Full text available at: http://dx.doi.org/10.1561/1000000040

High-Confidence Medical Device Software Development

Zhihao Jiang
University of Pennsylvania
USA

Rahul Mangharam
University of Pennsylvania
USA

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

Z. Jiang and R. Mangharam. *High-Confidence Medical Device Software Development*. Foundations and Trends® in Electronic Design Automation, vol. 9, no. 4, pp. 309–391, 2015.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-68083-069-9

Foundations and Trends® in Electronic Design Automation
Volume 9, Issue 4, 2015

Editorial Board

Full text available at: http://dx.doi.org/10.1561/1000000040

Editorial Scope

Topics

Foundations and Trends® in Electronic Design Automation publishes survey and tutorial articles in the following topics:

- System level design
- Behavioral synthesis
- Logic design
- Verification
- Test
- Physical design
- Circuit level design
- Reconfigurable systems
- Analog design
- Embedded software and parallel programming
- Multicore, GPU, FPGA, and heterogeneous systems
- Distributed, networked embedded systems
- Real-time and cyberphysical systems

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2015, Volume 9, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Foundations and Trends® in Electronic Design Automation
Vol. 9, No. 4 (2015) 309–391

DOI: 10.1561/1000000040

now
the essence of knowledge

High-Confidence Medical Device Software Development

Zhihao Jiang
University of Pennsylvania
USA

Rahul Mangharam
University of Pennsylvania
USA

Contents

Full text available at: http://dx.doi.org/10.1561/1000000040

Abstract

The design of bug-free and safe medical device software is challenging, especially in complex implantable devices. This is due to the device's closed-loop interaction with the patient's organs, which are stochastic physical environments. The life-critical nature and the lack of existing industry standards to enforce software validation make this an ideal domain for exploring design automation challenges for integrated functional and formal modeling with closed-loop analysis. The primary goal of high-confidence medical device software is to guarantee the device will never drive the patient into an unsafe condition even though we do not have complete understanding of the physiological plant.

There are two major differences between modeling physiology and modeling man-made systems: first, physiology is much more complex and less well-understood than man-made systems like cars and airplanes, and spans several scales from the molecular to the entire human body. Secondly, the variability between humans is orders of magnitude larger than that between two cars coming off the assembly line.

Using the implantable cardiac pacemaker as an example of closed-loop device, and the heart as the organ to be modeled, we present several of the challenges and early results in model-based device validation. We begin with detailed timed automata model of the pacemaker, based on the specifications and algorithm descriptions from Boston Scientific. For closed-loop evaluation, a real-time Virtual Heart Model (VHM) has been developed to model the electrophysiological operation of the functioning and malfunctioning (i.e., during arrhythmia) hearts. By extracting the timing properties of the heart and pacemaker device, we present a methodology to construct timed-automata models for formal model checking and functional testing of the closed-loop system. The VHM's capability of generating clinically-relevant response has been validated for a variety of common arrhythmias. Based on a set of requirements, we describe a framework of Abstraction Trees that allows for interactive and physiologically relevant closed-loop model checking and testing for basic pacemaker device operations such as maintaining the heart rate, atrial-ventricle synchrony and complex conditions such as avoiding pacemaker-mediated tachycardia.

Through automatic model translation of abstract models to simulation-based testing and code generation for platform-level testing, this model-based design approach ensures the closed-loop safety properties are retained through the design toolchain and facilitates the development of verified software from verified models. This system is a step toward a validation and testing approach for medical cyber-physical systems with the patient-in-the-loop.

Z. Jiang and R. Mangharam. *High-Confidence Medical Device Software Development*. Foundations and Trends® in Electronic Design Automation, vol. 9, no. 4, pp. 309–391, 2015. DOI: 10.1561/1000000040.

1

Medical Devices: Current State and Challenges

The medical device market is worth $289 billion, of which $110 billion is from the US alone, with this number projected to reach $133 billion in 2016. Examples include everything from adhesive bandages, stents, artificial joints, drug infusion pumps to surgical robots, implantable cardiac pacemakers, and devices still undergoing basic research like the artificial pancreas. To take one example of the societal impact of medical devices, an estimated 3 million people worldwide have implanted cardiac pacemakers (a heart rate adjustment device), with 600,000 added annually. Clinical trials have presented evidence that patients implanted with cardiac defibrillators (another heart rate adjustment device) have a mortality rate reduced by up to 31%. Implanted cardiac pacemakers and defibrillators have approximately 80,000-100,000 lines of software code which essentially makes all sensing, control and actuation decisions autonomously within the human body, over the 5-7 year device lifetime [1]. With the increasing complexity of combining hardware and software in a large class of these life-saving technologies, there is an urgent need for approaches to rigorously validate the device and therapy to be safe and efficacious.

[1] Paul L. Jones. Senior Systems/Software Engineer, Office of Science and Engineering Laboratories, U. S. FDA. Personal communication, 2010.

