VF episodes in 24 hours|2 36.0j 36.0j, At least one
shock unsuccessful, Three or more VT/VF episodes in 24 hours, VT therapy timeout, Duration over 1 minute,
SVT discriminators disagree, Rate Branch:Sinus Tach (V = A) Rate Branch, Sudden Onset|1 36.0j, At least one
shock unsuccessful, Three or more VT/VF episodes in 24 hours, SVT discriminators disagree, Rate
Branch:Sinus Tach (V = A) Rate Branch, Sudden Onset'
  Expected: '4 36.0j 36.0j, At least one shock unsuccessful, Three or more VT/VF episodes in 24
hours|3 No Rx, Non-sustained episode, Three or more VT/VF episodes in 24 hours|2 36.0j 36.0j, At least one
shock unsuccessful, Duration over 1 minute, SVT discriminators disagree, VT therapy timeout, Three or more
VT/VF episodes in 24 hours, Rate Branch:Sinus Tach (V = A) Rate Branch, Sudden Onset|1 36.0j, At least one
shock unsuccessful, SVT discriminators disagree, Three or more VT/VF episodes in 24 hours, Rate
Branch:Sinus Tach (V = A) Rate Branch, Sudden Onset'

```

Figure 11: Example of comparison report for VA report

## IV. RESULTS

The results for the Merlin.net Automation of externals reports process reduced the external reports verification of new and existing patient follow-up sessions significantly. The average execution time for each script component can be seen in Table 2 below. The total script execution time for one patient follow-up session is approximately 25.5 minutes. This is an average execution time taken from 10 patient follow-up sessions. The execution times are impacted depending on the number of parameters that are extracted with the TDE tool. If the device has a lower feature base or does not have a lot of activity such as episodes and alerts, the TDE script will extract a lower number of parameters causing it to execute faster.

*Table 2: Average execution time for each implemented script.*

<b>Script</b>	<b>Time (minutes)</b>
TDE	24
VBA Macros	.5
Perl Comparison	1
<b>Script Execution Time</b>	<b>25.5</b>
<b>Manual Extraction Time</b>	<b>15</b>
<b>TOTAL TIME</b>	<b>40.5</b>

TDE will only need to be executed when there is a new patient follow-up session. After the device parameters are saved to the expected values spreadsheet it will not need to be executed again. The manual extraction of the 15 parameters that are not extracted from the TDE script takes approximately 15 additional minutes per patient follow-up session. This makes the total time to extract and compare the results for new patient follow-up sessions around 40.5 minutes, compared to 5 hours with the manual process. The manual extraction and comparison time was averaged from 10 patient follow-up

sessions. After this stage the VBA macros and Perl comparison script take less than 2 minutes to execute. The results of this automation process benefits the Merlin.net verification team in two ways.

- Saves time to extract new patient follow-up session expected values in comparison with DDT processed values.
- Reduces the expected values generation and comparison of existing patient follow-up sessions to approximately 1.5 minutes.

The time savings for the newly added patient follow-up sessions enabled the verification team to meet the Merlin.net 4.0 deadline successfully. These deadlines are important to meet because they are closely staggered behind the new device releases. Physicians will not purchase the new devices if there are no remote patient management capabilities, since it's a major inconvenience not to have it. It is crucial to St. Jude Medical's device sales to have immediate remote patient management support of all new devices. Merlin.net is a free service and the division's revenue is heavily based on device sales, so immediate support is essential. Thirty seven new patient follow-up sessions were added to the Merlin.net 4.0 from the previous project. These sessions were created to cover the new parameters and range of parameter values introduced for this project. The new sessions also covers all new device models. The new device models, parameters, and parameter values must be completely tested to ensure full requirements coverage in the testing. This saved approximately 160 man-hours of testing. See Figure 12 for a visual representation of the time comparison of the manual process and the newly developed automated process.

