

Model Specification Process Standard

TABLE OF CONTENTS

1.	Scope	2
1.1	Background	2
1.2	Purpose	2
1.3	Model Development.....	2
1.4	Application	5
2.	References	6
2.1	Related Publications.....	6
3.	Definitions.....	6
4.	EE Commodity Analysis and Modeling Considerations.....	10
4.1	Analysis Requirements.....	10
4.1.1	Types of Analysis.....	11
4.1.2	Analysis Methodology and Simulation Tools	11
4.2	Modeling Considerations	12
4.2.1	Modeling of Wholly Electrical Commodities	12
4.2.2	Modeling of Mechatronic Commodities	12
4.2.3	Modeling Non-Electrical Entities in the Control Loop	12
4.2.4	Modeling of Control Modules.....	13
4.2.5	Control System Modeling	13
5.	Modeling Fidelity	13
5.1	Levels of Model Refinement	13
5.2	Levels of Feature Content	15
5.3	Forms of Simulation Results.....	16
5.4	Diminishing Returns of Fidelity	16
5.5	Accommodating Different Fidelity Needs in a Model.....	16
6.	Modeling Assumptions:	16
6.1	Units of Measurements.....	16
6.2	Connection Type Definition.....	16
6.2.1	Standard Sign Convention.....	17
6.3	Reference	17

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

TO PLACE A DOCUMENT ORDER: (724) 776-4970 FAX: (724) 776-0790
SAE WEB ADDRESS <http://www.sae.org>

6.4	Model Assumptions	17
6.5	Environmental and Operating Conditions	17
6.6	Input Parameters	17
6.7	Supplementary Output Variables	18
7.	Model Architecture	18
7.1	External Signal Interface Requirements	18
7.2	Model Internal Requirements	18
7.3	Textual Messages	19
7.3.1	Informational Messages	19
7.3.2	Overstress Messages	19
7.3.3	Warning Messages	19
7.3.4	Error Messages	19
8.	Validation Requirements	19
8.1	Functional Test Procedures	19
8.2	Characterization Test Procedures	19
8.3	Data Reduction Procedure	19
8.4	Validation Procedure and Data	20
8.5	Validation Criteria	20
9.	Deliverables	20
9.1	Models	20
9.1.1	Source Code	20
9.1.1.1	Header	20
9.1.1.2	Comments	20
9.1.1.3	Programming Languages	20
9.1.1.4	Drawing Symbols	20
9.1.2	Supporting Deliverables	21
9.2	Documentation	21
9.2.1	Model Application Documentation	21
9.2.1.1	Description	21
9.2.1.2	Netlist or Dataflow Hierarchy	21
9.2.1.3	Input Parameters	21
9.2.1.4	Connection Points	21
9.2.1.5	Output Variables	21
9.2.1.6	Usage Notes	21
9.2.2	Support Documentation	21
9.2.2.1	Model Features	21
9.2.2.1.1	Connection Points	21
9.2.2.1.2	Parameters	21
9.2.2.2	Template Usage	21
9.2.2.3	Model Limitations	22
9.2.2.4	Model Theory	22
9.2.2.5	Characterization Test Procedures and Data	22
9.2.2.5.1	Equipment Used	22
9.2.2.5.2	Data Acquisition Procedure	22
9.2.2.6	Model Validation Data	22
9.2.2.7	Model Datafile	22
9.2.2.8	Appendices	22
9.3	Support	22
Appendix A	EE Commodities	23
Appendix B	Example Saber Modelfile Header Layout	30

1. Scope—The modeling requirements listed in this document provide guidelines for specifying:

Analysis requirements
 Computer-Aided Engineering (CAE) simulator
Model fidelity
System assumptions
Model architecture
Validation requirements
 Deliverables

for electrical/electronic components and the integral interfacing *systems* (mechanical, thermal, hydraulic) associated with these components.

The *simulation models* are primarily intended to facilitate *functional* specification and operational evaluation between component and *system* manufacturers, and are not intended to otherwise constrain or restrict the use of CAE *simulation* tools for automotive design, *analysis* and *validation*.

