ANSYS

White Paper

An Open Solution for Systems and Embedded Software Simulation

Bernard Dion, Thierry Le Sergent, Todd McDevitt
ANSYS, Inc. and Esterel Technologies

This paper describes a complete Systems and Embedded Software Validation & Verification workflow that uses SCADE® for modeling and simulating the controller software and a combination of other modeling tools for plant simulation.

It introduces the Scade language and the SCADE Suite® toolset that includes a (certified) code generator for producing the controller software. It then describes how SCADE Suite has been connected with various tools for plant simulation, including MathWorks Simulink®, ANSYS® Simplorer®, Modelica®, and LMS AMESim®.

The final part of the paper brings it all together by introducing SCADE System®, a new SysML-based tool, to describe functional and architectural views of the system including all of its interacting components, thus providing a way to assemble a complete systems engineering tool chain.

Let us start by introducing the Scade language [1] which addresses software development for critical applications. Scade has been formally defined by Esterel Technologies in conjunction with industry leaders from the aeronautics, rail and nuclear domains, as well as their respective certification authorities.

The Scade language is deterministic, and the behavior of a SCADE model is completely defined by the Scade semantics, without any ambiguity. This is a key feature for the development of safety-related applications. More generally, it is also a feature that allows for high-fidelity simulation of the application on a host PC environment with a guarantee that the same behavior can observed in simulation on the PC and on the target platform, once the final code is embedded.

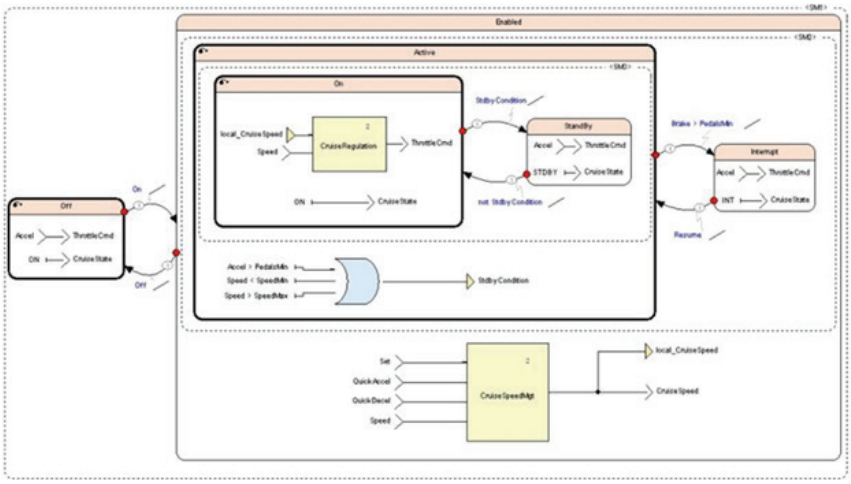

Figure 1: A Cruise Control Model in SCADE

Figure 1 provides an example of a cruise control system for a car, developed as a SCADE model. The notation is based on arbitrarily nested data flows (block diagrams) and parallel state machines. While executing such models, calculations expressed by a data flow diagram are activated only if it is positioned in a state that is currently active.

The execution model of SCADE is cycle-based. It is a direct computer implementation of the ubiquitous sampling-actuating model of control engineering. It consists of performing a con-tinuous

loop with a strict alternation between environment actions and control software actions. Once the input sensors are read, the cyclic function starts computing the cycle outputs. When the outputs are ready, or at a given time determined by a clock, the output values are fed back to the environment (actuation), and the program waits for the start of the next cycle.

The SCADE Suite model-based design environment has been built for the Scade language, and it provides an automatic code generator from a Scade model to equivalent C source code. This code generator has been certified in accordance with several industry standards, including DO-178B or C (aeronautics), EN 50128:2011 (railways), ISO 26262 (automotive), and IEC 60880 (nuclear), thereby guaranteeing that the generated source is a correct implementation of the input model.

SCADE Suite has been used to develop critical embedded software for a wide array of applications including: flight control systems for aircrafts, interlocking systems for railways, power control for electric cars, and safety systems for nuclear power plants.

The return on investment (ROI) in using SCADE is based on its certified code generator that enables a design workflow where V&V can be largely model-based and rely on host simulation rather than on late target testing. That said, effective end-to-end V&V relies on the existence of a complete systems and embedded software simulation environment on the host. We will now explore these possibilities.

