# Digital Mockups for the Testing of a Medical Ventilator

Bailey Miller
Computer Science & Engineering
University of California, Riverside
bmiller@cs.ucr.edu

Frank Vahid
Computer Science and Engineering
University of California, Riverside
vahid@cs.ucr.edu
Also with CECS at UC Irvine

Tony Givargis
Center for Embedded Computer Systems
University of California, Irvine
givargis@uci.edu

## ABSTRACT
Medical devices have become more difficult to test as hardware and software complexity grows. Device manufacturers must meet minimum standards while striving to reduce product development time. New techniques for providing comprehensive testing need to be developed that can be easily configured, cover a broad range of test scenarios, and facilitate automation. *Digital mockups* describe a method of testing a cyber-physical device wherein a digital model of the environment is used to stress a real device's embedded software and functionality. We demonstrate the use of a digital mockup to test a medical ventilator device. A lung model is hosted on an FPGA and connected to a ventilator by bypassing the ventilator's transducers. PC-based manager software allows user configuration in real-time of both the device and the model to facilitate test automation. An XML model description is embedded in the digital mockup framework to facilitate communication between the FPGA and PC software.

## Categories and Subject Descriptors
C.0 [**Computer Systems Organization**]: General – *Hardware/software interfaces, System architectures.*
J.3 [**Computer Applications**]: Health and Medical Sciences – *Health.*

## General Terms
Design, Reliability, Verification

## Keywords
Digital Mockups, Test Automation, System Design, Cyber-Physical Systems, Medical Cyber-Physical Systems, Mockup Electronic Data Sheets, MEDS, Hardware-in-the-Loop

## 1. Introduction
Cost and performance improvements in computing technology have been utilized by medical device manufacturers to produce increasingly more effective devices. For example, new medical ventilators often consist of multiple heterogeneous processors, complex transducers, and touch-screen user-interfaces. The growing computing complexity of medical devices comes with new problems for developers; an expanding software code base and the inclusion of more transducers into designs means that more time is spent during development and testing. Testing guidelines may often detail hundreds of requirements against which new medical devices must be verified. Reducing the time spent to perform necessary testing is a crucial goal of any developer.

A common testing method for devices interacting with physical systems, or cyber-physical systems, uses physical mockups, wherein a device is connected to a mechanical emulator of the normal physical environment. For example, a balloon consisting of some elasticity and flow resistance can be used to emulate a human lung. Physical mockups provide a simple mechanism for testing a system, but lack the sophistication to capture more complex physical system phenomena. A more thorough method of testing uses digital mockups [11][15], in which a mathematical model of the environment interacts directly with the device's computers, bypassing transducers, to stress device functionality. A digital mockup benefit is the flexibility to capture a broad range of environmental scenarios not always possible when using a physical mockup. For example, closed-loop mechanical ventilation that is coupled with oximeter readings to monitor blood oxygen levels would require a complex and perhaps unbuildable system of physical mockups to emulate respiratory mechanics and complete cardiopulmonary activity. In contrast, a digital mockup could utilize mathematical models of the physiological systems, of which many exist in the literature or on the web [12], to more simply produce the same system with greater flexibility. Another digital mockup benefit regards automation. Physical mockups usually require human involvement during testing to configure the mockup, to observe/capture data or waveform output, and to determine if the test was successful. Digital mockups enable full test automation because they can be configured automatically by outside software to adjust for variable environmental conditions. Drawbacks of digital mockups using bypassed transducers may include lack of testing of those transducers, and challenges in achieving accurate and real-time models of complex physical environments.

We present a demonstration of our digital mockup technology. A ventilator is connected to a digital mockup of human lungs. The ventilator operates as normal despite having its transducers bypassed such that all physical phenomena are translated completely to the digital domain. PC-based software is used to configure the model and ventilator settings, watch and record data streams, and facilitate test automation. An embedded XML file within the digital mockup describes the capabilities of the model to the PC software.


*IHI'12,* January 28–30, 2012, Miami, Florida, USA.