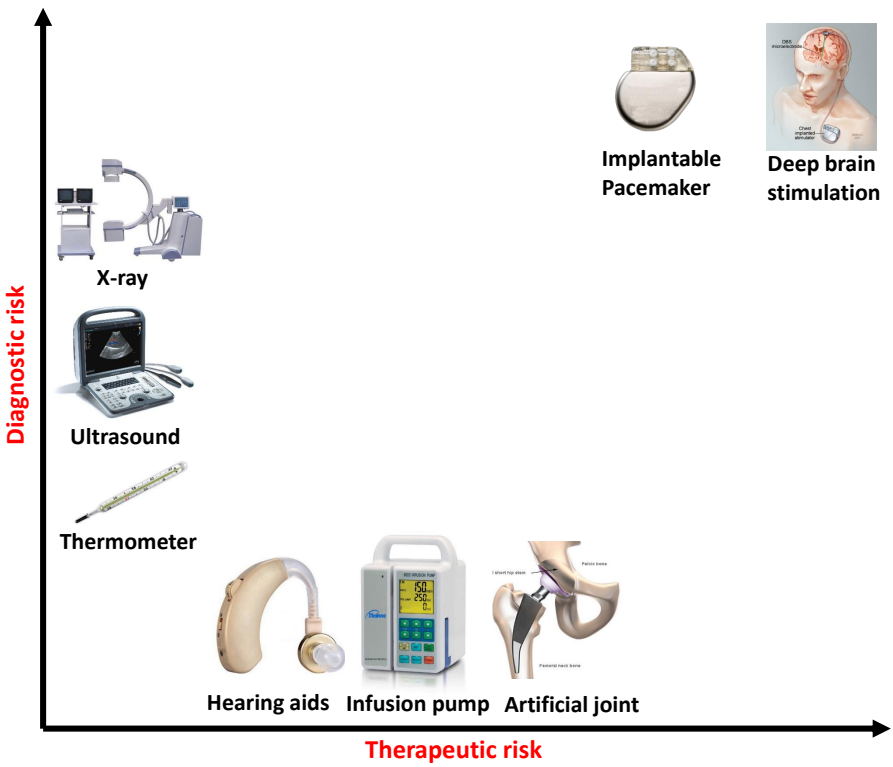


Figure 1.1: Current medical devices across a range of diagnostic and therapeutic risk. Implantable software-controlled devices such as the pacemaker and defibrillator which operate in a closed-loop of sensing, control and actuation are amongst the highest risk

The US Food and Drug Administration defines a medical device as an instrument, apparatus, implement, machine, or implant which is:

- intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease, in humans or other animals, or
- intended to affect the structure or any function of the human body or other animals, and which does not achieve any of its primary intended purposes through chemical action and which is not dependent upon being metabolized for the achievement of any of its primary intended purposes."

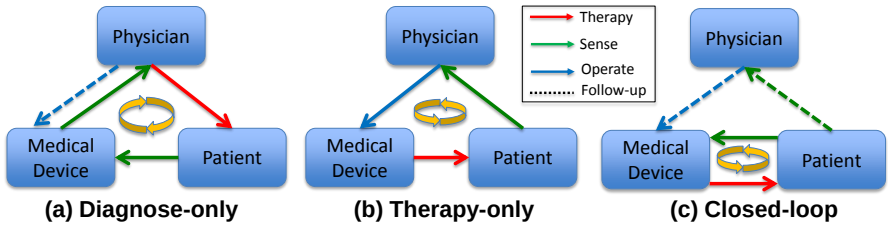


Figure 1.2: Diagnostic-only and therapy-only devices do not interact with the patient in direct closed-loop. The physician is responsible for the diagnostic and/or therapeutic decisions. However in closed-loop medical devices, the devices interact with the patient in closed-loop and have to make therapeutic decisions based on their own diagnosis.

In general, medical devices are categorized according to their risk factors - Class I, Class II and Class III, corresponding to low-risk, medium-risk and high-risk devices (Food and Administration [2014]). Fig. 1.1 gives an intuitive description of medical devices examples across a range of diagnostic and therapeutic risk.

1.1 Closing the Device-Patient Loop

Medical devices operate across a range of invasiveness and intervention with the patient in the loop. For diagnostic-only devices, like an X-ray machine, the physician operates the device to obtain patient data. Upon interpretation of the data, the physician performs diagnosis followed by delivery of proper therapy to the patient (Fig. 1.2.(a)). For therapy-only devices, e.g. a drug infusion pump, the physician configures the device infrequently based on prior diagnosis of the patient so the device executes the therapy on the patient (Fig. 1.2.(b)). We denote these devices as **Open-loop Medical Devices** as there is no direct feedback loop between the patient and the device. For open-loop devices, the device operates under the supervision of professionally-trained physicians. The device's safety is mostly determined by how accurately it provides information to the physicians or how faithfully it operates as instructed by the physicians.