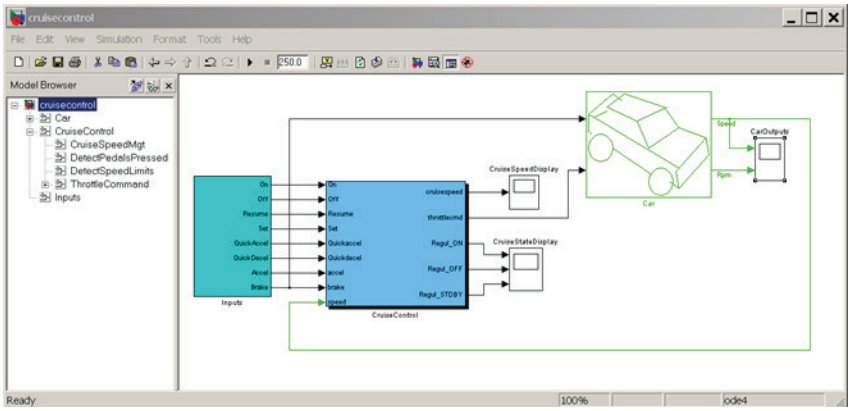

Figure 2: Cruise Control Co-Simulation with Simulink

Systems and Embedded Software Co-Simulation

In this section, we describe how SCADE has been coupled to a number of systems simulation tools in order to provide effective systems and software co-simulation at various stages in the development cycle. We will consider various connection mechanisms, including the use ad-hoc wrappers, as well as emerging standard interfaces such as FMI. This will allow modeling of the controller in SCADE while the plant model is itself described in some others appropriate modeling environments, such as Simulink, Simplorer, Modelica, and LabVIEW.

Co-Simulation using S-functions

We first describe the SCADE and Simulink co-simulation environment based on using MATLAB® S-functions.

Typically, the controller model (e.g. cruise control model) is described in SCADE, as in Figure 1, and the plant model (e.g. car model) is described in Simulink. A specific code generation target is chosen for the SCADE Suite code generator to generate the cruise control code wrapped as an S-function that can then be imported into Simulink, as shown in Figure 2.

Co-simulation between Simulink and SCADE comes in three different modes:

- **Black-box co-simulation:** direct use of an S-function for the SCADE model generated code within the Simulink plant model. This allows for simulating the SCADE model in its environment while examining the model inputs and outputs at each simulation cycle.
- **White-box co-simulation:** co-simulation between an interactive SCADE controller model simulator and the Simulink plant model simulator. This allows for simulating the SCADE model in its environment while examining the model inputs and outputs and all internal model variables at each execution of a cycle. It also allows, as with typical debuggers, to set breakpoints and stop-conditions, etc.
- **Grey-box co-simulation:** similar to black-box simulation, except that "probes" into the SCADE model are provided in the Simulink block interfaces in order to access internal information without compromising simulation performance.

Figure 3 illustrates white-box co-simulation of the cruise control model (SCADE) into the car model (Simulink).

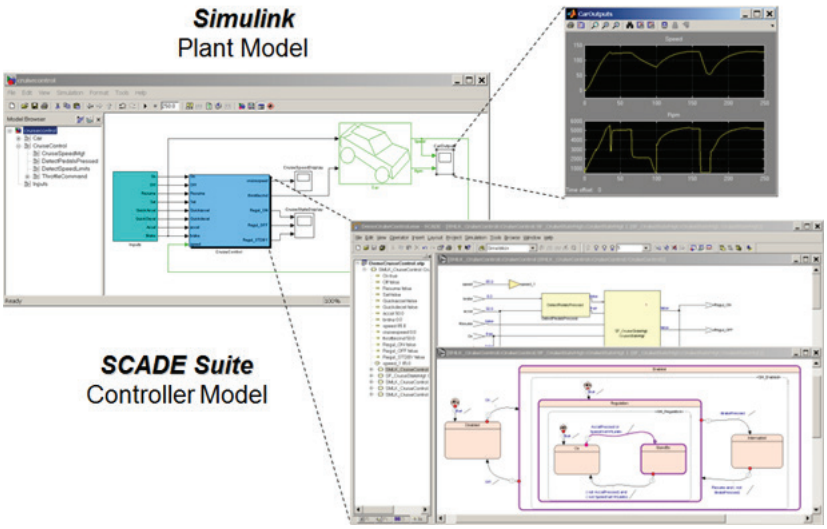

Figure 3: White-box Co-Simulation between SCADE and Simulink

Co-Simulation with ANSYS Simplorer

ANSYS Simplorer [2] is a mixed signal (analog/digital) simulation tool to design, optimize, and validate an entire system design. It is based on a number of modeling languages including VHDL-AMS, Spice, etc., and it allows for multi-domain simulation including electrical, mechanical, fluidic and thermal sub-systems.

Simplorer allows for both causal and a-causal modeling, which is different from what Simulink allows. Modeling is causal when outputs can only be described in terms of past and current inputs. This is what we have with block diagrams in Simulink. Modeling is a-causal when this restriction is lifted and a system can be described as an assembly of interacting physical components.

Another key advantage of Simplorer is that it supports multiple levels of simulation abstraction, including direct links to ANSYS 3D simulation tools and reduced order models (ROMs) (See section 3).

Simplorer modeling is illustrated by Figure 4 where a hybrid car model designed in Simplorer imports a cruise control model designed in SCADE through a dll of the generated code that is specifically wrapped for Simplorer.

Once the import performed, full black-box and white-box co-simulation between Simplorer and SCADE is available. This allows simulating the real production software of the SCADE controller with a representative physical model of the car in Simplorer, as shown in Figure 5.