1.1 Background—The advances in computer technology since the 1970's have enabled the development of powerful CAE tools. These tools are being increasingly applied to the highly competitive automotive industry for the development of new components and *systems*. CAE tools allow companies to reduce the number of prototype phases and are considered key enablers in their product development process to produce designs in less time, with higher quality and lower costs.

However, over the years many custom tools have been developed with specific modeling and *analysis* objectives. Today, suppliers are asked more and more to deliver component *models* that satisfy different sets of objectives. This leads to the development of numerous simulator specific *models* of the same basic component. When component *models* are developed with different objectives and with unspecified degrees of *fidelity*, it is difficult to capture component interactions that allow *system* trade-off investigations and thus, *system* design optimization.

With the need to produce more cost-effective designs, the automotive industry is ready for the development of the Model Specification Process Standard.

1.2 Purpose—The Model Specification Process Standard is intended to define a common set of standards in defining *model* requirements for automotive electrical/electronic and mechatronic components. These requirements include *functional* specifications, their intended applications and the deliverables.

1.3 Model Development—This standard provides a guide for writing a *simulation model* specification by the end user, known as the *Requester*, and is detailed in the Model Specification Process shown in Figure 1.

The general process steps are:

- a. The *Requester* requires a component *model* to perform *system simulations* and submits the component specification based on this standard to the component supplier, known as the Producer.
- b. The Producer receives the *model* specification and, possessing detailed knowledge of their component, develops a specific component *model*. This *model* can be developed from the ground up or it can be developed from generic component *models* that are assembled to provide the proper performance. These generic component *models* can be developed by the CAE simulator supplier or by other component manufacturers.
- c. The Producer validates the specific *model* against the Model Specification and delivers the *model* to the *Requester*.
- d. The *Requester* receives the requested *model* and performs the necessary *system simulations*.

