SAND2014-20142C

RAPID PROTOTYPING OF HARDWARE USING REAL-TIME HWIL SIMULATION ENVIRONMENTS

Jonathan F. Torres* **and Scott A. Kowalchuk†**

Sandia National Laboratories, Albuquerque, NM

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The Navigation, Guidance, and Control (NGC) Department at Sandia National Laboratories conducts flight test programs where rapid prototyping is essential. To successfully maintain schedule it is critical to have high confidence in the NGC hardware and software prior to integration testing. The NGC Department has developed a V-Model diagram approach to ensure high confidence with hardware and software prior to integration testing within a rapid prototyping environment. The V-Model detailed design process flow describes a design approach for testing hardware and software early and often using hardware-in-the-loop (HWIL) and software-in-the-loop (SWIL) simulations.

INTRODUCTION

The traditional V-model design diagram, shown in Figure 1, was a design process developed in Germany for government projects[1] that has been widely adopted. The V-model design process is a flowchart that is designed to be followed from the left of the V to the right of the V. It is designed based on the four phases of product development:

- system requirements
- modeling
- target implementation
- validation

The objective of the V-Model is to catch design problems early in the design phase and fix those problems in order to reduce the amount of time needed to develop the final product. This helps drive production cost to a minimum. Referring to Figure 1 of the V-Model, the objective is to pinpoint most errors in the design phase and remedy them before advancing to system testing [2][3]. At the design stage is where the component testing can be performed. Instead of advancing straight to system testing, simulation can be a valuable asset in ensuring that the hardware satisfies requirements.

*Member of Technical Staff, Sandia National Laboratories, P.O. Box 5800, MS 1174, Albuquerque, NM 87185-1174; jtorre4@sanida.gov.

†Principal Member of Technical Staff, Sandia National Laboratories, P.O. Box 5800, MS 1174, Albuquerque, NM 87185-1174; sakowal@sandia.gov.

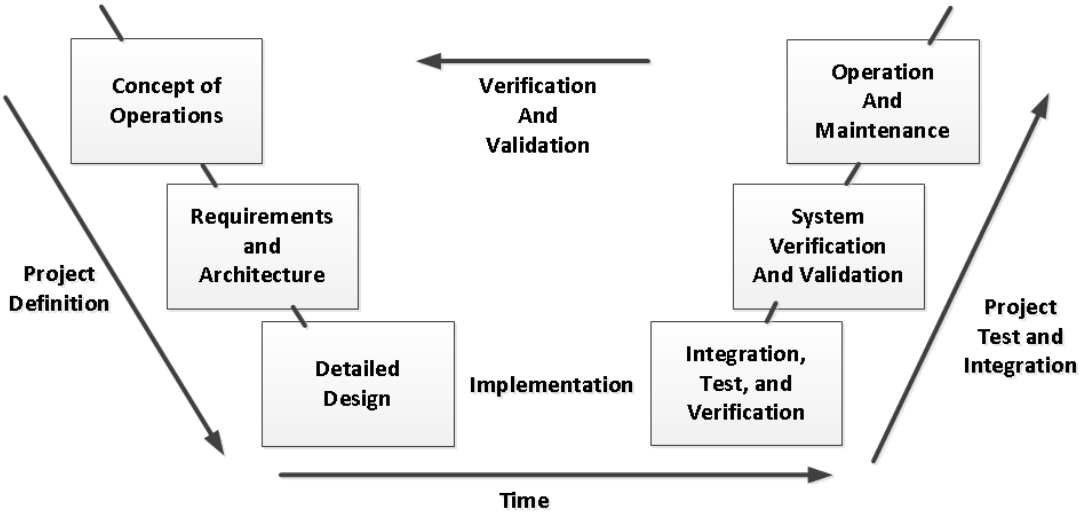

Figure 1: V-Model Design Process Diagram

V-MODEL DETAILED DESIGN PROCESS FLOW

Figure 2 shows a V-Model that expands on the design phase of the traditional V-Model in order to reduce design time and system verification. This model may consume more time in the design phase than a traditional V-model but provides the engineer with confidence that the final design is accurate and the potential for redesign is decreased. If the mission is critical and on a tight schedule, it is crucial to minimize the amount of necessary redesigns in order to attain the desired goal within the project deadline.