There is a class of devices with both diagnostic and therapeutic functions, i.e. implantable cardiac devices to treat cardiac arrhythmia, deep brain stimulation devices (Coffey [2009]) to treat Parkinson's disease and artificial

pancreas to treat Type-1 diabetes. These devices capture and diagnose the patient's physiological conditions from sensory data, *and* deliver therapy in response (Fig. 1.2.(c)). These devices usually operate (semi-) autonomously with very little human intervention. Although therapies can be delivered more timely with these devices, malfunctions or inappropriate therapies from these devices also cannot be corrected timely, which can cause serious adverse effects on patients' health. Therefore, these devices are usually classified into the highest risk category and undergo the most stringent regulation. We denote them as **Closed-loop Medical Devices**.

There are multiple challenges to develop safe and effective closed-loop medical devices:

1.1.1 Closed-loop Interactions with Complex Physiology

When using open-loop medical devices, the diagnosis and therapy decisions are made by medical professionals, who have expert knowledge of human physiology. Therefore they are able to identify adverse health conditions and adjust the therapy accordingly. On the other hand, closed-loop medical devices have to make both the diagnosis and therapy decisions on their own. The domain expertise required to make those decisions has to be programmed into the device. It is impossible to encode all the knowledge of human physiology into the device. Therefore, for unanticipated physiological conditions, when the appropriate response has not been programmed into the device, the device may deliver inappropriate therapy which can have an adverse effect on patient's health.

Technological development of materials, sensors, embedded computing, energy storage, communications and packaging usher new closed-loop therapies (e.g. deep brain stimulation). While the spectrum of closed-loop interactions between the device and the human physiology may not be fully understood, the challenge is to ensure the device never drives the patient into an adverse state under all physiological conditions. Furthermore, the incremental addition of new therapies in legacy devices (e.g. cardiac rhythm therapy), may result in conflicted diagnostics and behavior of the device for well-understood behaviors and result in inappropriate and unsafe operations.

1.1.2 Limited Diagnostic and Therapeutic Functions

One fundamental rationale behind closed-loop medical devices is to enable patients to live their lives normally with limited explicit interaction with the device, and also with minimal physician supervision. In fact, a large number of closed-loop medical devices are autonomous implantable devices. As a result, the sensing and therapy capabilities of these devices are limited, in order to minimize power consumption, heat dissipation and invasiveness. Limited sensing capabilities, and hence limited observability, may cause misdiagnosis as the device may be unable to distinguish the source between two sensed signals from different conditions that now seem similar and result inappropriate therapy. Due to limited therapeutic capabilities, there exists sub-optimal physiological conditions that are untreatable. The device may even drive the body to a less optimal state by over-treating the patient by preempting the body's natural response. In later chapters, we will describe examples in which an untreatable condition is deteriorated into an adverse condition due to the device interaction.

1.1.3 Software-related Medical Device Recalls

Due to the complexity of the diagnostic and therapeutic functions of the closed-loop devices, these functions are mostly controlled by their software components. Software embedded in a medical device, unlike electrical and mechanical components, does not fail due to corrosion, fatigue or have statistical failures of subcomponents. Software failures are uniquely sourced in the design and development of the system. According to the US Food and Drug Administration, in 1996, 10% of all medical device recalls were caused by software-related issues (Maisel et al. [2001]). This percentage rose to an average of 15% of recalls from 2008 to 2012 (Fig. 1.3). Malfunctions of closed-loop medical devices usually have severe consequences, which will be categorized as *Class I*, meaning there is a "reasonable probability that use of these products will cause serious adverse health consequences or death." (Food and Administration [2006], Zhang et al. [2015], Sandler et al. [2010]).

	Software change control	Software Design	Software design manufacturing process	Sum	% of all CDRH recalls
2008	13	141	2	**156**	**18.3%**
2009	9	111	1	**121**	**15.4%**
2010	4	73	3	**80**	**8.9%**
2011	11	182	10	**203**	**15.8%**
2012	12	169	5	**186**	**15.5%**
Sum	**49**	**676**	**21**	**746**	**15.1%**

Figure 1.3: Medical device recalls due to software issues have risen from 10% in the 1990s to ~15% in the past decade (Food and Administration [2012])

1.2 Medical Device Regulation Efforts and Challenges

The medical device industry is regulated to ensure the safety of the patients and the public. In the United States, the FDA is the primary regulatory authority responsible for assuring the safety, efficacy and security of patients using medical devices. Based on the rationale that 1) manufacturers know their devices better than the regulator, and 2) the variety of medical devices requires a variety of approaches, it is the device manufacturers' responsibility to demonstrate the safety and efficacy of the medical devices. Manufacturers are required to complete a pre-market submission before the devices can be released to the market. The level of requirements for the submission is determined by the safety classification of the devices. A set of general guidelines are recommended by the FDA (Food and Administration [1997, 2002, 2005]) which list the activities that need to be performed to ensure device safety.

In safety-critical industries such as automotive electronics, avionics and nuclear systems, international standards are enforced for software system development, evaluation, manufacturing and post-market changes (Fürst et al. [2009], Feiler et al. [2010]). This awareness is only beginning to enter the medical device industry as compliance with international standards are "recommended" in the aforementioned guidelines (Jetley et al. [2006]) but the burden of their interpretation and enforcement is on the device manufacturer. The basic rationale behind these standards is that: if all the risks/hazards of the device are identified and reasonably mitigated, and the device is developed with rigorous process, the device is *reasonably safe*.