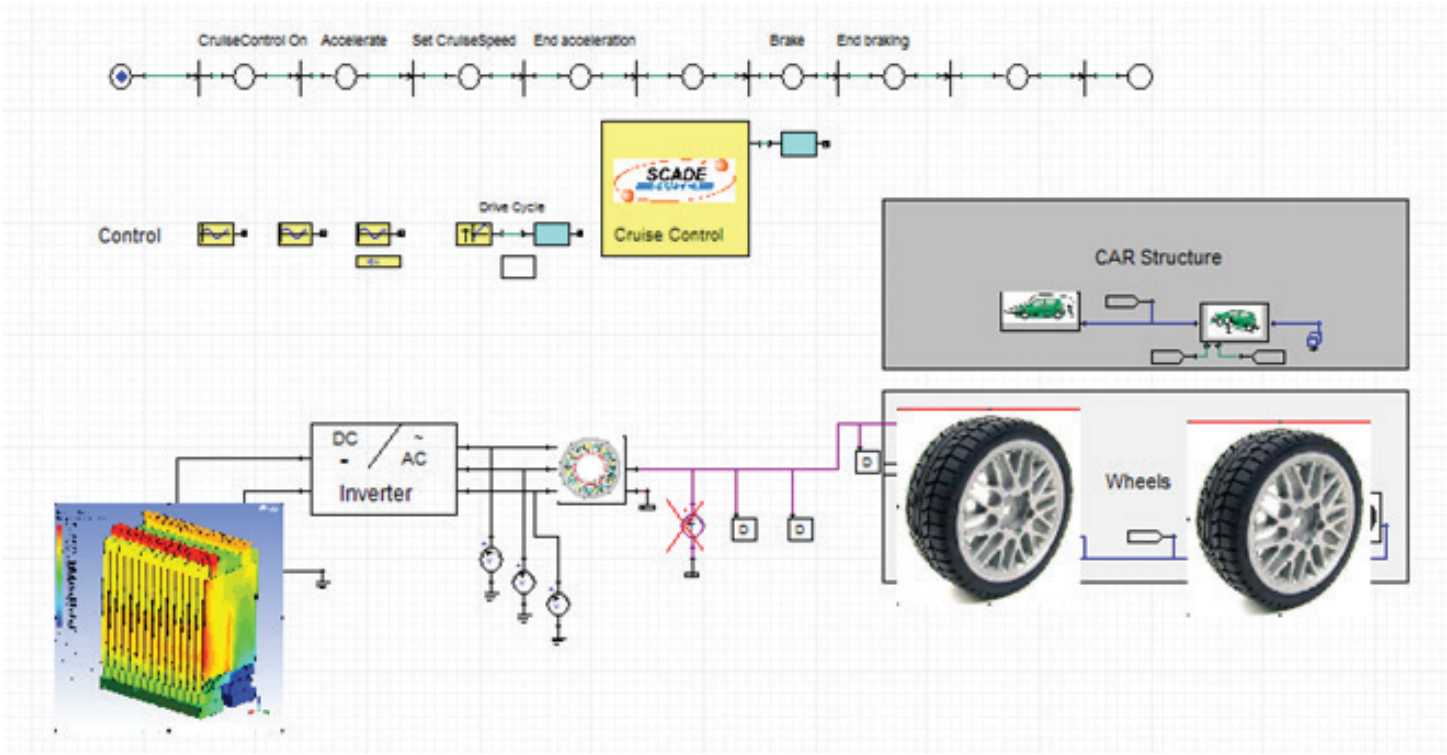

Figure 4: Co-Simulation between SCADE and Simplorer

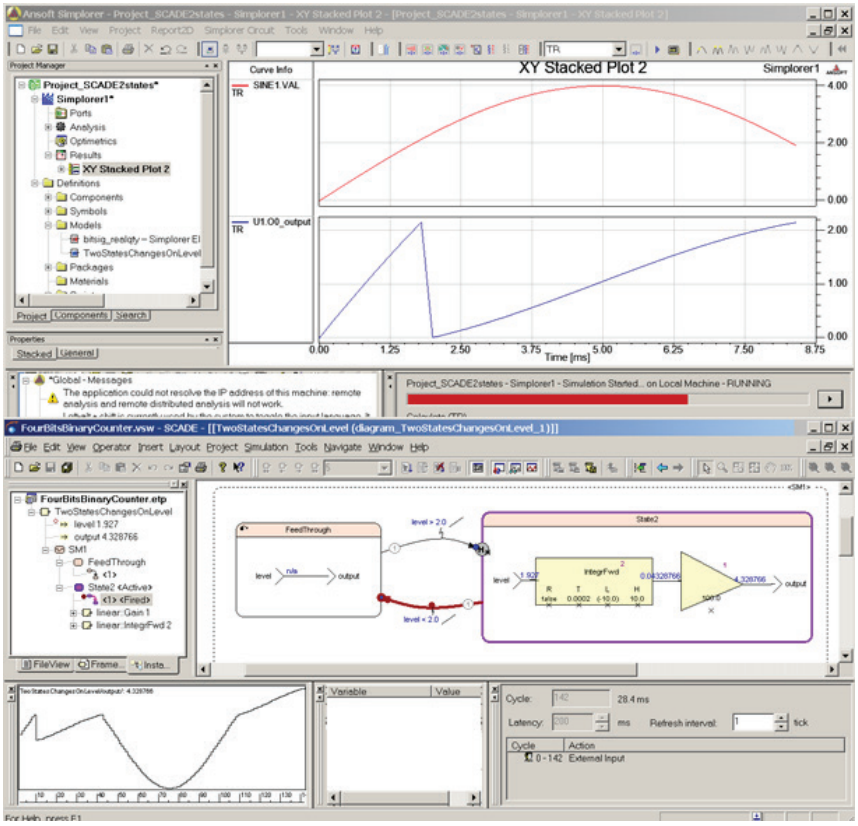

Figure 5: White-box Co-Simulation between SCADE and Simplorer

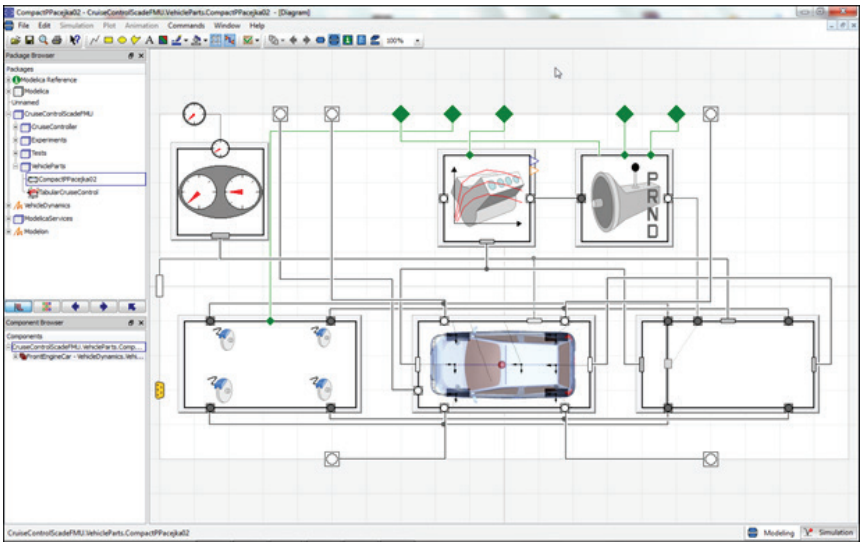

Figure 6: A Car Model in Dymola

In this section, we will introduce Modelica®. Once this is done, we will describe the FMI standard and how it can be used to establish co-simulation of a SCADE controller in a standard way with a variety of tools.

Modelica ([3], [4]) is an object-oriented, declarative, multi-domain a-causal modeling language for component-oriented modeling of complex systems. Modelica is defined by the Modelica Association.

Figure 6 illustrates a car model in Dymola, a Modelica toolset from Dassault Systèmes.

The functional mock-up interface (FMI) is managed as a sub-project of the Modelica Association, and it defines a standard interface for simulating complex heterogeneous systems. "If the real product is to be assembled from a wide range of parts interacting in complex ways, each controlled by a complex set of physical laws, then it should be possible to create a virtual product that can be assembled from a set of models that each represent a combination of parts, each a model of the physical laws as well as a model of the control systems assembled