Utilizing the traditional V-Model in Figure 1, if the engineer is first testing the final widget at the integration, test, and verification stage of the V-Model, there is more potential for product redesigns when testing to make sure every requirement of the final product is satisfied. Suppose a mission is to be completed within 24 months and the engineer finishes the design of the product within 10 months. The V-Model proposed in this paper significantly minimizes risks and improves confidence of redesign work for each component that needs to be integrated into the system. The proposed V-model, shown in Figure 2, depicts how the NGC Department develops their desired hardware.

As can be seen in Figure 2, the design process is significantly more involved. The HWIL simulation will start to be developed at a much earlier stage – in the design stage as opposed to the system verification stage. Even though it is necessary to model the whole system in a HWIL simulation at the system verification stage, developing the HWIL simulation at the design stage allows the engineer to test the final product at the component level instead of testing every aspect of the final product. Table 1 outlines how the progression of each design phase is related to the HWIL and system requirements.

Assuming the concept has been developed and the requirements have been defined in the V-Model, the objective of building the HWIL system at the design stage is to verify that individual requirements are operating as expected. In order to test that this is occurring, it is not necessary that the final product be developed. The engineer can start with Commercial-Off-The-Shelf (COTS) hardware or some piece of hardware that is in their inventory that performs the same input/output (IO) function as the desired final product. At this stage of design, the engineer is only testing that the IO is functioning properly. All software algorithms should not be analyzed for accuracy at this stage of design. All models can be developed and cables defined in order for the piece of hardware to communicate with the RTOS computer. Even if the COTS hardware does not have identical IO as the final widget, it can be documented the IO functions that were tested using the COTS hardware

