Preprint
UCRL-JC-146628

Optimum Vehicle Component Integration with InVeST (Integrated Vehicle Simulation Testbed)

Walter Ng, Erma Paddack, Salvador Aceves

This article was submitted to
ITS (Intelligent Transportation Systems) America 2002 12th Annual Meeting and Exposition, Long Beach, California, April 29-May 2, 2002

U.S. Department of Energy

Lawrence Livermore National Laboratory

December 27, 2002

Optimum Vehicle Component Integration with InVeST (Integrated Vehicle Simulation Testbed)

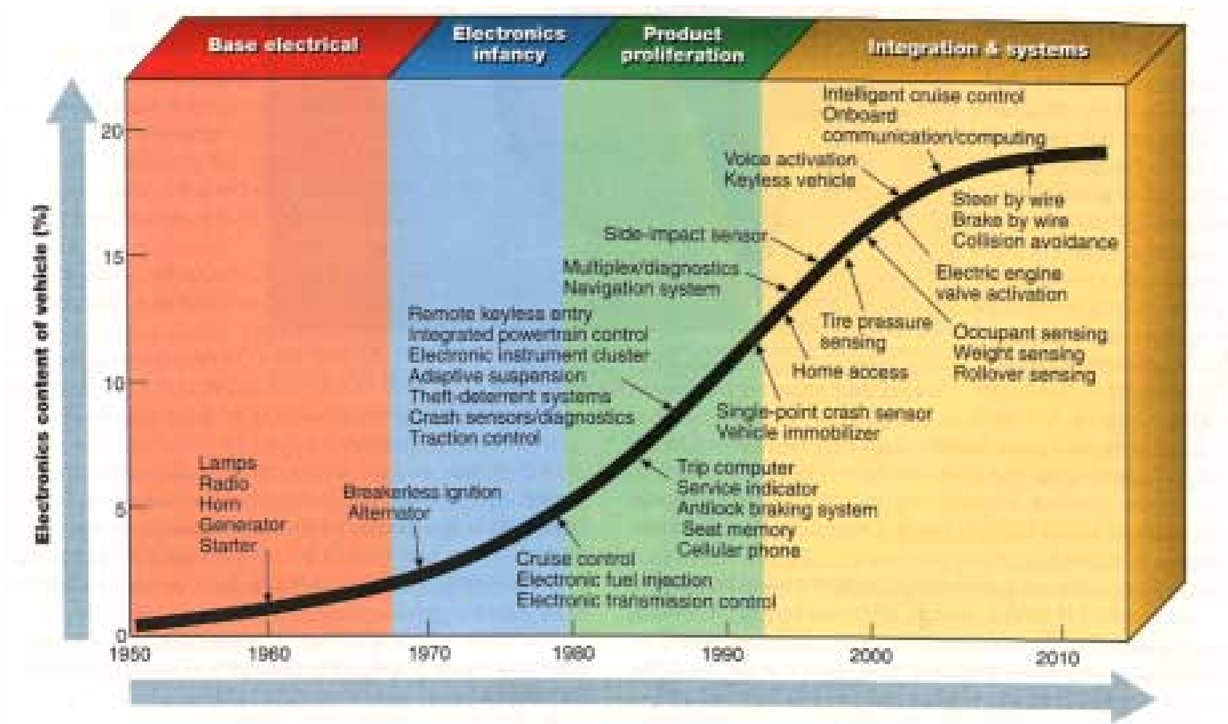
Walter Ng, Erma Paddack, Salvador Aceves
Lawrence Livermore National Laboratory
PO Box 808, L-644
Livermore, CA 94551
SAceves@LLNL.GOV
Phone (925) 422-0864
Fax (925) 423-7914

Abstract

We have developed an Integrated Vehicle Simulation Testbed (InVeST). InVeST is based on the concept of Co-simulation, and it allows the development of virtual vehicles that can be analyzed and optimized as an overall integrated system. The virtual vehicle is defined by selecting different vehicle components from a component library. Vehicle component models can be written in multiple programming languages running on different computer platforms. At the same time, InVeST provides full protection for proprietary models. Co-simulation is a cost-effective alternative to competing methodologies, such as developing a translator or selecting a single programming language for all vehicle components. InVeST has been recently demonstrated using a transmission model and a transmission controller model. The transmission model was written in SABER and ran on a Sun/Solaris workstation, while the transmission controller was written in MATRIXx and ran on a PC running Windows NT. The demonstration was successfully performed. Future plans include the applicability of Co-simulation and InVeST to analysis and optimization of multiple complex systems, including those of Intelligent Transportation Systems.

