William Banks

CS-6388 Model-Integrated Computing

Final Project - Report

12/14/2017

Model-based design for medical devices

Background

Medical devices presented a uniquely challenging design environment. Strict safety considerations must be enforced and utilizing an appropriate test environment is a non-trivial issue. While medical devices cover a wide range of modalities, from adhesive bandages to artificial joins and surgical robots, patient critical devices such as pacemakers represent a sizeable portion. An estimated 3 million individuals worldwide have an implanted pacemaker and that number is increasing by 600,000 annually. Pacemakers represent 10s of thousands of lines of code and must be relied upon to safely and effectively execute autonomous actions within the body over a 5 to 7 -year period [1]. Given this understandable complexity, it is imperative that a rigorous approach to validate both device and treatment be developed.

Medical devices can be categorized into 2 branching categories: open-loop and closed-loop. Open-loop devices such as X-ray machines operate under direction of a supervising physician. For these devices, safety is primarily a function of accuracy. Closed-loop devices operate without direct supervision and therefore must be capable of its own decision making. These types of devices incorporate both diagnostic and therapeutic functionality. For example, the cardiac pacemaker must capture timing information from the heart muscle excitation potential and diagnose any arrhythmia by pacing the heart to a safe level. Because these devices are relied upon to operate mostly autonomously, any malfunction cannot be corrected in a timely manner. Because of this, these types of devices must adhere to the most extensive level of safety regulations.

While other engineering domains possess well defined systems (automotive and aeronautical being chief examples) human physiological models are not well developed. This results in a poor testing environment. For example, every individual part of a car can be modeled for thermal and mechanical strain, electrical interference, etc. Open-loop devices such as the X-ray machine have the benefit of a trained professional physicians with years of education in human physiology to guide treatment and interpret results. In closed-loop systems, devices do not have the benefit of a well-defined physiological model. The pacemaker is limited to a small parameter space (possesses only 2 sensing nodes in fixed places), must be able to ignore erroneous physiological signals and administer therapy within safe levels.

Any medical device must validate design goals in clinical trials with real patients. Because of the extreme safety considerations and need to obtain unambiguous data, clinical trials are very limited in scope. Therefore, clinical trials are an expensive and time-consuming process. Furthermore, because there is no well define physiological model for testing of closed-loop devices, there exists a safety gap between the design phase and the clinical trial phase. Developing complete physiological models for testing closed-loop medical devices is imperative to streamlining the development cycle.