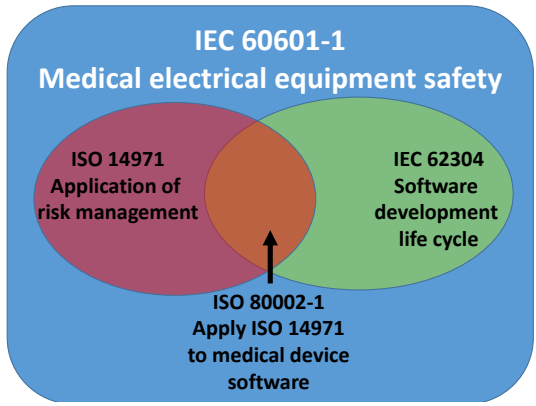


Figure 1.4: International standards for medical device safety. These standards define the required activities during the development process.

Fig. 1.4 describes the primary standards to ensure medical device safety and their relationships. The IEC 60601 Medical Electrical Equipment - General requirements for basic safety and essential performance is a product safety standard that all electronic medical devices must comply to. IEC 60324 specifies the processes and activities needed to perform during the software development life cycle to ensure software safety.

Risk management is a core activity throughout the software development life cycle. ISO 14971 is specified for the application of risk management to medical devices. In addition, for each risk management activity of ISO 14971, ISO 80002-1 provides additional guidelines for the software component, which highlights and explains approaches to assuring that software safety is adequately addressed.

1.2.1 Risk Management Challenges of Closed-loop Systems

While it is not normally possible to develop a device that is safe with a probability of 100% under all physiological and operating conditions, approaching the problem along the lines of risk analysis, risk evaluation and risk control helps better address a “designed-for-safety" mindset. Fault Tree Analysis (FTA) is a common tool in risk analysis in which hazards of the system are first identified and the possible causes of the hazards are analyzed until the initial faults are reached. Fig. 1.5.(a) demonstrate an example fault tree for

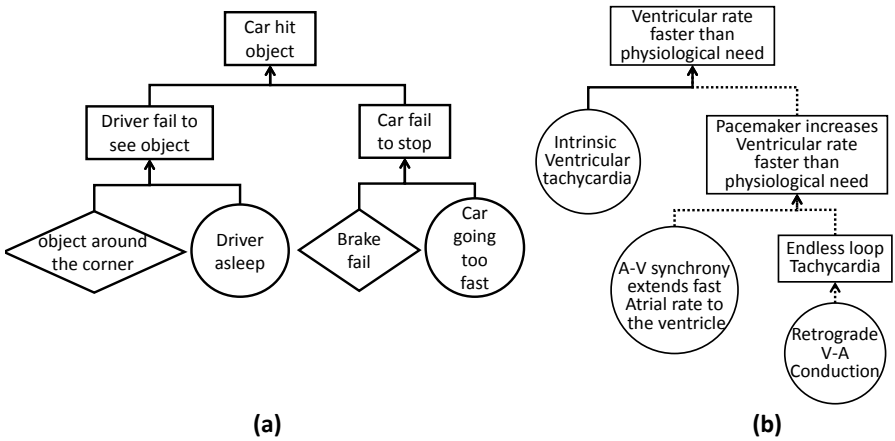


Figure 1.5: Fault Tree Analysis (FTA) Examples. (a) FTA for a hazard for a car; (b) FTA for a hazard in implantable pacemaker. The dashed line shows two mechanisms that were not identified during hazard analysis but discovered in post-market studies.

automobile. FTA is very good at showing how resistant a system is to single or multiple initiating faults. It is not good at finding all possible initiating faults since causes are conjectured and analyzed manually.

In closed-loop medical devices, there may exist interactions between the device and the patient that can cause certain hazard, but are unknown due to limits in physiological knowledge, behavior not captured in patient trials and the separation of the software development teams and the medical domain experts. Fig. 1.5.(b) describes an example fault tree for a hazard for an implantable pacemaker. There are several causes for undesirable fast ventricular rate. The well-understood cause is the intrinsic ventricular tachycardia (solid line). However, with pacemaker implanted, new mechanisms to cause hazard are introduced into the closed-loop system, as illustrated by the two branches with dotted lines. These two branches were not identified during the initial fault tree analysis, and were only identified after the devices have been released into the market, causing unnecessary adverse effects to the patients Furman and Fisher [1982]. Risks identified at this late stage are also more costly to fix, increasing the cost for device development.