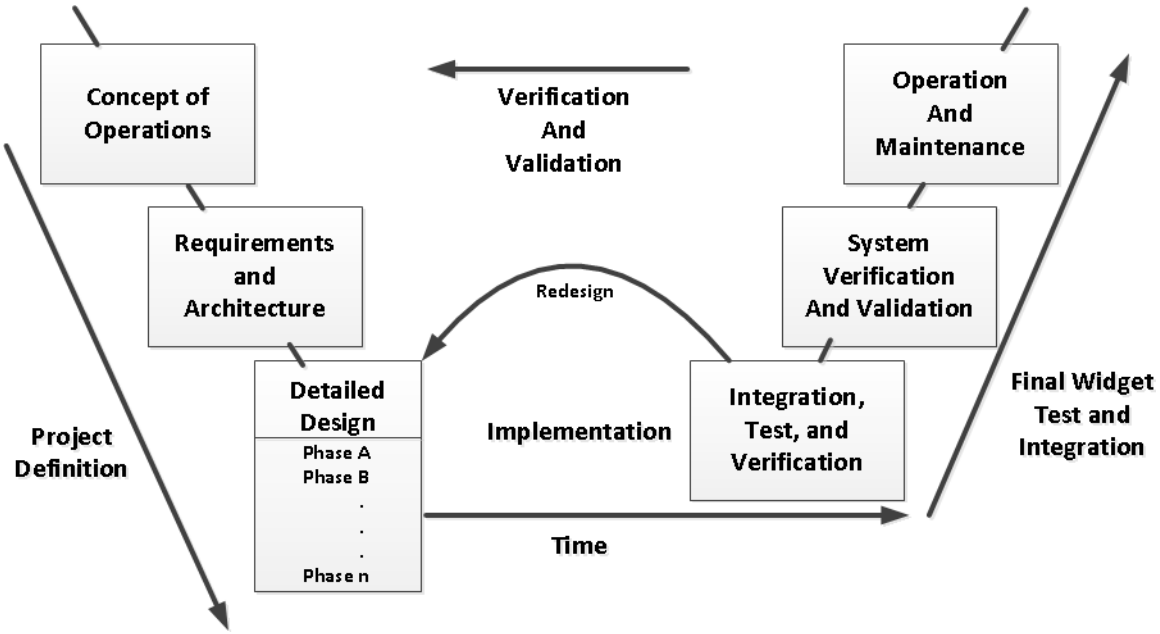

Figure 2: V-Model Design Process Diagram for NGC Hardware Development

Table 1: Design Phase Testing

Design Phase	Objective of Phase	Hardware/ Software	Real-Time HWIL	Pre-Verification Requirement
A	Test IO Function	COTS	IO Test	x
B	Confidence to satisfy req. y	Prototype 1	Detail to satisfy req. y	y
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
n	Confidence to satisfy req. z	Prototype n	Detail to satisfy req. z	z

during this phase of the design.

As revisions of the final product are released, the engineer can utilize the prototypes to check if the requirements are fulfilled. If an operation is not working as expected, the engineer is able to catch the mistake early in the design process and correct it before the final widget is released. The more requirements that are able to be satisfied with the prototype builds, the less risk of an error occurring when the final widget is built. As prototype builds are released, the engineer is not only able to verify that the hardware is operating correctly, but they are also able to verify that software algorithms are operating correctly also. Therefore, each prototype that is released is not required to be a hardware modification; the modifications made could enable the engineer the ability to verify several software algorithm requirements also.

Figure 3 is a basic flowchart that describes how HWIL testing, COTS, and prototype hardware aids the engineer in a faster design process. The engineer will start in the initial phase and develop the necessary HWIL simulation in order to test that the objective requirement of the initial phase is satisfied. If it is satisfied the engineer will progress to Phase B. Another revision of hardware or software may need to be released in order to test the objective requirements for Phase B design. During the Phase B design, the engineer will not only need to ensure that the requirements for Phase B are satisfied but also that the requirements for Phase A are satisfied using the prototype