digitally. The FMI standard thus provides the means for model based development of systems and is used for example for designing functions that are driven by electronic devices inside vehicles. Activities from systems modeling, simulation, validation and test can be covered with the FMI based approach."

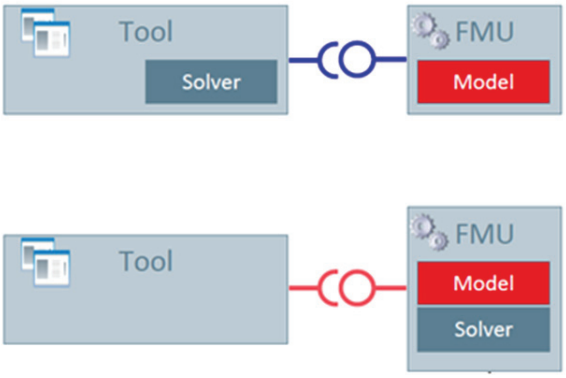

Figure 7: Model Exchange and Co-Simulation in FMI

In practice, the FMI implementation by a software modeling tool enables the creation of a simulation model that can be interconnected or the creation of a software library called FMU (Functional Mock-up Unit). As shown in Figure 7, FMI allows two modes of communication:

- FMI for model exchange
- FMI for co-simulation.

An FMI compliance checker has been developed by Modelon and is available from the Modelica Association. A number of tools are compliant, including Dassault Systèmes Dymola, LMS AMESim, National Instruments LabVIEW, and SCADE.

The FMI/FMU interface is implemented in SCADE Suite to enable co-simulation between a software controller developed in SCADE and a model of the physical environment developed in Modelica, or in other tools that implement the FMI interface. This is again shown in Figure 8Figure 8, where the cruise control model developed in SCADE co-simulates with the above car model developed in Modelica.