After the fault tree has been constructed, probabilities for the initial faults are analyzed bottom up to calculate the probability of each hazard. The tech-

Probability of occurrence	Severity I Catastrophic (death, serious injury)	Severity II Significant (Reversible serious injury)	Severity III Marginal (inconvenience)	Severity IV Negeligible
Frequent	1	3	7	13
Probable	2	5	9	16
Occasional	4	6	11	18
Remote	8	10	14	19
Improbable	12	15	17	20

Hazard Risk Index	Acceptance Criteria
1 to 5	Unacceptable
6 to 9	Undesirable: Written and reviewed decision required
10 to 16	Acceptable upon completion of quality assurance review
17 to 20	Acceptable without review

Figure 1.6: Top table: Risk index according to occurrence and severity. Bottom table: Risk control using risk index

nique is called Failure Mode and Effects Analysis (FMEA). Then the risks are evaluated by assigning risk index to each hazard according to their occurrence and severity (Fig. 1.6). After the risks are evaluated, different activities are required to mitigate the risks according to the risk index. The risks are then be re-evaluated to calculate the residual risk and analyze the risk/benefit. This is part of the risk control process. FMEA is good at exhaustively cataloging initiating faults, and identifying their local effects. It is not good at examining multiple failures or their effects at a system level.

1.2.2 Pre-Market Evaluation with Clinical Trials

Regardless of how rigorous the risk management and the device development process are, the devices have to be able to achieve their design goal on the real patient, which can only be evaluated within its physiological environment. Devices that have high risk factors, including the closed-loop medical devices, are required to submit clinical evidence for their safety and efficacy, often in form of clinical trials. In clinical trials, the devices are used on a preselected population of patients following carefully-designed protocols. The goal of a medical trial, in part, is to obtain unambiguous results for the pri-

mary question of the trial which can support the safety and/or efficacy of the devices. However, conducting clinical trials is very time consuming and expensive, and risks found during clinical trials are very expensive to fix (U. S. Food and Drug Administration [2013]).

To address this **safety gap** between ensuring the device satisfies its therapeutic requirements with the patient-in-the-loop and testing its software specifications, new approaches for closed-loop validation of the device software within the physiological context are needed - this is the primary focus of this article.

1.3 Model-based design to improve medical device safety

With the deluge of software-based closed-loop medical devices in the coming years, relying on clinical trials as the only closed-loop evaluation method to identify risks rooted in device software is not scalable. Model-based design and virtual integration have been proposed and applied in other industries like automotive and avionics (Fürst et al. [2009], Feiler et al. [2010]), and can potentially help during the development process and provide extra confidence to the device before conducting clinical trials. However, unlike man-made systems like automobiles and aircrafts, physiological systems are less understood with larger variations for the type and degree of patient conditions. The lack of faithful models of physiological environment of the closed-loop medical devices is one of the reason that model-based design is not well-adopted in the medical device industry.

As computational models of human physiology are developed, they can be used to interact with closed-loop medical devices or their models. The FDA is starting to recognize in-silico modeling and simulation as regulatory-grade evidence for device safety and efficacy. For example, Ghorbani and Bogdan [2013] developed glucose-insulin models that can be used to evaluate control algorithms for artificial pancreas devices which can sense blood glucose and deliver insulin. Simulation results with the models have been recognized by FDA to replace animal trials, in part, which significantly reduced cost (B. P. Kovatchev and M. Breton and C. Dalla Man and C. Cobelli [2009]). With the increasing interest and recognition from the regulators, computer models and simulations are expected to play bigger role as as

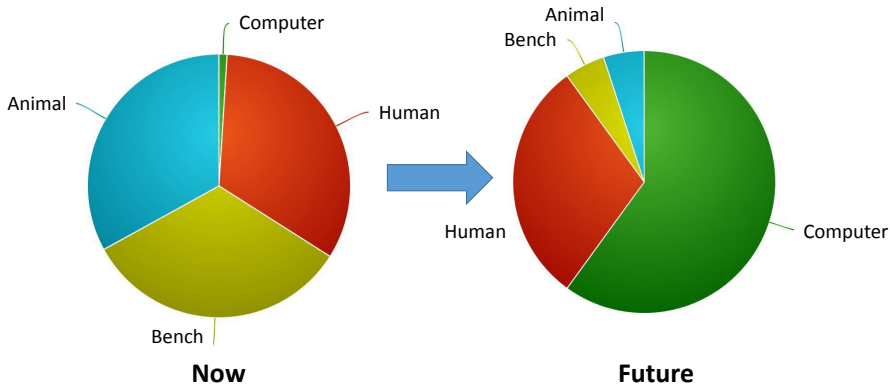


Figure 1.7: Percentage of computer simulation is expected to increase as safety and effectiveness evidence of medical devices

"regulatory-grade evidence" evidence in the development of future closed-loop medical devices (Fig. 1.7).

1.4 Contributions

In this article, we use an implantable cardiac pacemaker as a working example to demonstrate how model-based design can help improve the safety and efficacy medical device software. We demonstrate the application of model-based design in several design activities during the development process, from the perspective of the manufacturer's design validation team. We assume availability of design artifacts including pacemaker design and physiological requirements. By demonstrating the process of developing verified models to generate verified code, the results of our model-based closed-loop evaluation should be able to support the device's safety and efficacy requirements during the regulation process.

Our proposed model-driven design for closed-loop medical devices (Fig. 1.8) begins with developing heart models that can interact with real and modeled pacemakers (Jiang et al. [2012a]). In Chapter 2, we introduce our heart models for closed-loop *model checking* and *testing* of implantable cardiac devices, and the rationale for the difference between heart models used in these two applications. For closed-loop evaluation, the heart models have to be able to represent and respond under different physiological conditions.