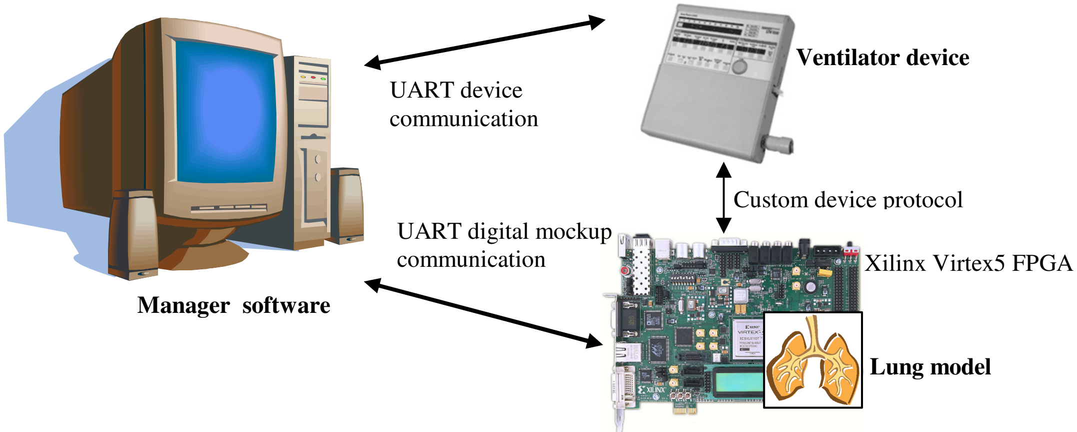


**Figure 1:** Configuration consisting of PC-based testing software, digital mockup of a lung, and ventilator device. The ventilator's transducers are bypassed; the ventilator communicates directly to the mockup through a shared bus protocol. PC-based manager software communicates through serial interfaces to the digital mockup and ventilator to configure settings or stream data and to run tests.

## 2. Related Work

In previous works, digital mockups have been used to effectively test various designs. Pimentel [13] built a heart simulation on an FPGA which uses digital-analog converters to communicate with a pacemaker device. The Penn State Virtual Heart Model (VHM) involved an electrocardiographic model in order to test various pacemaker algorithms [5]. Sirowy [15] introduced the transducer bypass method to facilitate communication between device and model. Miller presented a framework for building codesigned hardware/software digital mockup architectures [11].

Physiological system modeling can be used to create better-quality medical cyber-physical systems. Lee [7] described the need for the accurate modeling of patients to deliver higher-confidence devices. Arney [1] used a patient model consisting of drug absorption levels and patient vitals to test a closed-loop system of heterogeneous medical devices which includes a PCA pump and pulse oximeter. Lee presented a closed-loop artificial pancreas design [7], which is validated through the use of a human diabetic subject simulator [6]. Lee suggested that comprehensive models are too complex to be used for design purposes, however digital mockups can deliver such high-fidelity patient models in real-time [9].

## 3. System Components

The demonstration consists of three main components: The ventilator under test, the digital mockup platform, and PC-based manager software as shown in Figure 1. The objective of the system is to stress the ventilator's software control-algorithms by interfacing to a lung model on the digital mockup platform. PC-based manager software provides a user-interface that supports manual configuration of ventilator and lung-model settings, monitoring of model or device outputs, and the provision of mechanisms for test automation.

### 3.1 Ventilator

The ventilator described in this demonstration and accompanying video, available online from www.cs.ucr.edu/~bmiller/digital_mockups_IHI_2012.mp4 and http://www.youtube.com/watch?v=ThUKVhqoA3Q, was obtained from a collaborating medical device manufacturer. The ventilator is a complex heterogeneous computing platform that utilizes a wide variety of sensors and actuators to deliver positive pressure ventilation to the patient airway. A touch-screen interface allows the user to configure particular settings such as delivered volume or breathe rate. An embedded processor utilizes sensor readings to actuate a variable speed fan in order to deliver appropriate air flow to the patient.