Introduction

Vehicle designs are becoming more complex and sophisticated. Industry sources estimate that as much as thirty-six percent of the production cost of a present-day, new vehicle is allocated to electronics and electronic control systems. This allocation is forecasted to increase rapidly. For example, parts suppliers will provide an ever-increasing array and variety of electronic and electromechanical components, and any selected component must be rapidly incorporated into the overall design in the most efficient manner possible. The issue of component controls interfacing, communication protocols, flexible data bus, and multiplexing architecture become critically important. Close coupling of many distributed electronic control systems will also be required, and different equipment suppliers will develop many candidate solutions. Figure 1 shows the trend of electronics content per vehicle [1]. The ability to efficiently model and simulate these interactions is essential for the accelerated development of more fuel-efficient and safer vehicles with reduced emissions. Currently there does not exist a universal modeling and simulation capability that addresses, in an integrated fashion, all vehicle electronic and mechanical subsystems. There is also no capability that captures the interactive effects between vehicle electronics and functional subsystems. Rather, individual subsystem simulations are independently performed using various commercial software packages, but the increasingly important integrated controls issue is not being adequately addressed.



Electronics content per vehicle.

Automotive Engineering International/September 1998

Figure 1. Growth of electronics content in new vehicles as a function of model year.

This paper reports our past work and future plans to design and develop a computational environment where existing and newly developed models of automotive subsystems can be selected and executed as an integrated component of a more complex system model in an Internet environment. The goal is to develop a methodology that allows the use of multiple modeling languages and running on different computer platforms, while at the same time providing full protection for proprietary models. This project has been titled Integrated Vehicle Simulation Testbed (InVeST), and has been performed at LLNL and in Detroit in collaboration with multiple industrial partners, including General Motors, Daimler Chrysler, Delphi, Visteon, Siemens, Hewlett Packard, and modeling/simulation software developers, including Applied Dynamics International, WindRiver, Ansoft, Avant!, MathWorks Inc., iLOGIX Inc., and Sun Microsystems.

The paper also includes a brief discussion on how Co-simulation and InVeST can be employed to analyze and optimize Intelligent Transportation Systems (ITS). Current research shows great promise for this complex and very important application.

Review of Existing Technologies

There are three ways to integrate models developed using different modeling languages into a simulation hierarchy: standardize the development language/tool, build a translator that understands and can interact with all modeling languages, and lastly, using the Co-simulation environment. The pros and cons are discussed below:

Standard Modeling Language

Models developed using a standard language will, in most cases, interact with each other. However, existing models are typically developed using different modeling languages favored by the individual code developer. Rewriting all models to a standard language is a monumental undertaking for the following reasons:

- Difficulty in obtaining consensus on the selection of a standard modeling language due to participants own familiarity and preference.
- Rewriting takes a lot of time; costs are prohibitive.
- Re-validation of new models is necessary.
- Different modeling languages are more suitable for particular situations.
- Time and cost can be prohibitive.

Translator

Another common solution is to build a translator that can understand all modeling languages in use and translate all sub-system models into standard executable codes. These code modules can then be linked and executed by the simulation engine (see Figure 2). In this case, existing models are not modified. However, the translation process invalidates the validation of the original models and the code modules or the translation process must be validated again. The disadvantages of building a general translator are:

- Building a translator and the simulation engine is a very complex undertaking and involves significant effort and cost.
- It is very time consuming.
- It requires validation of the translation process, the resulting code modules and the simulation engine.