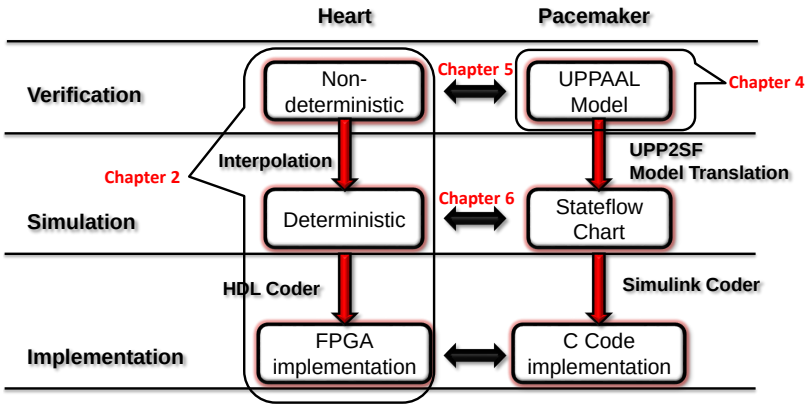


Figure 1.8: Model-driven design for verified models to verified code for the closed-loop heart and pacemaker system

The heart models are available in different formalisms to interact with the pacemaker design in closed-loop across different design stages. In Chapter 3, we validate the heart models and discuss how to identify model parameters from patient data so that the heart model can represent different physiological conditions.

In Chapter 4, we introduce the pacemaker software specification which is referenced from a dual chamber pacemaker design from Boston Scientific (Boston Scientific Corporation [2007b]). The software specification is converted to an abstract formalism called *Timed Automata* (Alur and Dill [1994]). The timed automata model of the pacemaker will be the starting point for our model-based analysis and implementation. In Chapter 5, we identify two basic hazards for pacemaker and use the UPPAAL model checker (Larsen et al. [1997]) to evaluate whether the hazards have been reasonably mitigated. With the help of heart models introduced in Chapter 2, we are able to cover the closed-loop behaviors of large variety of heart conditions so that we can evaluate whether there exists any known and even unknown mechanism to induce hazards (Jiang et al. [2014]). Pacemaker and heart models used in model-checking are abstract as model checkers do not scale well with increased model complexity. So complex dynamics of the heart and pacemaker are not captured at this stage.

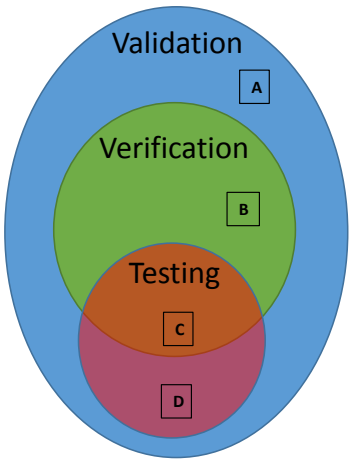
In Chapter 6, we describe the development of an automatic model translation procedure to translate models from UPPAAL to Stateflow (Inc. [2016]) to ensures that abstract models used for verification over-approximate the more detailed models used downstream (Pajic et al. [2012]). The Stateflow model of the pacemaker is then evaluated with heart models with relatively complex dynamics (Jiang et al. [2010], Jiang and Mangharam [2011], Jiang et al. [2011]). Once the detailed models pass simulation-based testing with closed-loop dynamics, they are automatically generated into code and are subject to platform-level integration testing (Jiang and Mangharam [2016]). This model-driven design approach ensures the closed-loop safety properties are retained through the design toolchain and facilitates the development of verified software from verified models.

1.5 Useful terminologies for often misinterpreted terms

Ensuring the safety of complex medical devices has drawn interest not only from stakeholders like regulators and industries, but also medical professionals and academia. Different communities have different interpretations over certain terminologies, often causing misunderstandings. In this paper we adopt the terminologies from the regulation perspective, so that the results we have fit into the regulation framework. Most of the definitions are referred from the FDA guideline document General Principles of Software Validation (Food and Administration [2002]). Below are several terminologies that we use throughout the paper which worth clarifying.

1.5.1 Requirements vs. Specifications

By the definition of FDA (Food and Administration [2005]), the requirements of a system describe **what** the system should achieve and the specifications of a system describe **how** the system is designed to satisfy the requirements. For example, a requirement for an autonomous car is "The car should not hit objects". The corresponding specification can be "brake if the speed of the car is greater than x and the distance to the object is less than y". We can see that a car satisfying its specification may not satisfy the requirement (e.g. when the car is driving too fast or the obstacle pops up right in front of the car). In this paper, we use the word requirement in particular to denote the intended uses of the medical devices to improve physiological conditions.



