Requirements-centric closed-loop validation of implantable cardiac devices

Weiwei Ai, Nitish Patel and Partha Roop
Department of Electrical and Computer Engineering
University of Auckland, New Zealand
Email: wai484@aucklanduni.ac.nz, nd.patel, p.roop@auckland.ac.nz

***Abstract*—Implantable medical devices are recommended by physicians to sustain life while improving the overall quality of life of the patients. In spite of the rigorous testing, there have been numerous failures and associated recalls which suggest that completeness of the testing is elusive. We propose a new validation framework based on formal methods for real-time closed-loop validation of medical devices. The proposed approach includes a synchronous observer acting both as an automated oracle and also as a requirements coverage monitor. The observer combines an on-line testing adequacy evaluation module together with a heuristic learning module. This methodology was applied to validate a pacemaker over a virtual heart model. A subset of the requirements was used to test its efficacy. The results show that the proposed methodology can, in real-time, evaluate the test adequacy and hence guide the on-line test case generation to maximize the requirements coverage.**

I. Introduction

Medical devices have become increasingly complex both in the functionalities and in the ways they interact with the environment. Many devices, such as implantable cardiac devices, are designed to monitor the physiological conditions as well as autonomously deliver proper treatments. In 2009, 1,002,664 pacemakers and 328,027 implantable cardioverter defibrillators (ICDs) were implanted worldwide [1], while a large number of safety recalls have taken place, e.g., over 600,000 devices were affected between 1990 and 2000 in the USA, of which 200,000 or 41 percent were attributed to firmware issues [2]. Meanwhile, pacing problems have been reported by cardiologists due to incorrect mode or parameters selection [3], unintended mode switching as a consequence of intrinsic rhythm disturbances triggering algorithms inappropriately [4], design issues [5], etc. Most problems emerge after multiple interactions between the heart and the device. These are impossible to discover without a closed-loop context.

The efforts of Jiang et al. [6] and Chen et al. [7] illustrate that involving a heart model in the closed-loop is promising with respect to validating cardiac devices. However, none of the approaches have applied the rigorous requirements coverage criteria, such as UFC [8], PICC [9], etc. Also, there is no known approach for closed-loop validation of cardiac devices that uses run-time observers for improving the coverage.

Requirements coverage is fundamental adequacy metrics to determine whether a test suite covers the user requirements. It becomes more challenging in the closed-loop validation of cardiac devices due to decreased controllability of inputs. Once the configuration of the heart model has been completed, the inputs/outputs sequence will be totally determined by the closed-loop dynamics. Additionally, test case generation and execution are usually dovetailed together. In this context, run-time test oracle and adequacy evaluation are very important.

We develop the requirement-centric approach for closed-loop validation of cardiac devices. It can adapt to more rigorous coverage criteria to discover errors in the early stage of development. Given the number of devices recalls, we believe that it can facilitate the certification process.

Our contributions are the following: (1) We propose a closed-loop validation framework which includes synchronous observers simultaneously acting as an automated oracle and also a requirements coverage monitor. (2) We have created an enhanced testing adequacy evaluation module by incorporating a heuristic learning module with the potential to guide on-line test generation to achieve a given requirements coverage.

The organization of this paper is as follows. In Section II, we demonstrate the validation challenge and related work. The closed-loop validation framework is described in Section III. We present the experimental results and discussion in Section IV followed by conclusions and future work in Section V.

II. Challenge and related work

Many pacing problems over individual patients have been reported. For instance, in case [5], a pacemaker induced tachycardia manifests itself after more than 28 cycles of interaction between the device and the heart, which can not be captured by open-loop testing. As implantable cardiac devices continuously react to the environment (human heart), the validation should provide an input sequence that takes into account the device's outputs history, i.e. the previous outputs of the device have an impact on the future input sequence.

The necessity of closed-loop testing has drawn academic attention. Jiang et al. build a heart model to test the device under several conditions, such as atrial flutter [6] and sinus bradycardia with AV block [10]. Chen et al. [7] develop a hybrid heart model in Simulink, which incorporates stochastic mode switching. It is able to validate more complex pacemaker functionalities, like rate-adaptive pacing [11].