Similar co-simulation based on FMI/FMU has been exercised between LMS AMESim and SCADE.

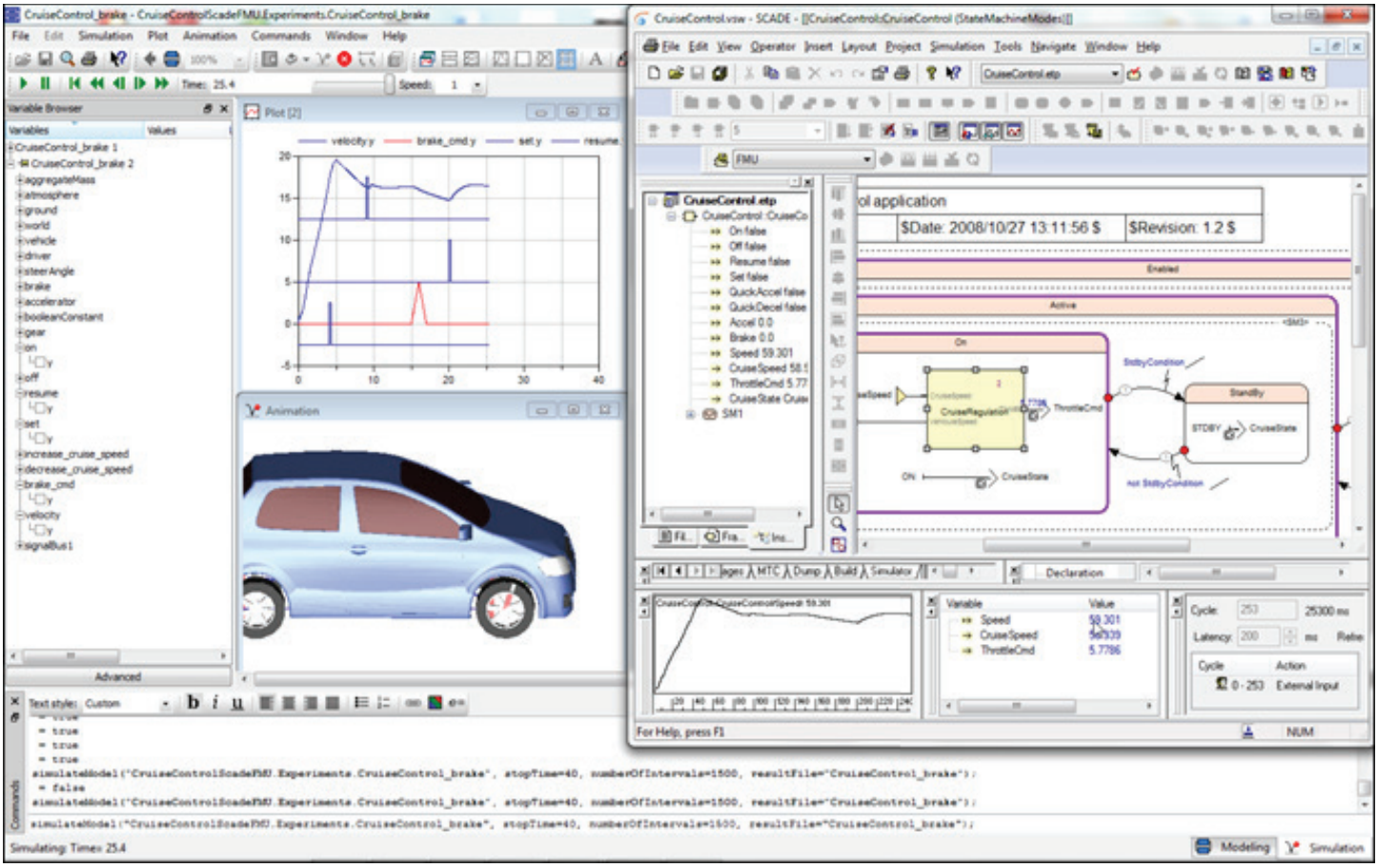

Figure 8: White-Box Co-Simulation between Dymola and SCADE trough FMI

Hardware-in-the-loop simulation with LabVIEW™

Moving from the product design phase, including Software-in-the-Loop (SiL) simulation as we have described it above, we now direct our attention to system integration and validation. To this end, Hardware-in-the-Loop (HiL) simulation is a vital step before moving to the final simulation of the controller in its real setting.

HiL simulation provides an effective platform by adding the complexity of the plant under control to the test platform including the final electronics hardware. The complexity of the plant is included in tests and development by assembling the models of all related dynamic systems. The embedded system is then evaluated as it interacts with the overall plant model.

We now demonstrate how this can be accomplished with the National Instruments LabVIEW platform using co-simulation between SCADE and the National Instruments tools through VeriStand™.

As in the previous cases, it is possible to establish co-simulation between LabVIEW and SCADE. The Figure 9 illustrates white-box co-simulation. In this example, the SCADE generated code is wrapped in a proper dll that is imported into the LabVIEW environment.