### Time Comparison of Manual vs Automated

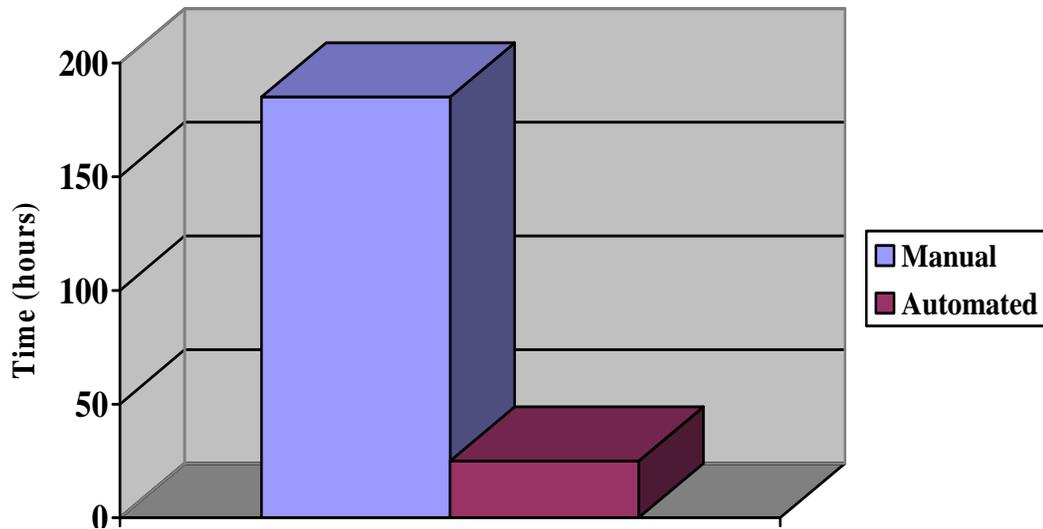


Figure 12: Time Comparison of manual and automated process.

The VBA macros will require some maintenance if any of the mapping requirements change. This will simply require the VBA code change to the parameter that changed in the requirements, and the macro then must be saved and re-executed. After the Perl comparison script is executed, generated results file must be manually analyzed for discrepancies. Any issues encountered are analyzed to determine if there is an issue in the DDT software or an issue in the verification scripts.

The Formal Testing process also benefits with this automated process. Formal Testing consists of executing a set of verification test procedures that provide full requirements coverage. All patient follow-up sessions will be added during the dry run process of the Merlin.net project, so Formal Testing will only include existing patient follow-up sessions. During Formal Testing, the only script that needs to be executed is the Perl comparison script with each DDT release to ensure that the DDT processed values are matching the expected values, so the total execution time only takes about 5

minutes for all patient follow-up sessions. The results files are then manually analyzed for each release until there are no discrepancies. All processed files, scripts, and results files are stored and maintained in the IBM ClearCase Software versioning control system.

## **V. CONCLUSION AND FUTURE WORK**

The Merlin.net Automation of external reports process project met the needs of the Merlin.net verification team and enabled the team to meet their previous deadline. The automation process saved 160 hours on the new patient follow-up sessions alone, and allowed the team to be successful even though lacking adequate resources. The Merlin.net project continues to grow in scope for each new release due to new system features and a desire to expand our testing coverage. Since Merlin.net is a free service, project management has not approved requests for additional resources. This stresses the importance to come up with innovative ways to automate tests in order to accommodate more work with the same amount of resources.

The verification team is still in the process of examining the use of other tools to complete the device parameter extraction of the fifteen remaining parameters that TDE could not extract. The DIA (Device Image Analyzer) tool is one of the tools being considered. The tool is able to read a device image and display the parameter values in table format however, this would still require a person to manually navigate through various screens outputted by the tool to retrieve the desired parameter. Further investigation needs to be conducted to analyze whether the DIA tool can be included in the automated process and if the required parameters exist in the output of the tool. Training will have to be conducted on the tool at its next availability, which will start the analysis process. This training will cover the features of the DIA tool that will be used by the Merlin.net team to see if it's a viable option.

The team is also working on a way to integrate all the different script executions in the process in order to completely automate the process. SAINTS (Software

Automated Integrated Network Test System), is an internally developed test management system that is being considered for this activity. In the SAINTS system, the Scheduler controls the execution of automated tests. The purpose of the Scheduler is to execute tasks, and automated tests are simply a specialized version of a task.

When executing an automated test, the Scheduler executes the task noting the task type, which must be previously defined. There are three basic task types: Test, Clean Up, and Maintenance. Within the Test type, a sub type must be defined. Some examples of typical test sub-types are:

- Batch File
- QTP test
- JRules
- JUnit
- UPTE
- Compiled executable

Sub-types can be individually defined for very specific test types, or existing types may be "cloned" and reused. Test task types can also specify other tasks such as specific Clean-up Tasks to be run after the test has completed.

When a test type is defined, the execution method is specified and the allowed choices are NONE, EXE, SCRIPT, and COMMAND. These basic types define the overall execution method and allows the additional parameters as needed to describe the execution method, such as script name, or execution engine (i.e. wscript.exe or another, perhaps custom, launch executable).

Execution of the test involves resources which also must be predefined within the greater SAINTS system. To execute a test, the task is registered with the scheduler, and at the specified execution time, if the requested resource is available, SAINTS copies all required components of the test (various include files, .xml, the actual executable or the executable script code, etc.) to the resource machine, and via a resident execution manager service, calls either the compiled executable or the command line sequence for the script engine with the associated script component file.