Co-simulation

Co-simulation is a relatively new approach. In Co-simulation, only input and output data are transferred between models under the coordination of a Commercial Off The Shelf (COTS) Co-simulation software package. Co-simulation does not require any modification or translation of the original model. Better yet, it does not even require access to the source code of the model. This is a very important issue in protecting proprietary information and intellectual property rights among project partners in the competitive market. The Co-simulation methodology is illustrated in Figure 3. The advantages of this approach are many:

- No interference to existing models in terms of ownership, control, and proprietary protection.
- No modification or re-validation of existing models is needed.
- Relatively inexpensive because no rewriting or translation is needed. This makes Co-simulation the most cost-effective option. This is an overwhelming advantage that preserves existing model integrity and substantial savings in time and expenses.
- Complex system models can be executed in a multi-platform, multi-language, over an Internet environment.
- The integration is at the modeling language (tools) level which eliminated the need to deal with the actual models.
- Co-simulation is applicable in all levels of organizations: project, group, department, corporation, business partners, and industry.

All these advantages make Co-simulation the best choice for InVeST.

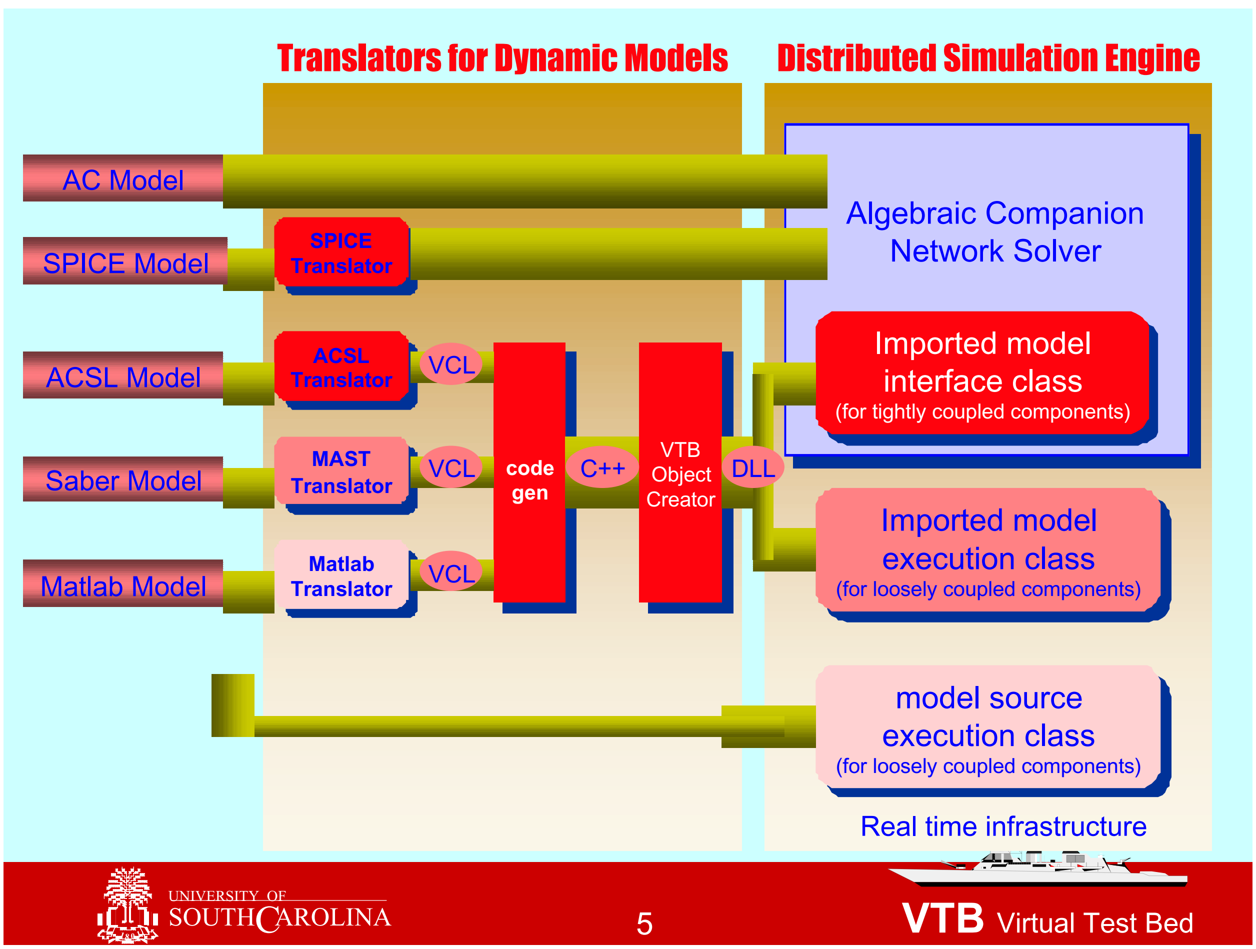


Figure 2. Schematic of a software model translator [2].