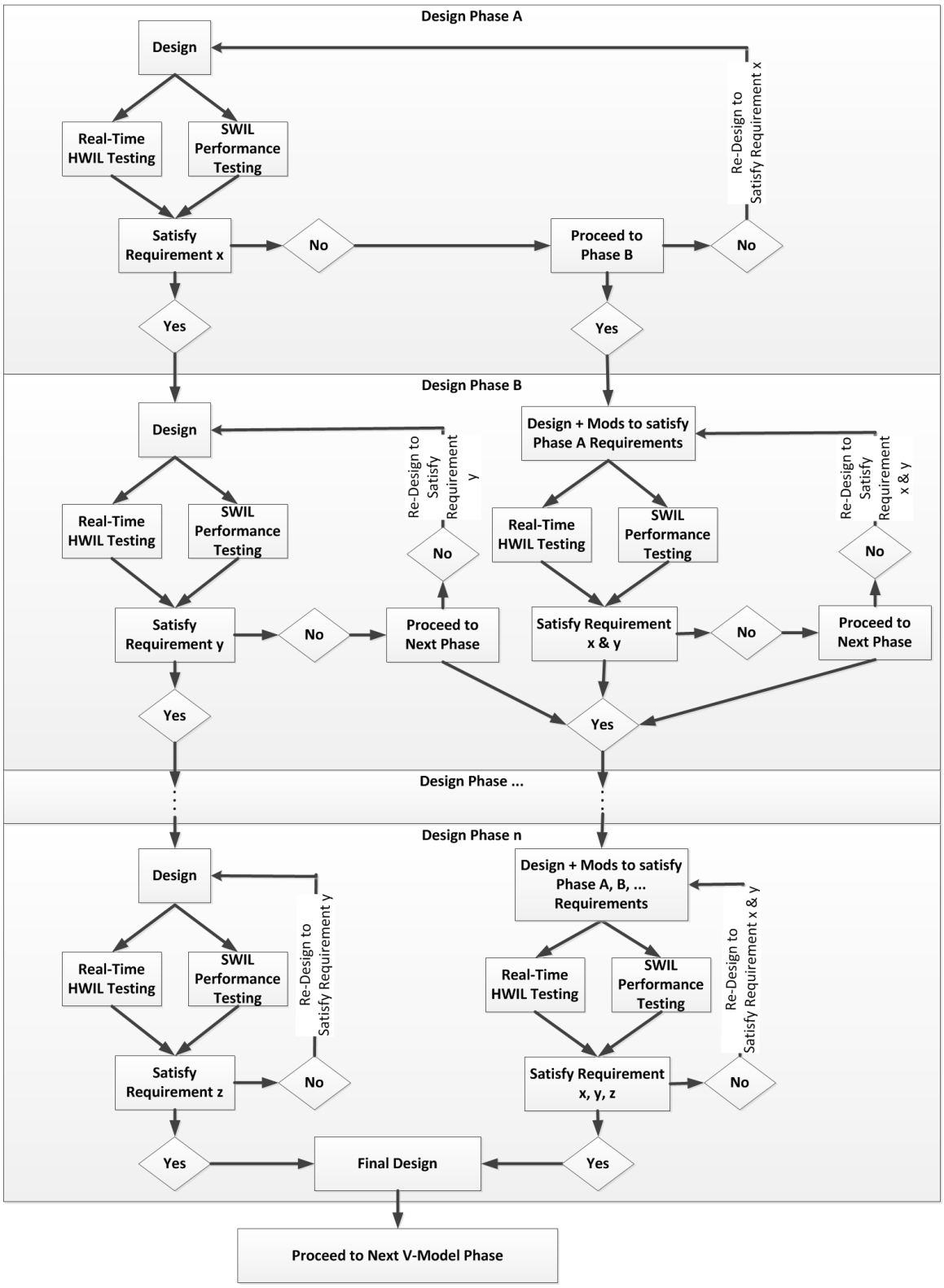

Figure 3: Flowchart of prototype HWIL Development

from Phase B. The engineer will progress through different hardware/software requirements until all requirements have been tested and the hardware/software is ready for a final release. If the engineer encounters a problem with one of the design phases, they will have to decide whether they want to troubleshoot the objective requirement during the current design phase or progress to the next design phase and test the previous phase's design requirement along with the current design phase requirements. During the early stages of development, specifically when using COTS hardware, the engineer cannot expect 100% accuracy when attempting to fulfill the requirements. In order to advance the design process, the engineer should enforce that some percent of the design requirement pass during early design phases, such as stipulating 90%. As more design phases are complete, the engineer can improve their desired requirement accuracy for every HWIL simulation component test.

Building the HWIL simulation along with HWIL development also aids the testing process of the final widget. If the HWIL simulation is constructed as the final widget is developed, less work will need to be invested into testing the final widget. All component models will have been constructed, and the engineer can then leverage the component tests to tests the final product. There should only be a small time commitment to ensure that the final product is ready for system testing from an HWIL perspective. In order to utilize component testing for final product verification, all component tests must be organized and easily reusable. The NGC Department has its own protocol for easier usability of all component tests.

COMPONENT TEST

The Navigation, Guidance, and Control Department at Sandia National Laboratories has a multi-phase process for developing hardware. Prototype hardware is first developed and then transitioned into hardware. In order to test each stage of the hardware, the hardware will be incorporated into an HWIL simulation. The HWIL simulation will model the dynamics of the system in a real-time environment incorporating the hardware into the simulation to examine if the hardware is operating in the desired manner. The SWIL simulation also models the dynamic of the system, however, the SWIL is able to execute the simulation faster than real-time and supports robustness of the software through analysis with methods such as Monte Carlo. The SWIL also provides a third environment to verify mission requirements. The NGC Department runs HWIL simulations by using Mathworks' software. Simulink is used to design the model and Mathworks' Simulink Real-Time Workshop will deploy the Simulink model on a target PC.

In order to allow rapid prototyping of hardware, test models will be developed for each piece of hardware that will be integrated into the HWIL simulation. These models will be developed regardless of whether the piece of hardware is a prototype for a piece of hardware or is specific to the HWIL simulation. This is beneficial for several reasons. Test models are necessary because it is much easier to debug one piece of hardware as opposed to several pieces of hardware at the same time. If a test model is developed for a single piece of hardware, one will have a knowledge of the functionality of that piece of hardware before interacting it with another piece of hardware. This makes the process of interfacing multiple pieces of hardware much more seamless.