Now moving to HiL, we see in Figure 10Figure 10 below how the controller code for the cruise control SCADE model can be generated for the target processor and loaded into a LabVIEW platform (PXI, CompactRIO).

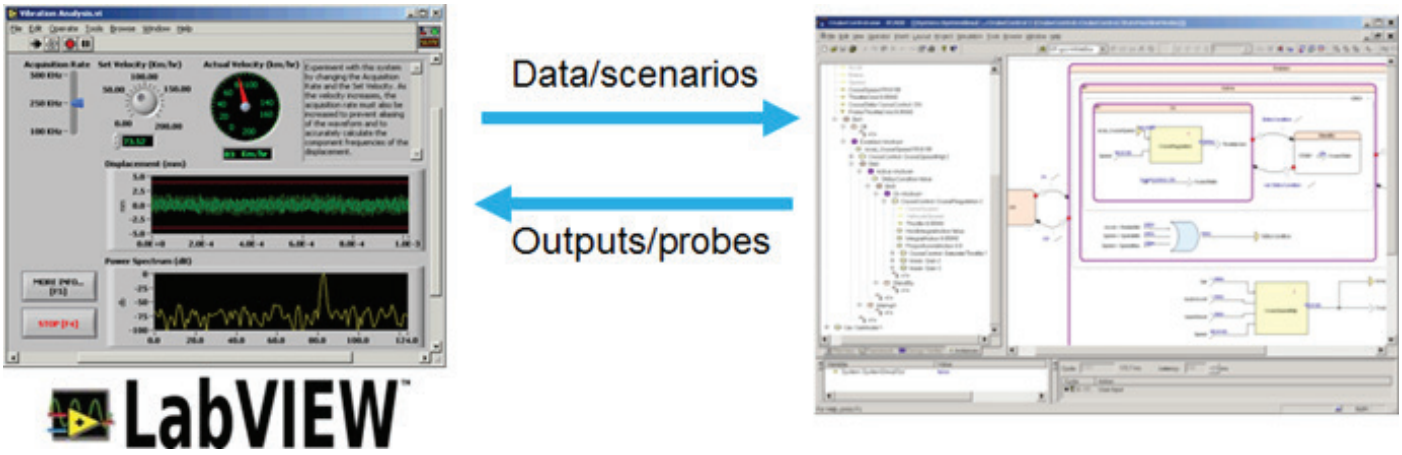

Figure 9: White-box Co-Simulation between LabVIEW and SCADE

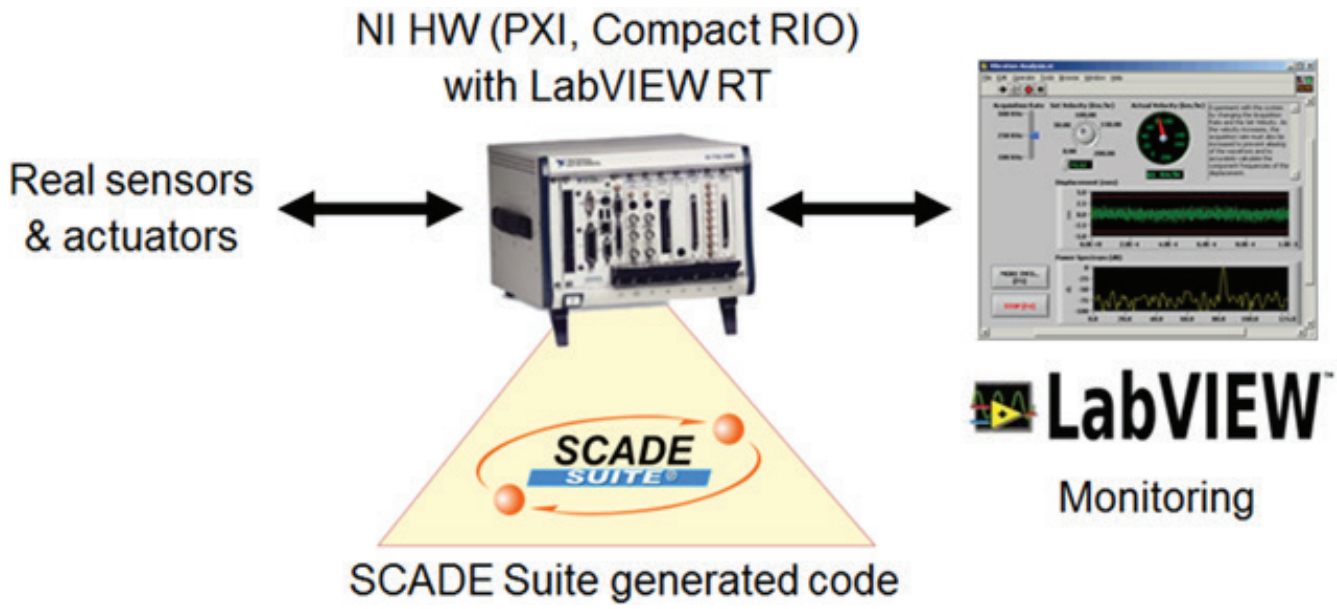

Figure 10: HiL Simulation

3 Bringing it all together

In the first two sections of this paper, we have described an efficient way to develop the controller software and to verify and validate it through co-simulation with a plant model expressed in various modeling languages and tools, e.g. MathWorks Simulink, ANSYS Simplorer, Modelica, LMS AMESim, and National Instruments LabVIEW. Some co-simulation mechanisms have been implemented through the implementation of specific wrappers; some have been implemented through the functional mock-up interface (FMI) standard.