Model and Problem

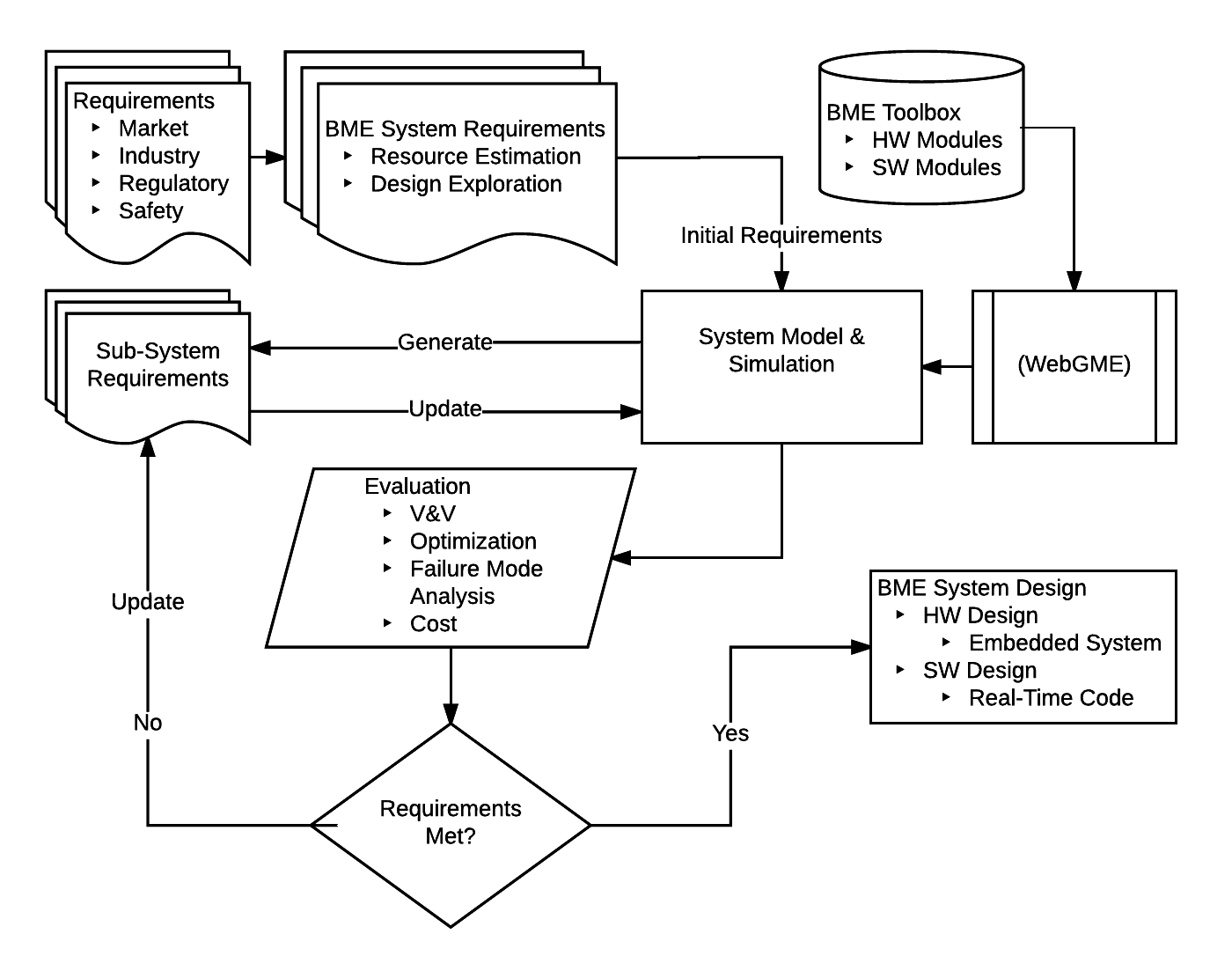


Figure 1 Medical device design flow chart

Initially, the focus of the WebGME DSML was to describe a total biomedical device domain. This domain would incorporate physical design requirements and safety requirements. After a domain is established, components adhering to safety regulations would be created and validated. The presumed advantage to this validation step to create ‘approved’ components is to limit the amount of reevaluation required in designing new devices. For example, instead of having to validate that 2 components work together, the designer could select components from a list of preapproved components which must work together by definition.

These well-defined and validated components would then be used to create some ‘medical device’ as a proof-of-concept. The DSML could then be integrated into an overall design paradigm (see fig 1).

An initial problem was identified to incorporate all major design phases.