Due to agreed upon restrictions regarding public displays of the ventilator, for demonstration purposes, we have also developed a prototype ventilator on a Xilinx Spartan 3E FPGA platform, which emulates the pressure/flow control loop of a ventilator. A PID controller is implemented on an embedded processor, which retrieves pressure sensor values and a desired target pressure as inputs. The controller sets an output flow of the ventilator as the controller attempts to match the sensed and targeted pressures. Various other user inputs, such as breath rate and rise time, are also considered to better capture the various operating modes of the ventilator.

### 3.2 Digital Mockup

The digital mockup component is responsible for hosting the lung model which interacts with the ventilator device. The digital mockup must be able to run in real-time to successfully interact with the device under test. Physiological models may vary in complexity from tens to thousands of ordinary differential equations, and thus may not run in real-time on a standard PC. We therefore utilize an FPGA platform that can harness parallel processing techniques to compute complex models in real-time. For our current implementation, we utilize a Xilinx Virtex5 FPGA to host a circuit synthesized via the Xilinx XST tool chain [17]. We use a previously described tool [11] to generate the digital mockup framework prior to synthesis. Using an FPGA is not a requirement for the framework, as less complex models may meet real-time constraints on a desktop computer. We choose an FPGA in order to better resolve communication between ventilator and mockup, which requires a custom circuit to implement a shared bus protocol.

To achieve communication between the ventilator and the digital mockup, we utilize a transducer bypass method [15]. The fan actuator and pressure sensing modules of the ventilator have been removed from the design. The digital mockup is connected directly to an internal bus of the ventilator, that bus being responsible for transmitting commands between the ventilator's

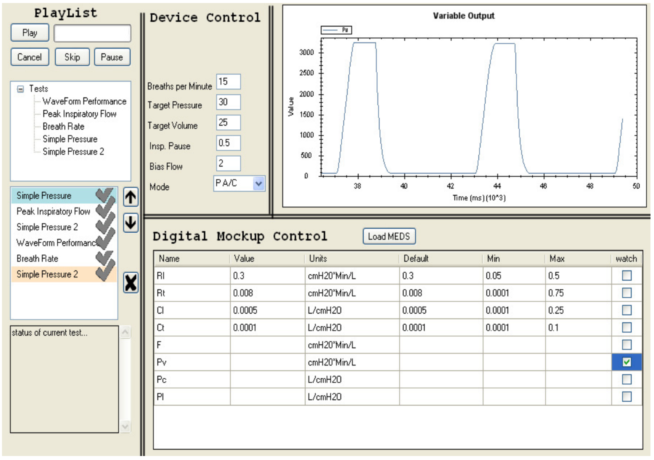


**Figure 2:** Screenshot of the PC-based manager software user interface.

main control processor and the transducers. Commands from the main control processor are intercepted by transducer models hosted on the digital mockup. The intercepted commands are translated into inputs/outputs for the lung model on the digital mockup.

## 3.3 PC-Based Manager Software

The front end of the system is the PC-based manager software, which has three objectives. The first is to provide an interface that allows users to manually or automatically configure model parameters or ventilator settings. A soft serial interface has been included by the manufacturer on the ventilator to provide for the remote alteration of device settings instead of the normal settings via the devices touch screen and knobs. The digital mockup similarly provides an interface for the alteration of model parameters through a MEDS module, an embedded XML data sheet which describes mockup capabilities [11]. The second objective is to provide the ability to watch data streams either of the ventilator or of internal lung model values. Watching real-time streaming data provides testers with the ability to effectively evaluate the current operating status of the device and model. The same interface used to access the MEDS content of the digital mockup is also used to request and stream internal lung values to the software for display to the user. The third objective is the ability to automate test procedures – potentially providing substantial time savings and cost reductions to device manufacturers.

The software is built such that the same application can be used to support multiple models or devices. The lung model of the digital mockup can be swapped for more complex or differently behaving models through the MEDS module, and thus the user-interface is necessarily separated from the underlying model of the digital mockup. Likewise new device interfaces may be included in the software and arbitrarily chosen from at run time if desired. Our software was written in Microsoft Visual Studio 2008, using C# as the primary language [3], with usage of the Python programming language [14] to define dynamically available test procedures.