Developing test models for specific pieces of hardware is also valuable from a safety perspective. The test model that will be developed for the prototype hardware allows the engineer to more familiarize themselves with the piece of equipment. After the test model is constructed, the engineer will understand the functionality of the hardware and know what to expect when integrating it with other hardware. This is valuable because if two pieces of hardware are integrated with each other without any common knowledge there is risk of damaging the equipment.

Component tests also makes the debug process easier because the user always has a test model to refer to as the prototype hardware iterates through several revisions. If a test model is associated with each revision, one always has a basis to refer to before another hardware revision is made. Therefore, if a hardware revision is made and the hardware is not operating successfully in the overall HWIL simulation, the user could test the performance of the individual piece of hardware

in question by referring back to the test model. This makes it easier to debug the overall HWIL simulation.

Component tests are not necessarily specific to an individual piece of hardware. Once individual hardware component tests are designed, component tests that are comprised of interfacing several hardware components could be designed. This allows the engineer to characterize sections of the entire system and test these systems independently. Each HWIL component will now be run at a subsystem level. This also enables the engineer to debug the system much easier. If there is a problem with the HWIL system, the engineer only now has to pinpoint the problem down to the subsystem level instead of the individual piece of hardware that is the root of the problem. Component tests are not only valuable with troubleshooting hardware; they also can be used to verify software also.

Since it is vital that the same result is attained from an SWIL or HWIL simulation, the engineer can also perform component tests at a software level also. An SWIL simulation utilizes the same dynamics of the system, however, it is run in desktop mode. Essentially, one subsystem of the model will dictate whether the model is to be executed in HWIL or SWIL mode. When the model is set to run in HWIL mode, hardware will execute its intended function providing input or output to the RTOS. During an SWIL simulation, all processing will be done on the PC running Mathworks. The hardware that is present in the HWIL system will be emulated to execute the same functionality as when ran in the HWIL simulation. The outputs of the SWIL will then be compared against the HWIL to gauge the fidelity of the emulated hardware in SWIL mode. Comparing the SWIL results to the HWIL should be performed at a component and system level.

DEVELOPMENT OF HARDWARE

It is not necessary to wait until the final hardware is produced in order to integrate the piece of hardware into the HWIL simulation or testing of preliminary evaluation of requirements. The hardware should be characterized into specific functions that it must perform. As these functions are developed, the HWIL simulation can be developed simultaneous to the hardware design process. Even though the hardware is not available to test, COTS hardware can be purchased that (may) perform a similar operation that the hardware is being designed for. The COTS hardware can then be integrated into the HWIL simulation until the hardware prototype is ready for testing.

The Simulink development that is necessary for the hardware to communicate with Mathworks' RTOS can be developed using the COTS hardware. Once the prototype hardware is ready for testing the prototype hardware can then be substituted for the COTS hardware into the HWIL simulation. The HWIL model for the COTS hardware can then be leveraged to implement any necessary changes for the prototype hardware to be integrated into the HWIL simulation.

Initial development using the COTS hardware is advantageous for a few reasons. First, the engineer does not need to start development of the prototype hardware without any foundation. Utilizing the COTS hardware to develop a Simulink test model allows the engineer to familiarize themselves with the desired functionality of the hardware. They also do not need to invest as much time designing a test model for the prototype hardware since a model has already been developed that can serve as a basis. The only necessary changes that will need to be implemented are the differences between the COTS hardware and the prototype hardware.

Once each piece of hardware is developed, the data generated from the test models will then be compared against the data generated from the SWIL simulation. This analysis will prove if the component test is suitable for the piece of hardware. It will need to be determined if the SWIL simulation needs to be changed to align with the HWIL data or if the HWIL test model will need to be tuned so it reflects what the SWIL system is outputting. This is a crucial reason why it is necessary to develop the HWIL simulation in parallel with the SWIL simulation. It is essential that the SWIL and HWIL data align throughout the development process. Following this pattern

throughout the whole design process will ensure high confidence in the hardware/software design, SWIL, and HWIL.