Validation activities	Zone
Planning	A
Requirements	A
Traceability	A,B
Change management	A
User site testing	D
Defect resolution	A,B
Risk management	A
Intended use	A
Evaluations	B
Design reviews	B
System testing	C
Regression testing	C,D

Figure 1.9: Validation activities during the software development life cycle (D A. Vogel [2011])

1.5.2 Validation vs. Verification vs. Testing

As defined in Food and Administration [2002], software validation is the confirmation by examination and provision of objective evidence that:

1. software specifications conform to user needs and intended uses, and
2. the particular requirements implemented through software can be consistently fulfilled

The first aspect ensures the device is safe and effective. The second aspect maintains the traceability of requirements throughout the development life cycle. Software verification fulfills the second aspect of software validation by "providing objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. "

Testing is the technique that can be used for validation and/or verification. Fig. 1.9 illustrates the relationship between validation, verification and testing, and different activities during the software development life cycle to ensure the safety and effectiveness of the software.

1.5.3 Closed-loop vs. Open-loop Evaluation

In open-loop evaluation, i.e. open-loop testing, input sequences are send to the system and system outputs are compared with expected outputs. In open-loop testing, the system outputs do not affect the inputs afterward. In closed-loop evaluation, the environment of the system is taken into account. System outputs affect the state of the environment and thus affect the input sequences. For closed-loop medical devices, clinical trials are currently the most common closed-loop evaluation method. Enable closed-loop evaluation at model level requires models of the environment, which is human physiology for closed-loop medical devices.

Closed-loop evaluation accomplishes two goals in model-based design: 1) It enforces environmental constraints so that the test space is smaller and the test cases have physiological relevance. 2) Execution traces can be better interpreted as the physiological models encode domain knowledge.

References

R. Alur and D. L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

B. P. Kovatchev and M. Breton and C. Dalla Man and C. Cobelli. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes. *Journal of Diabetes Science and Technology*, 3, 2009.

J. Beaumont, D. C. Michaels, M. Delmar, J. Davidenko, and J. Jalife. A Model Study of Changes in Excitability of Ventricular Muscle Cells. *American Journal of Physiology. 268*, 1995.

G. Behrmann, A. David, and K. G. Larsen. A Tutorial on UPPAAL. *Formal Methods for the Design of Real-Time Systems, Lecture Notes in Computer Science*, pages 200–236, 2004.

P. Bogdan, S. Jain, and R. Marculescu. Pacemaker control of heart rate variability: A cyber physical system perspective. *ACM Transactions on Embedded Computing Systems*, 12(1s):50:1–50:22, 2013.

Boston Scientific Corporation. PACEMAKER System Specification. Boston Scientific. *Device Documentation*, 2007a.

Boston Scientific Corporation. The Compass - Technical Guide to Boston Scientific Cardiac Rhythm Management Products. *Device Documentation*, 2007b.

E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counter Example-Guided Abstraction Refinement for Symbolic Model Checking. *Journal of the ACM*, 50 (5):752–794, 2003.

E. M. Clarke and E. A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *Logic of Programs, Workshop*, pages 52–71, 1982.

E. M. Clarke, O. Grumberg, and D. E. Long. Model Checking and Abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, 1994.

R J. Coffey. Deep brain stimulation devices: A brief technical history and review. *Artificial Organs*, 33(3):208–220, 2009.

D A. Vogel. *Medical Devices Software: Verification, Validation and Compliance*. Artech House, 2011.

E.W. Hsu and C.S. Henriquez. Myocardial fiber orientation mapping using reduced-encoding diffusion tensor imaging. *Journal of Cardiovascular Magnetic Resonance*, 3(4):339–347, 2011.

P. Feiler, L. Wrage, and J. Hansson. System architecture virtual integration: A case study. *Embedded Real-time Software and Systems Conference*, 2010.

U. S. Food and Drug Administration. Design Control Guidance For Medical Device Manufacturers. *Center for Devices and Radiological Health*, 1997.

U. S. Food and Drug Administration. General principles of software validation; final guidance for industry and fda staff. *Center for Devices and Radiological Health*, 2002.

U. S. Food and Drug Administration. Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. *Center for Devices and Radiological Health*, 2005.

U. S. Food and Drug Administration. Ensuring the Safety of Marketed Medical Devices: CDRH's Medical Device Postmarket Safety Program. *Center for Devices and Radiological Health*, Jan 2006.

U. S. Food and Drug Administration. Medical device recall report - fy2003 to fy2012. *Center for Devices and Radiological Health*, 2012.

U. S. Food and Drug Administration. Classification of medical devices. *US FDA documents*, 2014.

B. Fuertes and J. Toquero. Pacemaker Lead Displacement: Mechanisms And Management. *Indian Pacing Electrophysiology Journal*, 2003.

S. Furman and J. D. Fisher. Endless loop tachycardia in an av universal (ddd) pacemaker. *Pacing and Clinical Electrophysiology*, 5(4):486–489, 1982.

S. Fürst, J. Mössinger, S. Bunzel, T. Weber, F. Kirschke-Biller, P. Heitkämper, G. Kinkelin, K. Nishikawa, and K. Lange. Autosar–a worldwide standard is on the road. In *14th International VDI Congress Electronic Systems for Vehicles*, volume 62, 2009.

M. Ghorbani and P. Bogdan. A cyber-physical system approach to artificial pancreas design. In *2013 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pages 1–10, 2013.