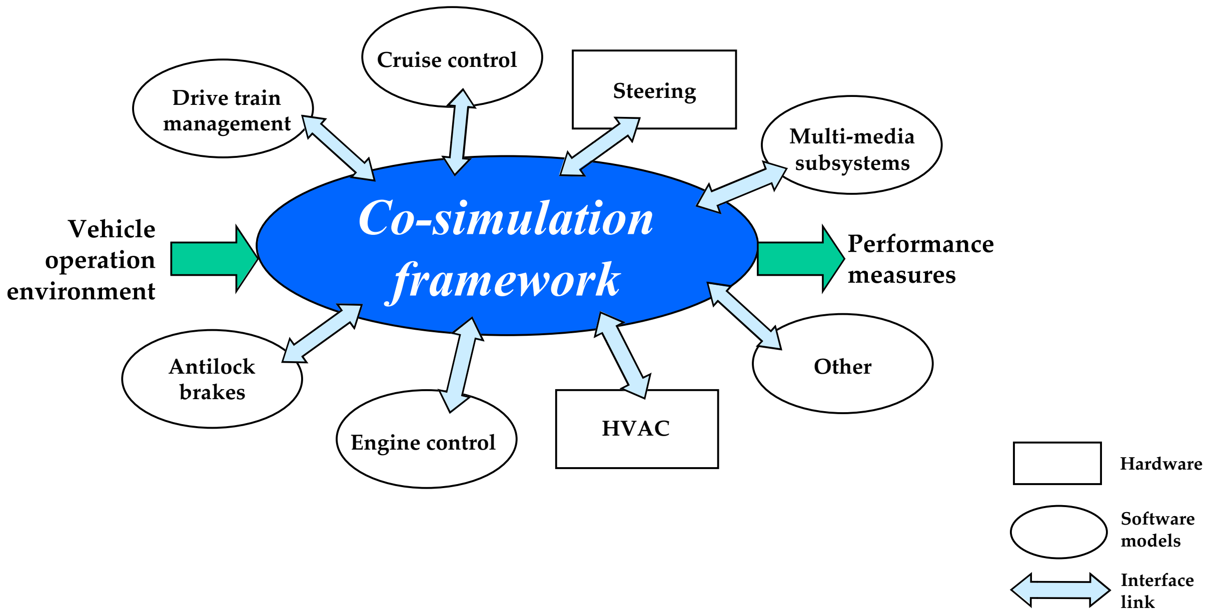


Figure 3. Example of an overall vehicle systems architecture employing Co-simulation.

Proof of Concept Demonstration

Co-simulation is a relatively new technology. This makes it important to conduct a Proof-of-Concept demonstration to verify that a working system can be developed. It is also necessary to test the commercial off the shelf (COTS) software available in the market place for applicability to InVeST. The COTS product chosen for the prototype and demonstration is pLUG&SIM ; a software product released by Integrated Systems, Inc., which later became WindRiver Corporation.

A project team was assembled with personnel from LLNL, WindRiver, Avant! and Sun Microsystems, with input from General Motors Research (GMR). Several automotive systems were considered for the proof of concept demonstration. Finally, it was decided to use a complex model of an automobile transmission written in SABER. The transmission controller was removed from the SABER model and re-coded in MATRIXx. MATRIXx was run on a PC running Windows NT, while SABER was run on a four-processor SUN/Ultra 80 running SOLARIS (see Figure 4). The computers were connected through a network hub using TCP/IP protocol. The team met for the first time two days before the scheduled demonstration. We were able to overcome numerous expected and unexpected hardware and software problems and finally succeed in a flawless demonstration of a multi-host, multi-tool, Co-simulation seen for the first time in public in Detroit. The proof of concept demonstration was successfully conducted. The demonstration provided a crucial proof of concept for the InVEST technical approach by demonstrating a multi-host, multi-tool Co-simulation. It also demonstrated the effectiveness of the InVeST team formed by LLNL and industrial partners.