Assuming the compiled test (or the executable script) has also included the SAINTS callback API, the resident execution manager service will communicate test status (pass, fail, executing, and error) back to the SAINTS server. That status is reported in near real time for the user to review based upon the user's request for that information via a webpage interface.

When test execution is complete, the execution manager reports that status. That report triggers any associated clean-up tasks which typically collect all test artifacts, logs, etc., encapsulates them in a ZIP archive, and sends them to the SAINTS server for import into its secure database. The SAINTS system is newly available to the Merlin.net Verification team and therefore is being investigated for the feasibility to include and complete our automation process.

## REFERENCES

- 1) "Pacemakers". 2002. MedicineNet.com. 24 April. 2002 <  
<http://www.medicinenet.com/script/main/art.asp?articlekey=1947>>.
- 2) Hratch L. Karamanoukian, MD, "Natural Pacemaker of the Heart." 2008. Vein Treatment News. 14 December. 2008 <  
[http://www.veinsveinsveins.com/vein\\_treatment\\_news/65/Natural\\_Pacemaker\\_of\\_the\\_Heart.html](http://www.veinsveinsveins.com/vein_treatment_news/65/Natural_Pacemaker_of_the_Heart.html)>.
- 3) Kingwell BA, McPherson GA, Korner PI. Assessment of gain of tachycardia and bradycardia responses of cardiac baroreflex. *Am J Physiol*. 1991;260:H1254-H1263.
- 4) Valentin Fuster, John Willis Hurst, Robert A. O'Rourke. Hurst's the heart, Volume 1. McGraw-Hill Professional, 2004.
- 5) M.Glikson, P.Friedman. The implantable cardioverter defibrillator. *The Lancet*. 2001;357:H1107-H1117.
- 6) Housecall Plus™ Remote Follow-up System Receiver Model 3180-R Reference Manual. St. Jude Medical Cardiac Rhythm Management Division, 2008.
- 7) Merlin@home™ Transmitter Model EX1100 User's Manual. St. Jude Medical Cardiac Rhythm Management Division, 2009.
- 8) Merlin@home™ Transmitter Quick Start Guide. St. Jude Medical Cardiac Rhythm Management Division, 2009.
- 9) Merlin.net™ Patient Care Network Help Manual. St. Jude Medical Cardiac Rhythm Management Division, 2009.

- 10) Steven Holzner. Perl Core Language, Second Edition. Arizona: Paraglyph Press, Inc., 2004.
- 11) John Walkenbach. Excel 2003 Power Programming with VBA. Indiana: For Dummies, 2004.
- 12) Astrid Fiano, “St. Jude Medical’s Remote Monitor for Cardiac Devices FDA Approved.” 2008. DOTmed News. 30 July. 2008 <  
<http://www.dotmed.com/news/story/6604/>>.
- 13) Merlin.net External Parsing Service Manager (XPSM) Software Sub-system Design Specification. St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40005522), 2009.
- 14) Follow-up DDT Software Requirements, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40004327), 2009.
- 15) Matrix-PCS Follow-up Summary Data Parameter Mapping Specifications, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40004733), 2009.
- 16) Matrix-PCS VUP Interface Specification, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40004660), 2009.
- 17) IEEE 1073 Parameter Mapping – HL7 V3.0, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40003460), 2009.
- 18) Matrix PCN GUI Data File Mapping Specification, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 40005877), 2009.

- 19) Unity Export to PC Database Interface Specifications, St. Jude Medical Cardiac Rhythm Management Division (Team Center Document Number 60004665), 2009.
- 20) SAINTS Test Data Manager Component Formalized Requirements, St. Jude Medical Cardiac Rhythm Management Division, 2005.
- 21) Device Image Interpreter Software Requirements, St. Jude Medical Cardiac Rhythm Management Division, 2005.
- 22) Ron Patton. Software Testing, Second Edition. Indiana: Sams Publishing, 2006.
- 23) Glenford J. Myers, Tom Badgett, Todd M. Thomas, and Corey Sandler, The Art of Software Testing, Second Edition, New Jersey: John Wiley & Sons, Inc., 2004.

## **APPENDIX A: TDE SCRIPT**

This section was intentionally left blank for proprietary reasons.

## **APPENDIX B: VBA MACROS**

This section was intentionally left blank for proprietary reasons.

## **APPENDIX C: PERL COMPARISON SCRIPT**

This section was intentionally left blank for proprietary reasons.