# 4. Demonstration

The demonstration provides the audience with the ability to interactively configure both the ventilator and lung model settings in real-time through the software hosted by a PC, whose interface

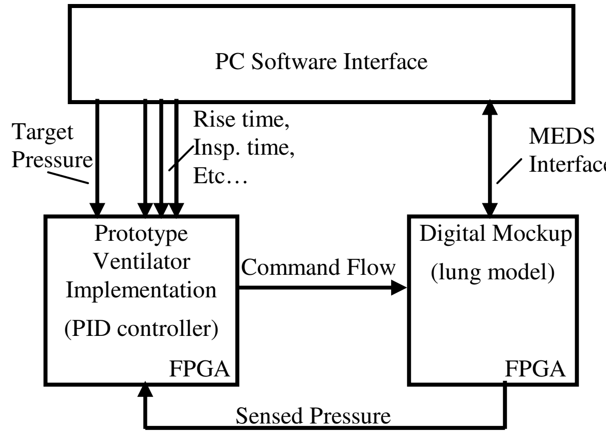


**Figure 3:** Control loops in demonstration

is shown in Figure 2. Example tests are also provided that can be used to build a sample test suite for automated testing.

## 4.1 Ventilator Configuration

The ventilator provides the ability to configure settings for the following parameters:

- Target pressure
- Breath rate
- Rise time
- Inspiratory time

The target pressure setting refers to the desired pressure of the patient airway during each inspiration. The goal of the embedded controller of the ventilator is to achieve this target pressure during inspiration. The breath rate controls the number of breathes per minute that should be delivered to a patient. Rise time controls the speed with which the controller should attempt to match the sensed pressure to target pressure. A small rise time implies that the slope of the plant output waveform will be large. Inspiratory time is the time of inspiration for a single breath.

## 4.2 Model Configuration

The lung model hosted by the digital mockup consists of the RRCC model described by Borello [2]. Using the previously described MEDS concept, the RRCC model information is loaded dynamically into the software and presented to the user. The various parameters and state variables described by the MEDS component are loaded into the table shown at the bottom of Figure 2. The RRCC model consists of the following configurable parameters and state variables:

- Resistances: $R_{Tube}$, $R_{Lung}$
- Compliances: $C_{Tube}$, $C_{Lung}$
- Pressures: $P_{Ventilator}$, $P_{Ccircuit}$, $P_{Lung}$
- Flow: Q

Resistances and compliances in the model can be altered by users directly in the software interface to induce changes in the behavior of the model. Ventilators must be tested over a wide variety of possible lung conditions, and thus the configurable parameters of R and C provide simple levers for altering healthy lung models into different states of non-healthy models, such as an asthmatic lung. $P_{Ventilator}$ represents the pressure at the ventilator source. $P_{Circuit}$ represents the pressure at the patient mouth. Q represents the flow of air through the patient circuit, originating from the ventilator output. The pressures and flow state variables are internal lung values which constitute the output of the model. State variables may not be written to, but may be streamed from the digital mockup for reading.

The RC model presented here is a simple model, not necessarily requiring FPGA acceleration or capturing lung behavior as accurately as possible. Current work is focusing on developing more complex models, such as a multi-generation Weibel lung model [16], to better represent lung behavior. The modularized design of the digital mockup architecture allows new models to be swapped in quickly, without requiring a complete refactoring.

### 4.3 Test Procedures

Full automation in our system allows testers the ability to run example test suites without requiring humans in the loop. The software provides for the automation by reading in a user-selected file that describes the locations of various test procedures on the local file system targeted to the device under test. The tests can then be individually selected and placed into a "playlist" of tests to be executed. The test suite once activated will commence and execute each test in a sequential order once. Before activating the test suite, the communication between the software and the device/mockup is closed. The closure allows for the active python test script to open its own communication channels with the peripherals in order to send commands and receive data. Data can be optionally redirected from the active test script to the main software interface in order to display data waveforms or create logs for later review. At the end of execution of each test procedure, the script will return a Boolean that indicates the success of the test. The user is informed via green or red indicators the status of each test as it completes its execution. Test failures need not halt the entire test suite unless specified to do so, supporting the situation where each test is an independent procedure, and whose failure is not indicative of the result of the following procedures.