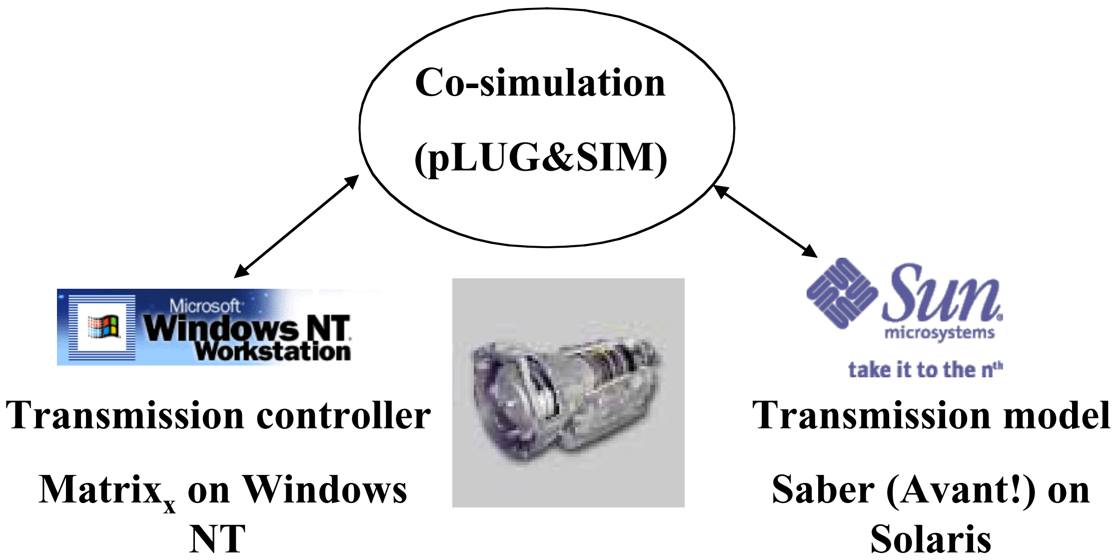


Figure 4. Software and hardware configurations used for the proof of concept demonstration.

Investigation of other key issues

After the successful demonstration of InVeST, the work focused on a further study of the capabilities and limitations of Co-simulation technology. The goal is to provide practical automotive design support for OEMs and suppliers, and collaborate with tool vendors in the development of adaptors.

Multiple back planes

After surveying the market, there is at least one other vendor; iLOGIX Corporation, that has a product (Statemates/MAGNUM s Trailblazer) that has the capability to be the back plane in the InVeST Co-simulation environment. This product also includes application programming interfaces to SABER (Avant!), MATRIXx (WindRiver), and Simulink (Mathworks). These are the most popular modeling tools in use today in the industry. Another possibility is a software product under development from Mathworks.

Network latency

One of the frequently raised concerns about Co-simulation is the latency between models where the network is subject to traffic delays. While it is true that network traffic are unpredictable and synchronization between models that run under different time steps may cause one model to wait for the results from the other. However, there are ways to avoid or minimize the difficulties by selecting the appropriate transmission media and the proper placement of the models in the right computing environment. For example, share memory may be used in a multi-processor system where the transfer of data and control between models residing in different processors within one computer will take only one memory access cycle. The concept of Data Flow used in the massively parallel computer systems can help resolve the synchronization issues. Further investigation and a prototype should be built to investigate the effect of various approaches and explore the scalability issues. There will also be a limit to scalability. Nevertheless, with the continued increase in power of hardware and distributive computing technology, the limit of scalability has also been raised to a new height.

Applicability of InVeST and Co-simulation to Intelligent Transportation Systems

The design, development and deployment of Intelligent Transportation Systems (ITS) require the successful integration of many disparate building blocks, including subsystems for sensing, control, communication, computation and display. These components may be in vehicles, connected with travelers themselves, on roadways, or in fixed-location centers. Figure 5 shows high-level ITS functions defined by the National ITS Architecture. Each of these functions, when present in a system, consists in turn of numerous components interconnected with each other at finer levels of detail. In many respects, this system is similar to an automobile, where different components interact with each other in complex, non-linear fashion (Figure 3).