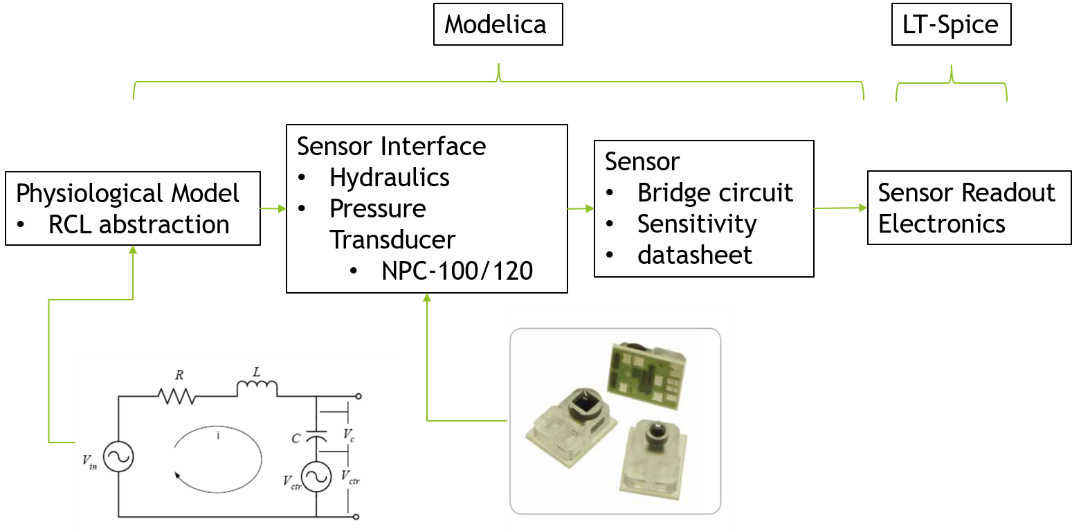


Figure 2 Example proof-of-concept model

A RCL abstraction of a venous signal would serve as the physiological model. The pressure transducer would be the medical device. A plugin would translate the WebGME model into a Modelica file. Modelica would be used to create piecewise linear function (PWL) file to import into LT-Spice for further analysis. In total, this represented a complete proof-of-concept system. It has the physiological model, the medical device and the data analysis.

An initial WebGME meta model was developed.

A close up of a map

Description generated with high confidence

Figure 3 Initial metamodel design

The objective of this meta model was to develop a framework to interconnect electrical and mechanical components and allow for recursive development of new components. For example, bridge circuit might be built from the existing resistor component. This new bridge circuit might be included as a lead in to a pressure transducer component, etc. In development, it proved difficult to couple the two domains via a Modelica translator plugin.

After consulting with Dr. Sztipanovits and my project advisor, Dr. Baudenbacher, I concluded that my project scope was too large and needed refinement.

I refined the scope of the project to focus on physiological models and a domain to develop physiological analogs. Developing physiological models is a primary building block for leveraging model-based design. Electrical analogs of physiological systems proved attractive, specifically a RCL abstraction of a venous pressure signal.

I simplified the metamodel to reflect the newly revised scope. This model incorporates an electrical domain to create analog electrical components.

A screenshot of a computer

Description generated with very high confidence

Figure 4 New metamodel design

The metamodel defines a circuit which is comprised of components. Components can be connected via electrical connections. Different components are defined as subcomponents based on the number of pins required. This is so pin placement automatically updates for new components created. A set of simple analog subcomponents is defined. Each subcomponent has a name attribute, parameter attributes (usually just the unit), and a Modelica string attribute. The Modelica string attribute is used to define each component in a way that Modelica can read.

These components are used to create the RCL analog.

A picture containing text

Description generated with high confidence

Figure 5 2nd order RCL venous waveform abstraction [2]

This RCL circuit, as described in Patel (2008) is a physiological model of a venous pressure wave This is an appropriate model of sufficient complexity because the pressure transducer (for example) only needs to know what it is measuring.

In WebGME, the RCL circuit is represented as follows

A screenshot of a cell phone

Description generated with very high confidence

Figure 6 2nd order RCL venous waveform abstraction in WebGME

Each component has an appropriate icon for visual clarity. The electrical connections are represented as dotted lines. A plugin runs that traces the model and creates a list of components and their respective attributes and connections. This is then translated into the Modelica model file format. This creates a \*.mo file which can be opened and processed with Modelica [3]. The WebGME help documentation was used heavily for this portion.

A close up of a map

Description generated with high confidence

Figure 7 Modelica output of voltage across capacitor

Figure 7 is an example of a typical output one might use. In this case, the voltage across the capacitor represents the venous signal.

Conclusions

Electrical analogs are convenient way to represent physiological systems that depend primarily on signal-in/signal-out dynamics. Since all other areas of complexity, such as chemical interactions, mechanical influences, etc. are ignored, it produces a model that is easily digested by a simulator. This WebGME project produces a DSML capable of creating these models. It is the author’s intent to continue adapting this DSML for more complicated physiological analogs.

References

1. Zhihao Jiang and Rahul Mangharam. 2015. High-Confidence Medical Device Software Development. Found. Trends Electron. Des. Autom. 9, 4 (December 2015), 309-391.
2. Patel, Darshan. "A Real-Time Technique for the Correction of Invasive Blood Pressure Measurements using Counter Pressure." Electronic Thesis or Dissertation. University of Akron, 2008.
3. *WebGME Tutorial*. ISIS/Vanderbilt University, 2017, https://webgme.readthedocs.io/en/latest/index.html. Accessed 14 December 2017