Previous work is based on ad-hoc test generation in which neither automated oracles nor systematic testing adequacy has been considered. In reactive system testing, manual oracles are

978-3-9815370-6-2/DATE16/©2016 EDAA

more time-consuming. Since a test case is a sequence of inputs and outputs, one needs to examine not only the combination of inputs and outputs in each step but also the order of the combinations. Therefore, automated oracles are essential.

Conventionally, *structure coverage* is used to measure test adequacy. Test cases are generated with the goal of covering all required structures of design. However, structure coverage guided testing, also referred to as structural testing, has no way of finding incomplete requirements [12]. We are more interested in *requirements-based testing* to validate the design meets user requirements without access to the internal structure of the system under test (SUT).

Structural coverage criteria over requirements, formalized as linear temporal logic (LTL) properties, are defined in [8] and Charles et al. [13] give more formal definition based the previous work. Gordon et al. [9] propose complementary criteria. With formalized requirements, one can generate off-line test cases using model checker. Nevertheless, in the case of closed-loop testing cardiac devices with a complex heart model, it is more desirable that the requirements model can simultaneously run with the system, which is able to dynamically determine how well the validation proceeds and achieve on-line testing.

III. Closed-loop validation framework

The proposed framework consists of four parts:

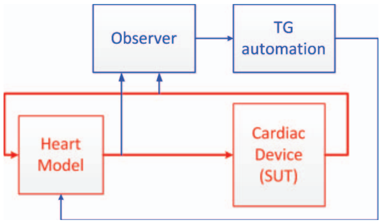


Fig. 1. The Closed-loop Validation Platform.

The heart model is programmed to exhibit the clinical requirements, i.e. a wide range of cardiac rhythms. The cardiac device is the SUT, which is a *black-box* because typically we cannot access its internal structure. The solid red line shows that the device is in a closed feedback loop. If the device is performing correctly, the heart will be guided from its arrhythmia into a clinically correct rhythm.

The synchronous observer monitors the run-time behaviour of the closed-loop system and serves as a test oracle. Moreover, the observer also monitors the requirements coverage. The outputs of the observers, carrying the coverage information, feed the test generation (TG) automation module. This module dynamically steers the operation of the heart by adjusting the model parameters. Hence the cardiac device will be thoroughly exercised.

A. Heart and Cardiac Device model

The heart model includes a network of cardiac nodes as in [7]. The cellular action potential model comes from the work [14] and enhanced with a conduction path model. To produce various heart rhythms, we have added more autorhythmic cells along the conduction system. Furthermore, stochastic rhythm controllers have been created to modulate the rates of autorhythmic cells to capture the heart rate variability.

The heart model can be dynamically reconfigured at run-time. The reconfigurable components primarily include the rhythm controllers and the conduction timing which are capable of covering a wide range of heart rhythms, such as Sinus Bradycardia and Tachycardia, Sinus Arrest, Ventricular Tachycardia, Premature Ventricular Complex and AV Block at different degrees.

The pacemaker model, in DDD mode [15], developed by Pajic et al. [16] is used to validate our framework. The device can sense the electrical excitation from the dual chambers (the right atrium and ventricle), denoted as atrial sense (AS) and ventricular sense (VS), respectively. With an absence of intrinsic activation, the device can deliver pacing pulses atrial pacing (AP) and ventricular pacing (VP). The device can also maintain Atrial-Ventricular (A-V) synchrony. We use atrioventricular interval (AVI) to represent the time difference between the atrial and ventricular activation. Fig. 2 illustrates four cardiac cycles. The first one is a normal intrinsic activation. In the second one, the device delivered a VP to synchronize with the AS because within AVI no VS occurred.

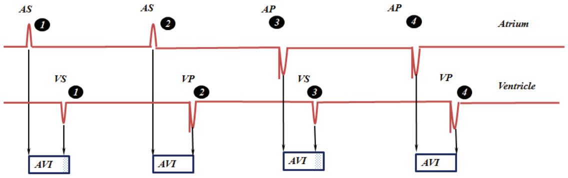


Fig. 2. Cardiac cycles.

B. Synchronous observers

As the ultimate objective is to determine whether the device meets the user requirements, formulating the requirements in [16] and testing obligations is of primary interest. We manually construct synchronous observers as shown in Fig. 3 to explicitly express the testing obligations corresponding to the Unique First Cause Coverage (UFC) proposed in [8].