## 5. Conclusion

A demonstration of digital mockup technology is presented utilizing a digital mockup hosting a lung model to test a medical ventilator device. PC-based manager software can be used to configure in real-time both the digital mockup and the device, make real-time internal lung values available to the user, and to create custom test playlists that facilitates test automation.

## 6. Acknowledgements

This work was supported in part by the National Science Foundation (CNS1016792, CPS1136146), the Semiconductor Research Corporation (GRC 2143.001), and a U.S. Department of Education GAANN fellowship.

## 7. References

[1] Arney, D., Pajic, M., Goldman, J., Lee, I., Mangharam, R., and Sokolsky, O. 2010. Toward patient safety in closed-loop medical device systems. *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems* (ICCPS '10). ACM, New York, NY, USA, 139-148. DOI=http://doi.acm.org/10.1145/1795194.1795214.

[2] Borrello, M. 2005. Modeling and control of systems for critical care ventilation. *American Control Conference, 2005. Proceedings of the 2005* , vol. 3, (June 2005), 2166-2180.

[3] Heljlsber, A., Wiltamuth, S., and Golde, P. 2003. C# Language Specification. *Addison-Welsey Longman Publishing Co., Inc.,* Boston, MA, USA

[4] Huang, C., Vahid, F., and Givargis, T. A custom FPGA processor for physical model differential equation solving. *Embedded Systems Letters*. 2011.

[5] Jiang, Z., Pajic, M., Connolly, A., Dixit, S., and Mangharam, R. 2010. Real-Time Heart Model for Implantable Cardiac Device Validation and Verification. *Euromicro Conference on Real-Time Systems* , 239-248.

[6] Kovatchev, B.P., Breton, M., Dalla Man, C., and Cobelli, C. 2009. In Silico Preclinical Trials: A Proof of Concept in Closed-Loop Control of Type 1 Diabetes. *Journal of Diabetes Science and Technology*, vol.3, no.1, (Jan. 2009), 44-55.

[7] Lee, H., Buckingham, B.A., Wilson, D.M., and Bequette, B.W. 2009. A Closed-Loop Artificial Pancreas Using Model Predictive Control and a Sliding Meal Size Estimator. *Journal of Diabetes Science and Techonology,* vol.3, no. 5, (Sept. 2009), 1082-1090.

[8] Lee, I., Pappas, G.J., Cleavelan, R., Hatcliff, J., Krogh, B.H., Lee, P., Rubin, H., and Sha, L. 2006. High-confidence medical device software and systems. *Computer*, vol.39, no.4, (April 2006), 33-38. DOI=10.1109/MC.2006.127.

[9] Lee, I., and Sokolsky, O. 2010. Medical Cyber Physical Systems. *Design Automation Conference* (DAC '10), (June 2010), 743-748.

[10] Licht, T.R. 2001. The IEEE 1451.4 proposed standard. *Instrumentation & Measurement Magazine, IEEE*. vol.4, no.1, pp.12-18, (Mar. 2001). DOI= 10.1109/5289/911168.

[11] Miller, B., Vahid, F., and Givargis, T. 2011. Application-Specific Codesign Platform Generation for Digital Mockups in Cyber-Physical Systems. *Electronic System Level Synthesis Conference* (ESLsyn '11).

[12] Physiome Project. www.physiome.org.

[13] Pimentel, J.C.G., and Tirat-Gefen, Y.G. 2006. Hardware Acceleration for Real Time Simulation of Physiological Systems. *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE* , (Sept. 2006), 218-223.

[14] Python programming language. www.python.org.

[15] Sirowy, S., Givargis, T., and Vahid, F. 2009. Digitally-Bypassed Transducers: Interfacing Digital Mockups to Real-Time Medical Equipment. *Internation Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC '09).

[16] Weibel, Ewald R. Morphometry of the human lung. *Academic Press*. 1963.

[17] Xilinx. www.xilinx.com