However, in order to implement a full V-process for the entire product lifecycle, we need a structured approach. We have decomposed the systems design and validation flow in three parts:

- Requirements management and functional design
- Detailed design and optimization
- Integration and Validation.

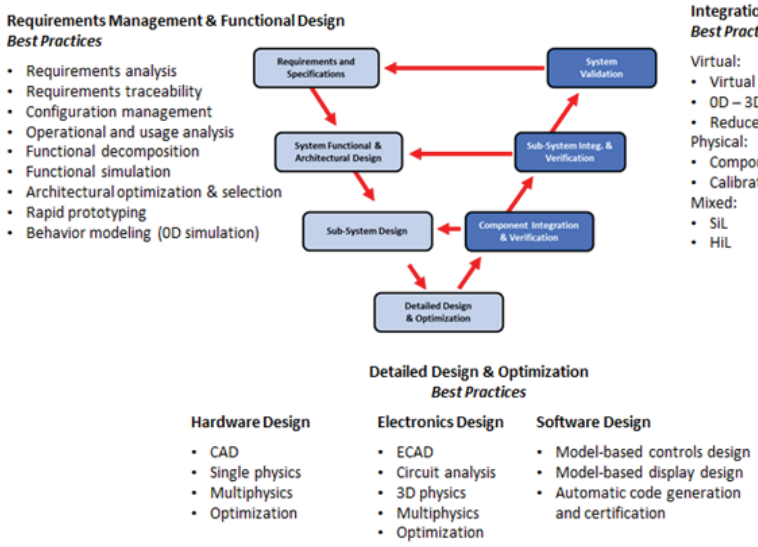

Figure 11: Systems Engineering and Best Practices

This is illustrated by Figure 11 below which describes, for each of these phases, the list of required best practices.

In order to support and structure these processes, the SCADE System tool [5] was created. SCADE System is a SysML-based tool for both functional decomposition and architectural design. Because SCADE System and SCADE Suite are hosted in an integrated development environment, a single platform is used to manage the physical description of the system, the associated models and the certified code generation, and the co-simulation mechanisms that have been described in the previous section. On this basis, Figure 12Figure 12 below describes a full systems engineering flow implementing the following steps:

- Starting from product requirements, the functional and architectural descriptions of the system are created with SCADE System
- On the basis of the system architecture, the controller model is created with SCADE Suite and co-simulation is performed between the SCADE controller and a 0D physical system model created in Simplorer.

In many cases, it will be useful to start physical modeling by using the appropriate detailed simulation software, as provided by ANSYS (e.g. Maxwell, Fluent, Mechanical). These tools allow multiphysics simulation and the creation of reduced order models (ROM) that can be used in the above Simplorer co-simulation with SCADE. These ROMs provide an efficient way to implement 0D co-simulation, while preserving the accuracy of the simulation results.