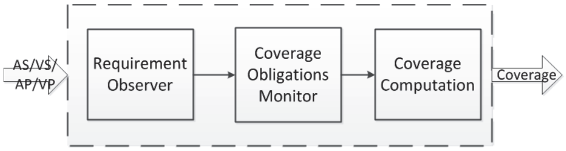


Fig. 3. The architecture of synchronous observers.

For instance, we intend to test the requirement “VP cannot occur during the interval $t_a \in (0, AVI)$”. The requirements observer monitors the behaviour of the closed-loop system including AS, AP, VS and VP, and the timing of events. As shown in Table I, the outputs indicate which combinations of individual requirement conditions are met, e.g., if VP occurs and $t_a >= AVI$, then the output is “P32”.

Furthermore, automated test oracles are implemented in the observers. In Table I, combination 2 has a false outcome, i.e. the requirement is violated and the test fails. In addition, we have incorporated clinical requirements, i.e. a wide range of arrhythmias that should be treated by the devices, which means we will make the testing prove that the design requirements are met under a variety of clinical conditions.

TABLE I
INDIVIDUAL REQUIREMENT CONDITIONS

No.	VP	$t_a >= AVI$	$VP\ imply\ t_a >= AVI$	Outputs
1	True	True	True	P32
2	True	False	False	P34, error
3	False	True	True	P33
4	False	False	True	P31

The outputs of requirements observers feed the coverage obligations monitor module, in which all the testing obligations imposed by the coverage criteria are modelled by state machines. An example of an obligation is: $P33\ or\ P31$ is followed by $P32$, which can be met by the input/output sequence of the first two cardiac cycles in Fig. 2.

The coverage is computed by using

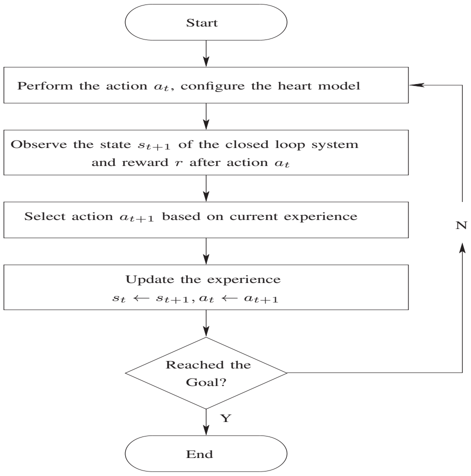
$$coverage = \frac{N_{met\ obligations}}{N_{total\ number\ of\ obligations}} \times 100\%$$

The coverage depends on the capability of controlling the heart model to produce required sequences. This computation result will then drive the test generation (TG) automation module.

C. Automated test case generation and execution

Based on the heart model and requirements coverage obligations, the validation becomes exploring the heart conditions to evaluate each obligation. However, it is nontrivial to identify the adaptation trajectory because the relationship between heart arrhythmias and the coverage obligations is not straightforward. A heuristic learning algorithm could be helpful to guide the exploration of heart conditions.

As an initial attempt, this study uses the on-policy SARSA (State–Action–Reward–State–Action) learning algorithm [17]. A set of states S defined as the finite set $\{s_1, ..., s_N\}$, represents the validation achievements, i.e. the coverage evaluation. A set of action A defined as the finite set $\{a_1, ..., a_M\}$, represents the configuration actions, e.g., a_1: atrial bradycardia; a_2: first-degree AV block, etc. Each action initially has a uniform probability. The heart model is placed in some condition by a randomly selected action. If the evolution of the closed loop response progresses through an *untested* requirements obligation, then the coverage will be numerically increased (towards 100%) and the corresponding action is rewarded by increasing its probability. Otherwise, its probability will be decreased. Note that the total probabilities $\sum_i p(i) = 1$. In the next iteration, a new action will be selected based on the updated probabilities. The generic learning process is shown in Fig. 4.



Start
Perform the action a_t, configure the heart model
Observe the state s_{t+1} of the closed loop system and reward r after action a_t
Select action a_{t+1} based on current experience
Update the experience $s_t \leftarrow s_{t+1}, a_t \leftarrow a_{t+1}$
Reached the Goal?
N
Y
End

Fig. 4. The automation of testing by learning.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A segment of the closed-loop system simulation in Simulink is shown in Fig. 5. The first two plots show the atrial and ventricular rhythms in beats per minute (bpm) followed by the design and clinical coverage evaluation. The reconfiguration of heart model in the last plot follows the procedure of Fig. 4. These plots show that the heart conditions have been varied over time resulting in a progressive increase of coverage.