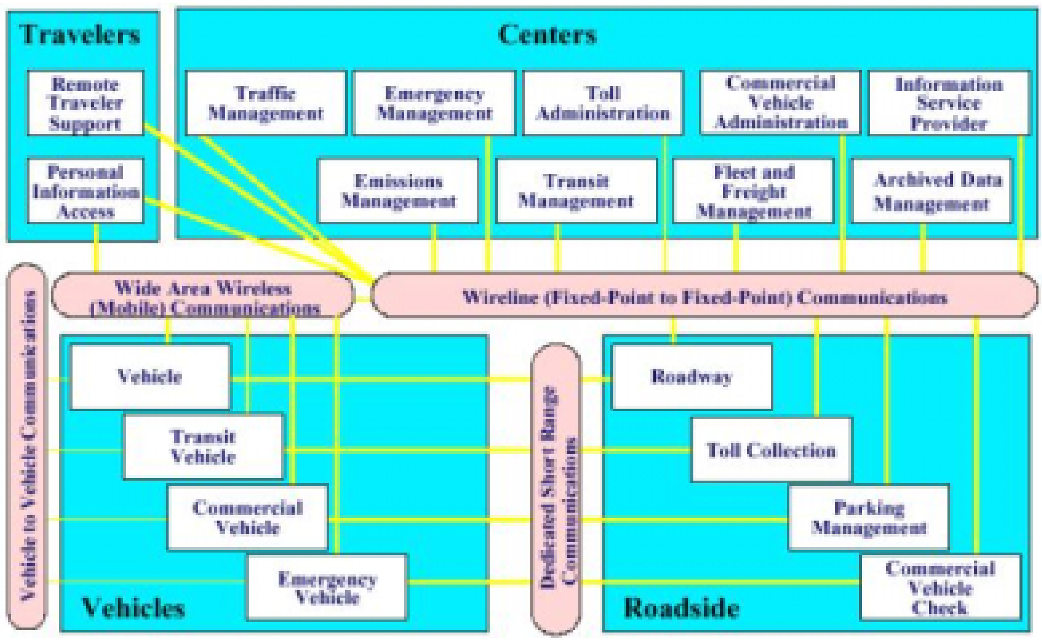


Figure 5. High-level ITS functions defined by the National ITS Architecture. Each of these functions, when present in a system, consists in turn of numerous components interconnected with each other at finer levels of detail.

To facilitate seamless connection of components into an effectively functioning overall system, USDoT has established detailed architectural and communication specifications for the many functions imbedded in Figure 5. In many cases, standards are also being established for specific interconnected subsystem types. As these standards become accepted and incorporated into commercial products, the problem of integrating components should become easier. However, because of the complexity of real subsystems, the enormous number of ways they may interact with each other and with their environment, and the inevitable differences in technology among various suppliers, the integration process will never become completely a routine process. Furthermore, because most communities have a substantial existing investment in legacy systems, there will continue to be a difficult integration problem in most affordable upgrades of ITS systems for some time to come.

At present, system engineers integrate ITS systems based on a combination of hard-earned knowledge and experience, on data from manufacturers, and on testing interconnected system components. This laborious process is costly in time, money, and potential performance degradation. In an era when it is critical to demonstrate cost-effective solutions to increasingly onerous traffic problems, public confidence can be easily undermined by cost overruns, implementation delays, and performance losses. For all these reasons, it would be extremely desirable to have an efficient simulation environment for testing ITS deployments prior to hardware purchase decisions and expensive prototype testing. Considering the similarity between the vehicle system design and the system components in ITS, it is apparent that InVeST and Co-simulation have great applicability to the general field of ITS.

InVeST would do for ITS simulation what the National Systems Architecture does for ITS systems themselves: support seamless integration of system models. It would help assemble existing models into functioning simulations, and it would also be the basis for standards and templates upon which vendors could build models for easier interconnection in the future. The basic benefit, of course, would be to streamline the process of system integration, and thus facilitate the deployment of ITS systems nationwide.