DEVELOPMENT OF COMPONENT TESTS FOR DETAILED DESIGN PHASES

Sandia National Laboratories' NGC Department has found that an effective design of component tests requires revision control, Mathworks' Simulink Project, and a structured model-based design flow as illustrated in Figure 4.

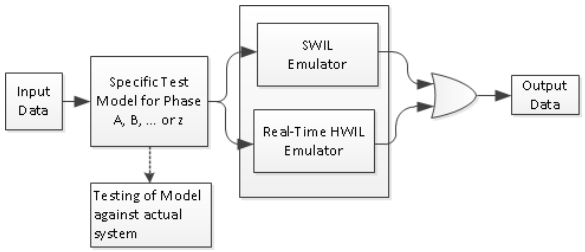

Figure 4: Common Engineering Development

All blocks contained in the component tests that are specific to the hardware should be contained in a Simulink Library. Each test model should be run with a common input data set so the engineer knows what to expect as an output. This will be beneficial when a new revision of hardware is released (Phase A, B, ...) and the engineer is running the hardware with the test model. It will already be known what is expected of the test model. As each test model is complete, revision control can be utilized to tag component test models so the engineer can revert back to different component tests for a specific design phase.

A report should be committed for every revision of the test model. The NGC Department uses Simulink Report Generator to auto-generate documentation about each test model. Using report generator, the engineer is able to record the input data, take snapshots of figures, and record any model output information in a PDF document. This is important when an engineer wants to check their test model output against previous revisions of the test model output. All the information is contained in one document.

Analysis scripts should be designed for every test model. This guarantees that all engineers will be looking at the same set of output data for each component test. As the prototype progresses, the analysis scripts will evolve to support any additional analysis needed for each component test. The report generator records the analysis results in a PDF format.

Once a piece of hardware is developed additional testing will be necessary to examine the fidelity of the HWIL simulation. In order to ensure confidence in the hardware, system testing needs to be performed and its accuracy needs be measured against the real-time HWIL system. System testing is invaluable when attempting to improve the accuracy of an HWIL simulation. Figure 4 shows a possible strategy for integrating testing at the component level. As the widget is further developed, they can be integrated into the system and subsystems can be tested to ensure that they are matching the outputs of the HWIL system. This allows the engineer to follow the same process as when a widget is between a design and final widget phase in the V-Model detailed design phase. However, when integrating several widgets into system testing, prototypes will act like subsystems and the final widget will be all widgets integrated into the system. Several iterations of system testing can allow the engineer to fine-tune the SWIL/HWIL model so that it is as close to the actual system output as possible.

CONCLUSION

The V-Model detailed design process flow described in this document allows an engineer to minimize risks when developing hardware and software. The V-Model detailed design phases allow the engineer to test the mission's requirements via HWIL and SWIL testing while the hardware is being developed. Therefore, mission requirements will be verified to a define confidence level during the design phase using COTS hardware and/or prototype hardware/software. This process requires the engineer to design component tests for each requirement which will aid in verifying requirements for the final widget design and minimizing redesign during later phases of the V-Model. After the final product is built, the engineer can then incorporate the final widget into the HWIL system to verify proper behavior. This provides more confidence and reduces risk during the system integration testing phase. System testing will then be performed to ensure the system meets mission requirements and the output data will be compared against SWIL and HWIL data in order to measure the fidelity of each simulation environment. The V-Model Diagram approach provided in this document has proven to support rapid prototyping and provide high confidence that the hardware and software will satisfy mission requirements prior to final integration testing.

REFERENCES

[1] Yang, Xia, Kirby, B., Zhao, Qiang, et al., "Model-based design process for product development of substation IEDs", Energy Conference and Exhibition (ENERGYCON), 2012 IEEE International, Sept 2012, (pp. 968–974).

[2] Vandenberghe, Patrick, "Design, Monitoring, and Operation of Adaptive Networked Embedded Systems", `http://wiki.demanes.eu/index.php/3.1`.

[3] Fosberg, Kevin and Mooz, Harold, "System Engineering for Faster, Cheaper, Better", Center of Systems Management, 1998, pp. 653–658.