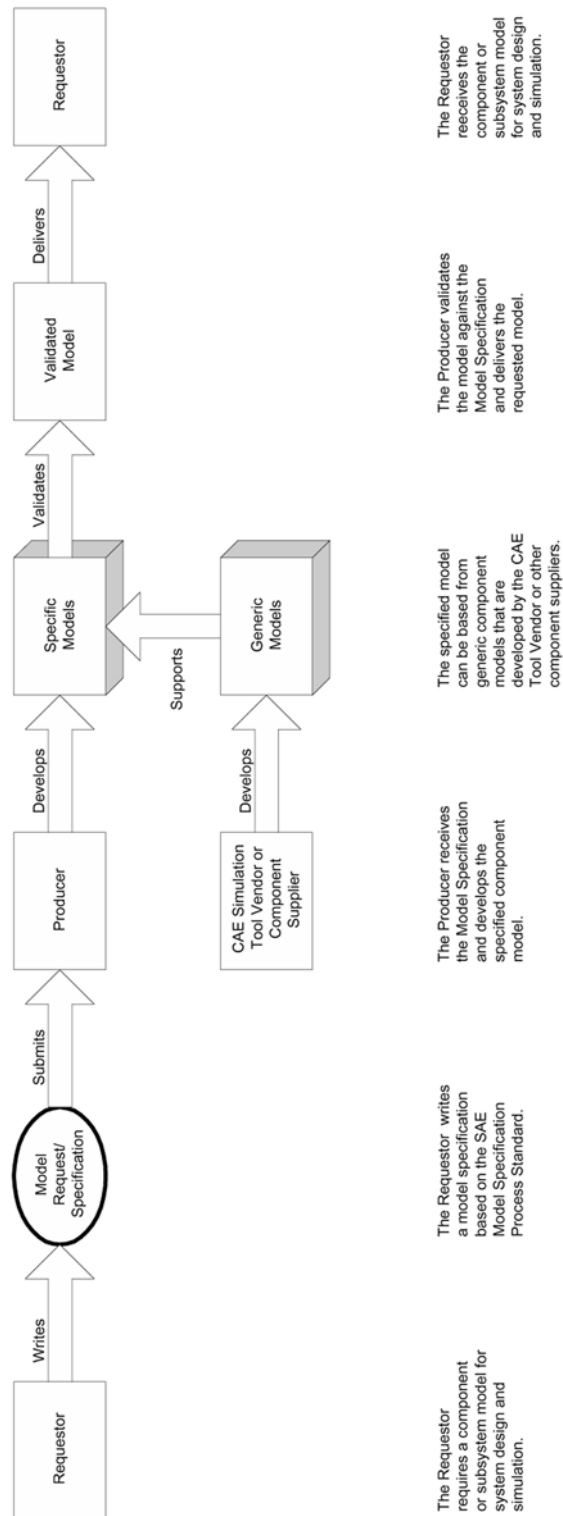


FIGURE 1—SIMULATION MODEL DEVELOPMENT UNDER THE MODEL SPECIFICATION PROCESS STANDARD

This process has the benefits of having a common vocabulary and standard interpretation of what is being requested and what will be delivered. Essentially, this Standard improves the communication between the system integrators and the component manufacturers in the automotive industry and will enable the development of automotive systems that represent high customer *value*.

This document will focus on developing the *model* requirement guidelines (highlighted oval in Figure 1) and this will allow the development of the SAE Commodity Model Standards.

- 1.4 Application**—This standard provides a guide for the development of *model request* specifications on a commodity-by-commodity basis. Each request specification describes the fundamentals of commodity behavior and contains a general “request form.” This request form in turn contains pre-defined checklist table entries for *feature/level* options as well as placeholders for free-form text descriptions, graphics and other request-dependent information. A *model Requester* can then fill out a “blank” form for a given commodity to specify the level of detail and desired functionality that is expected for a generic *model* of a particular device. A *model Requester* can also request characterization of an existing generic *model* to make a *model* of a specific component or part number.

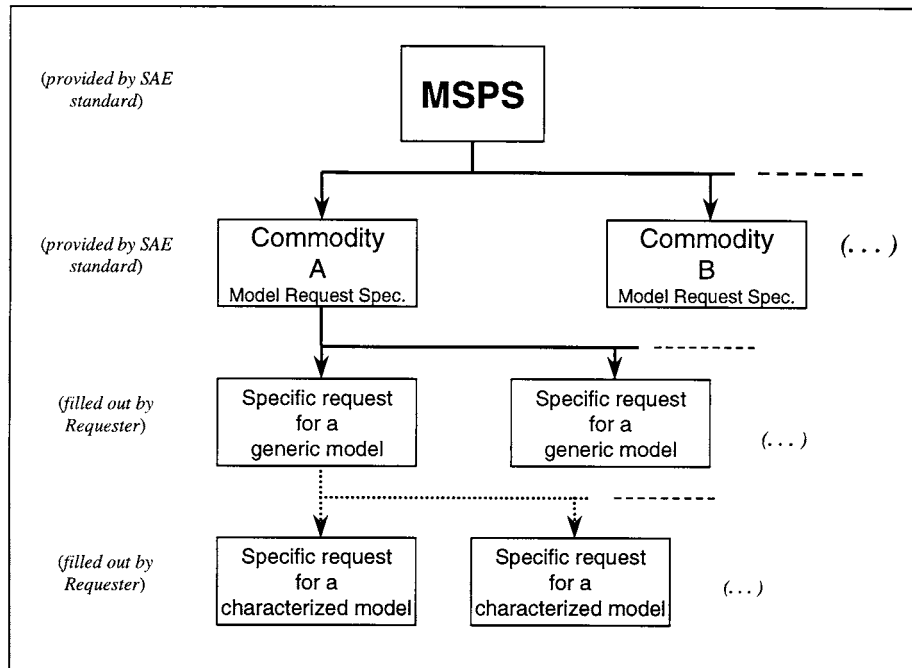


FIGURE 2—HIERARCHY OF COMMODITY MODEL SPECIFICATIONS
IN RELATION TO DEFINED STANDARDS

2. References

2.1 Related Publications—The following publications are provided for information purposes only and are not a required part of this specification.