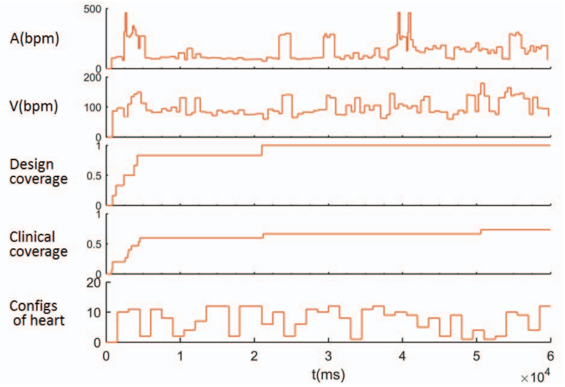


Fig. 5. Run-time coverage evaluation during validation

Fig. 6 is a snapshot of the dynamics of the closed-loop system. The encircled numbers from 1 to 4 in plot 5 denote each action/reconfiguration. Note that reconfigurations are only applied after the current cardiac cycle is completed.

During segment 1, the heart is configured for normal heart rhythm with rate variability. When the intrinsic atrial rates drop below 60 bpm, the pacemaker delivers an AP to maintain the rhythm. The configuration 2 emulates an atrioventricular block. Here the atrial activation cannot propagate to the ventricles and hence the device delivers a VP to synchronize the rhythms. Then an atrial tachycardia drives the pacemaker to deliver fast VP events to maintain the synchronization. The atrial rhythm recovers to normal in segment 4 but still with an atrioventricular block.

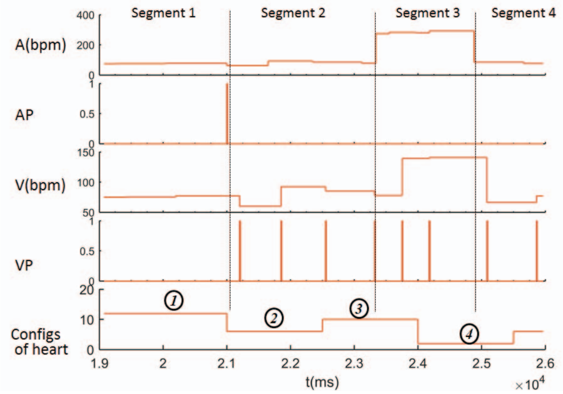


Fig. 6. A snapshot of the dynamics of the closed-loop system

With the automated oracles, we can identify scenarios where the pacemaker is unable to handle properly. Fig. 7 shows such scenario. There is a AVI component used to synchronize atrial and ventricular events in the device [16]. In scenario 1, as_1 makes the AVI component go to an intermediate state, in which as_2 is ignored. After that, the component delivers a vp_1 to synchronize as_1. At this moment, test oracle outputs a flag because the interval between as_2 and vp_1 is less than AVI. As a matter of fact, the heart presents arrhythmias like atrial flutter and DDD mode is not applicable.

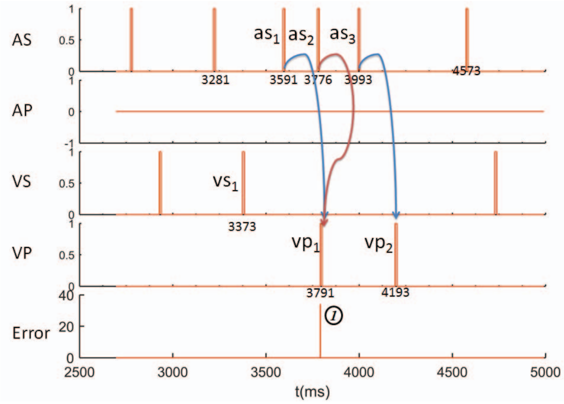


Fig. 7. The scenario of discrepancy between the requirements and the design

V. Conclusions and Future work

We have presented a closed-loop validation framework for cardiac devices that has the capability of significantly improving test adequacy by maximizing the requirements coverage. Our methodology integrates a heart model to a SUT with an adequacy evaluation module driving a heuristic learning algorithm. This integration automatically generates complex test-case sequences and hence has the potential to improve the functional safety of the SUT. This methodology could also be applied to validate other cyber-physical systems.