R. Grosu, G. Batt, F. H. Fenton, J. Glimm, C. Le Guernic, S.A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *Computer Aided Verification*, volume 6806 of *Lecture Notes in Computer Science*, pages 396–411. Springer Berlin Heidelberg, 2011.

Mathworks Inc. Matlab R2011a Stateflow Documentation. http://www.mathworks.com/help/toolbox/stateflow, 2016.

Md. A. Islam, A. Murthy, A. Girard, S. A. Smolka, and R. Grosu. Compositionality results for cardiac cell dynamics. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control*, (HSCC '14), pages 243–252, 2014.

R. Jetley, S. P. Iyer, and P. L. Jones. A Formal Methods Approach to Medical Device Review. *IEEE Computer*, 39:61–67, 2006.

Z. Jiang and R. Mangharam. Modeling Cardiac Pacemaker Malfunctions with the Virtual Heart Model. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 263 –266, Sept 2011.

Z. Jiang and R. Mangharam. Virtual Heart Model website - http://medcps.org, 2016.

Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam. Real-time heart model for implantable cardiac device validation and verification. In *2010 22nd Euromicro Conference on Real-Time Systems (ECRTS)*, pages 239 –248, July 2010.

Z. Jiang, M. Pajic, and R. Mangharam. Model-based Closed-loop Testing of Implantable Pacemakers. In *ACM/IEEE 2nd International Conference on Cyber-Physical Systems (ICCP'11)*, 2011.

Z. Jiang, M. Pajic, and R. Mangharam. Cyber-Physical Modeling of Implantable Cardiac Medical Devices. *Proceedings of the IEEE*, 100(1):122 –137, Jan. 2012a.

Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam. Modeling and Verification of a Dual Chamber Implantable Pacemaker. *Tools and Algorithms for the Construction and Analysis of Systems*, 7214:188–203, 2012b.

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

Z. Jiang, H. Abbas, PJ. Mosterman, and R. Mangharam. Tech Report: Abstraction-Tree For Closed-loop Model Checking of Medical Devices. *http://repository.upenn.edu/mlab_papers/73*, 2015.

M. E. Josephson. *Clinical Cardiac Electrophysiology*. Lippincot Williams and Wilkins, 2008.

K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a Nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, pages 134–152, 1997.

W. H. Maisel, M. O. Sweeney, W. G. Stevenson, K. E. Ellison, and L. M. Epstein. Recalls and Safety Alerts involving Pacemakers and Implantable Cardioverter-Defibrillator Generators. *JAMA*, 286(7), 2001.

A. Murthy, E. Bartocci, F. H. Fenton, J. Glimm, R. A. Gray, E. M. Cherry, S. A. Smolka, and R. Grosu. Curvature analysis of cardiac excitation wavefronts. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2): 323–336, 2013.

Nano-RK. Nano-RK Sensor RTOS, Carnegie Mellon University. *http://nanork.org*, 2007.

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 *Proceedings of the 2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium*, RTAS '12, pages 173–184, 2012.

M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam. Safety-critical medical device development using the upp2sf model translation tool. *ACM Transactions on Embedded Computing Systems*, 13(4s):127:1–127:26, 2014.

C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart. 1. immersed elastic fibers in a viscous incompressible fluid. *Journal of Computer Physics*, 81(2):372–405, 1989.

S. Rossi, R. Ruiz-Baier, L. F. Pavarino, and A. Quarteroni. Active strain and activation models in cardiac electromechanics. *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 11(1):119–120, 2011.

F. B. Sachse, A. P. Moreno, and J. A. Abildskov. Electrophysiological modeling of fibroblasts and their interaction with myocytes. *Annals of Biomedical Engineering*, 36(1):41–56, 2008.

K. Sandler, L. Ohrstrom, L. Moy, and R. McVay. Killed by Code: Software Transparency in Implantable Medical Devices. *Software Freedom Law Center*, 2010.

S. J. Saxonhouse, J. B. Conti, and A. B. Curis. Current of injury predicts adequate active lead fixation in permanent pacemaker/defibrillation leads. *Journal of the American college of Cardiology*, 2005.

R. Schulte, G. Sands, F. Sachse, O. Dossel, and A. Pullan. Creation of a Human Heart, Model and its Customisation using Ultrasound Images. *Biomedizinische Technik/Biomedical Engineering*, 46:26–28, 2001.

N. A. Trayanova and P. M. Boyle. Advances in modeling ventricular arrhythmias: from mechanisms to the clinic. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(2):209–224, 2014.

U. S. Food and Drug Administration. Human Subject Protection; Acceptance of Data from Clinical Studies for Medical Devices; Proposed Rule. *Docket No. FDA-2013-N-0080*, 2013.

S. Yamane. Timed Weak Simulation Verification and its Application to Stepwise Refinement of Real Time Software. *International Journal of Computer Science and Network Security*, 6, 2006.

S. Zhang, C. Kriza, S. Schaller, and P. L. Kolominsky-Rabas. Recalls of cardiac implants in the last decade: what lessons can we learn? *PLoS ONE 10(5): e0125987.*, 2015. .