Conclusions

The Co-simulation technology has been proven to be an effective solution to integrating models of sub-systems generated with different modeling languages residing in multiple hosts in an Internet environment. A user can effectively assemble a virtual automobile (Figure 6) by selecting from the component library (Figure 7) while sitting in front of a terminal. While additional investigation is needed to explore the scope of scalability and time synchronization of simulation steps, the many advantages of Co-simulation justify the continual investigation into further application of this technology. Future research needs for this technology include:

- Design and develop the infrastructure: WEB, Component Library, standardize glossary of terms and convention.
- Investigate Network simulation requirements and select tool to use.
- Build a medium scale Co-simulation prototype with as complex a scenario as possible.
- Explore test drive/track/environment effect modeling.
- Explore low cost simulator and trainer applications.

Co-simulation and InVeST also have great potential for applicability to Intelligent Transportation Systems. The applicability is currently being explored, and great possibilities exist to streamline the deployment of ITS systems nationwide.

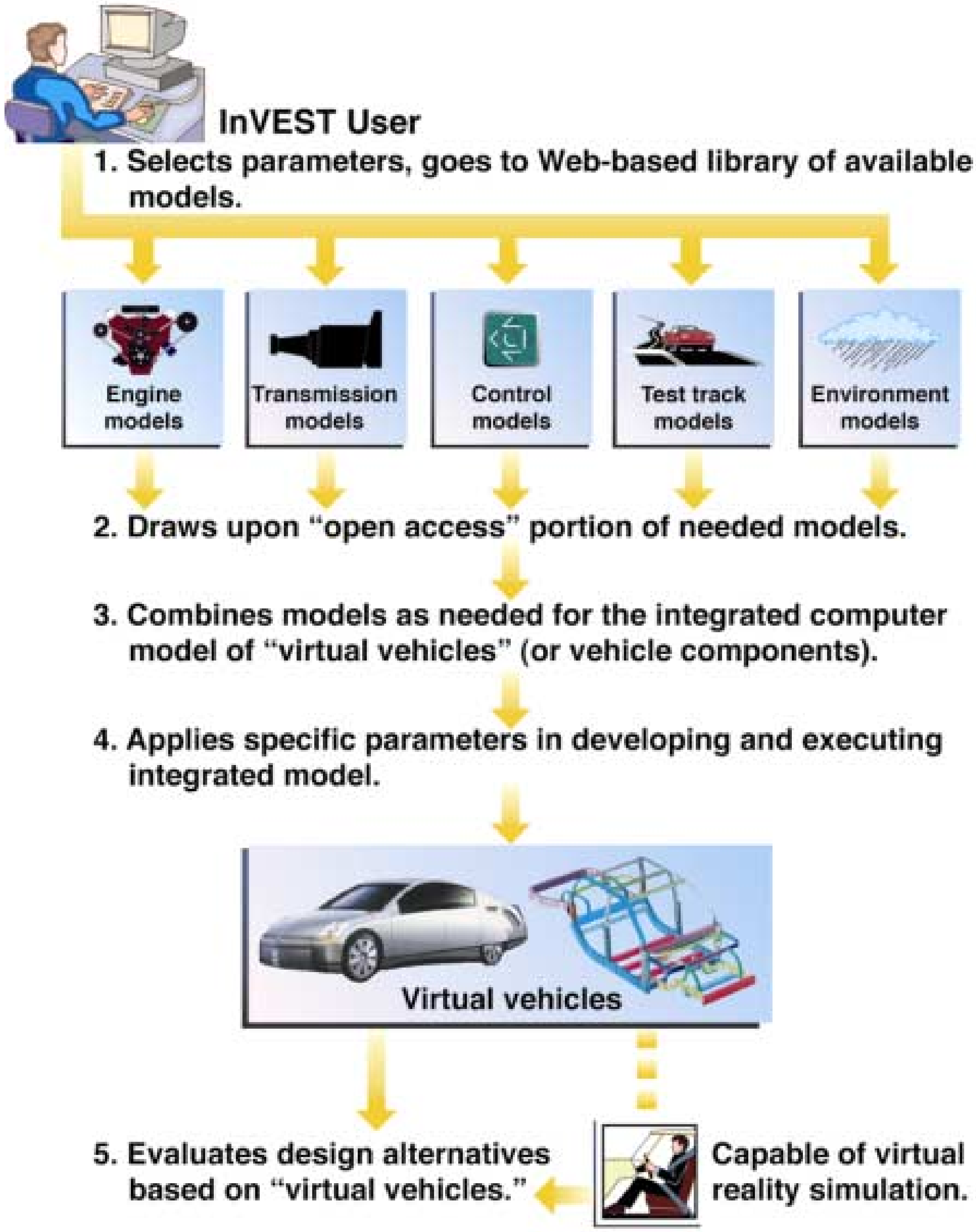


Figure 6. Schematic of the procedure for analysis and development of virtual vehicles using InVeST.

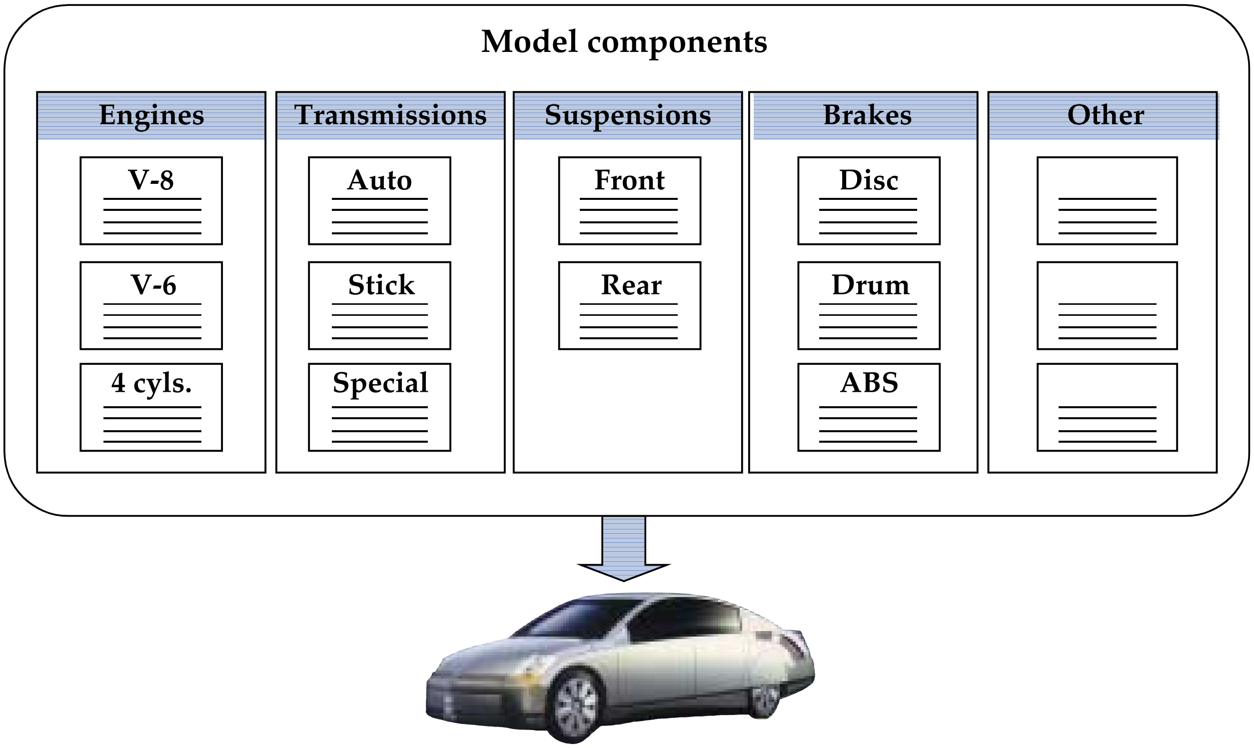


Figure 7. Virtual component library for use in developing virtual automobiles with InVeST.

Acknowledgements

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References

1. Buchholz, K., Electronics Innovations, Automotive Engineering International, Vol. 106, No. 9, September 1998, pp. 35-38.
2. Gokdere, L.U., Benlyazid, K., Santi, E., Dougal, R.A., and Brice, C.W., A Virtual Prototype for A Hybrid Electric Vehicle, submitted to Journal of Mechatronics, 2001.

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551