We intend to investigate more complex cardiac devices used in real application. More rigorous coverage criteria combination, dependency among the requirements and automated observer construction will also be studied. In addition, we will refine the learning model by considering the structure of observers and the stability of the closed-loop system to enhance the decision making process.

Acknowledgment

Authors express the gratitude to the group of Prof. Marta Kwiatkowska for sharing the Simulink heart model [7], which was foundational in developing the model in this paper.

References

[1] H. G. Mond and A. Proclemer, “The 11th world survey of cardiac pacing and implantable cardioverter-defibrillators: Calendar year 2009–a world society of arrhythmia’s project,” *Pacing and Clinical Electrophysiology*, vol. 34, no. 8, pp. 1013–1027, 2011.

[2] Z. Jiang, M. Pajic, and R. Mangharam, “Cyber–physical modeling of implantable cardiac medical devices,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 122–137, 2012.

[3] M. V. Perez, A. A. Al-Ahmad, P. J. Wang, and M. P. Turakhia, “Inappropriate pacing in a patient with managed ventricular pacing: what is the cause?” *Heart Rhythm*, vol. 7, no. 9, pp. 1336–1337, 2010.

[4] M. Almehairi, A. Baranchuk, A. Johri, and J. Caldwell, “An unusual cause of automatic mode switching in the absence of an atrial tachyarrhythmia,” *Pacing and Clinical Electrophysiology*, vol. 37, no. 6, pp. 777–780, 2014.

[5] D. Sharma, K. Narayanan, X. Wang, and M. Shehata, “An unusual form of pacemaker mediated tachycardia,” *Heart Rhythm*, vol. 2, no. 11, pp. 328–329, 2014.

[6] Z. Jiang, A. Connolly, and R. Mangharam, “Using the virtual heart model to validate the mode-switch pacemaker operation,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 6690–6693.

[7] T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre, “Quantitative verification of implantable cardiac pacemakers over hybrid heart models,” *Information and Computation*, vol. 236, pp. 87–101, 2014.

[8] M. W. Whalen, A. Rajan, M. P. Heimdahl, and S. P. Miller, “Coverage metrics for requirements-based testing,” in *Proceedings of the 2006 international symposium on Software testing and analysis*. ACM, 2006, pp. 25–36.

[9] G. Fraser and F. Wotawa, “Complementary criteria for testing temporal logic properties,” in *Tests and Proofs*. Springer, 2009, pp. 58–73.

[10] Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam, “Real-time heart model for implantable cardiac device validation and verification,” in *Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*. IEEE, 2010, pp. 239–248.

[11] M. Kwiatkowska, H. Lea-Banks, A. Mereacre, and N. Paoletti, “Formal modelling and validation of rate-adaptive pacemakers,” in *Healthcare Informatics (ICHI), 2014 IEEE International Conference on*. IEEE, 2014, pp. 23–32.

[12] K. J. Hayhurst, D. S. Veerhusen, J. J. Chilenski, and L. K. Rierson, *A practical tutorial on modified condition/decision coverage*. National Aeronautics and Space Administration, Langley Research Center, 2001.

[13] C. Pecheur, F. Raimondi, and G. Brat, “A formal analysis of requirements-based testing,” in *Proceedings of the eighteenth international symposium on Software testing and analysis*. ACM, 2009, pp. 47–56.

[14] P. Ye, E. Entcheva, S. Smolka, and R. Grosu, “Modelling excitable cells using cycle-linear hybrid automata,” *Systems Biology, IET*, vol. 2, no. 1, pp. 24–32, 2008.

[15] “Pacemaker system specification,” *Boston Scientific*, 2007.

[16] M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam, “From verification to implementation: A model translation tool and a pacemaker case study,” in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2012 IEEE 18th*. IEEE, 2012, pp. 173–184.

[17] M. van Otterlo and M. Wiering, “Reinforcement learning and markov decision processes,” in *Reinforcement Learning*. Springer, 2012, pp. 3–42.