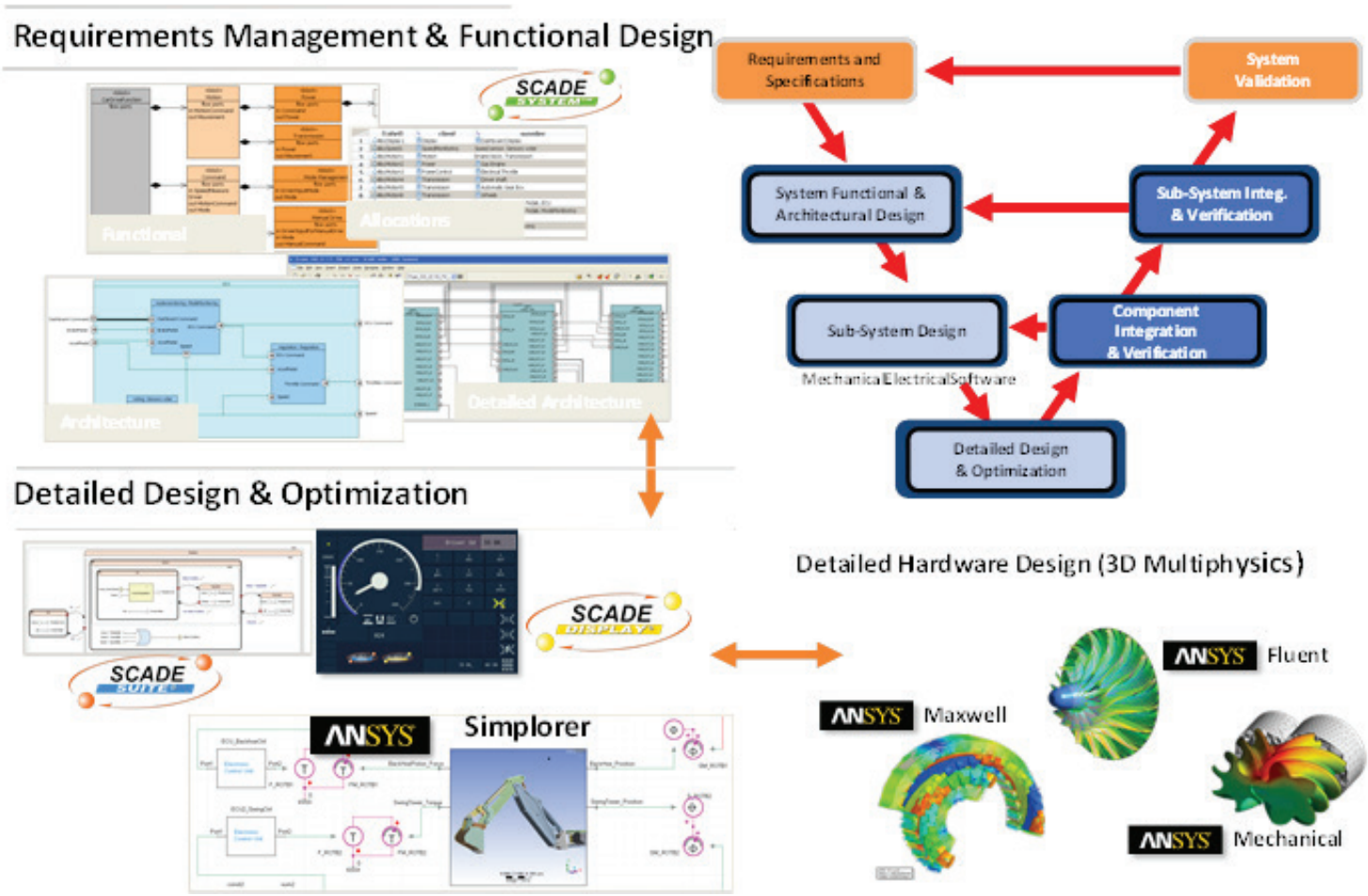

Figure 12: Implementing a Systems Engineering Flow

4 Conclusion

As a conclusion, in this paper, we have demonstrated how systems and software engineers can jointly benefit a complete tool chain to design and simulate a system. The main characteristics of this flow are the following:

- A complete tool chain can easily be assembled to design and simulate the entire system, not just individual parts or sub-assemblies
- Modeling languages are mostly based on open standards (SysML, Modelica, VHDL-AMS, etc.)
- Co-simulation mechanism are increasingly standardized (FMI)
- Scalable simulation fidelity spanning analytical can be achieved through reduced order models (ROM) and co-simulation with world-class 3D solvers
- Co-simulation between model-based controllers and embedded software is achieved thanks to SCADE
- Automatic and certified SCADE embedded code generation guarantees high-fidelity of software simulation.

References

[1] Jean-Louis Colaço, Bruno Pagano, and Marc Pouzet. A Conservative Extension of Synchronous Data-flow with State Machines. In ACM International Conference on Embedded Software (EMSOFT'05), Jersey city, New Jersey, USA, September 2005.

[2] Introduction to ANSYS Simplorer 10.0, ANSYS Inc.

[3] Michael Tiller, Introduction to Physical Modeling with Modelica, Kluwer Academic Publisher, 2001

[4] Modelica 3.3 Specification, Modelica Association, Modelica Association, May 2012

[5] Thierry Le Sergent, Alain Le Guennec, François Terrier, Sébastien Gérard, Yann Tanguy. "Using SCADE System for the Design and Integration of Critical Systems". In SAE Aerotech Congress 2011.

[6] Thierry Le Sergent, Mathieu Viala, Alain Le Guennec, Frédéric Roméas. Integrating System and Software Engineering for Certifiable Avionics Applications. In Avionics 2012.

[7] Thierry Le Sergent, Frederic Roméas, Olivier Tourillion. Integrating System and Software Engineering Activities for Integrated Modular Avionics Applications. In 2012 SAE Aerospace Electronics and Avionics Systems Conference, Phoenix Arizona, USA.

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
U.S.A.
724.746.3304
ansysinfo@ansys.com

Toll Free U.S.A./Canada
1.866.267.9724

Toll Free Mexico
001.866.267.9724

Europe
44.870.010.4456
eu.sales@ansys.com

ANSYS, Inc. is one of the world's leading engineering simulation software providers. Its technology has enabled customers to predict with accuracy that their product designs will thrive in the real world. The company offers a common platform of fully integrated multiphysics software tools designed to optimize product development processes for a wide range of industries, including aerospace, automotive, civil engineering, consumer products, chemical process, electronics, environmental, healthcare, marine, power, sports and others. Applied to design concept, final-stage testing, validation and trouble-shooting existing designs, software from ANSYS can significantly speed design and development times, reduce costs, and provide insight and understanding into product and process performance.
Visit www.ansys.com for more information.