2.1.1 IEEE PUBLICATIONS—Available from ANSI, 25 West 43rd Street, New York, NY 10036-8002 or Web server <http://web.ansi.org>, or <http://standards.ieee.org>.

IEEE Std 1076.1-1999—IEEE Standard VHDL Analog and Mixed Signal Extensions

IEEE Std 1076.4-1995—IEEE Standard for VITL Application-Specific Integrated Circuit-ASIC

IEEE Std 1364-1995—IEEE Standard Hardware Description Language Based on Verilog

IEEE Std 1499-1998—IEEE Standard Interface for Hardware Description Models of Electronic Components

IEEE Std 1481-1999—IEEE Standard for Integrated Circuit (IC) Delay and Power Calculation System

2.1.2 RELATED MODELICA INFORMATION—Modelica and the Modelica Association, <http://www.modelica.org/documents.shtml>.

The object-oriented modeling language Modelica is designed to allow convenient, component-oriented modeling of complex physical systems, e.g., systems containing mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. The free Modelica language, free Modelica libraries and Modelica simulation tools are available, ready-to-use and have been utilized in demanding industrial applications, including hardware-in-the-loop simulations. The development and promotion of Modelica is organized by the non-profit Modelic Association.

3. Definitions

3.1 Accuracy—The numerical closeness of fit of *model* behavior in *simulation*, to target data from the real world.

3.2 Algorithm—A formal recipe for solving a specific type of problem.

3.3 Analysis—Extracting behavior from description.

3.4 Argument—A piece of information that can be assigned or “passed” to an *instance* of a *model* prior to *simulation*. An avenue for setting or passing a *parameter* or a *variable*.

3.5 Behavioral—Descriptive of a *model* designed around results that represent physical behavior and internal workings of a device. See also *Functional*.

3.6 Connection—A modeling artifact that represents communication of data or signals into or out of other *model* components such as junctions, blocks, or circuit components. For electrical circuit *models*, *instances* of a component are embedded into a design document or *netlist* with their pins or *connection points* linked by *nets*.

3.7 Connection Point—An external linkage point on a *model*, usually corresponding to a port, terminal, connector or cavity on the physical device. In most cases, *connection points* must be attached to one or more external components for a valid *simulation*.

3.8 Connection Type—The specific technology or character of a *connection point* on a *model*, defining the allowable *connections* between *models*. The type may determine the physical character of a *connection*, such as electrical, thermal, mechanical or hydraulic, or it may identify a particular *simulation* technology, such as digital, signal *flow* or conserved.

3.9 Convergence—Computing a *model's* behavior with acceptable *accuracy*. Also, solving an implicit mathematical expression until the achieved result agrees with an extrapolated result within some specified *tolerance*. The criterion by which a solution is accepted.

- 3.10 Degradation**—A *dependence* of a device *property* on previous operation of a device.
- 3.11 Distribution**—A non-time series of data *values*. The independent *variable* could be frequency, location, deviation, etc.
- 3.12 Dependence**—A characteristic relationship between *variables*.
- 3.13 Documentation**—The *documentation* of a *model* explains to the user the principles of the *model's* operation and defines its proper use and limitations. Some *documentation* may be contained in the *template* file itself in the form of comments, but usually these must be supplemented with a separate document.
- 3.14 Dynamic Thermal Effects**—A modeling technique including ambient temperature, induced heating and heat transfer from the device to ambient. The resulting device temperature directly affects the electrical, mechanical or other behavior of the *model*. See also *Static Thermal Effects*, *Temperature-dependent Effects* and *Mutual Thermal Effects*.
- 3.15 Feature**—An aspect of an object's nature captured in a *model*, and the capability to control or acquire that aspect in *simulation*.
- 3.16 Feature Level**—Feature Level classifies how an individual *feature* of a *model* can be applied.
- 3.17 Fidelity**—The degree to which a representation, such as a *model*, captures the nature of the real object being represented.
- 3.18 Flat Model**—A *model* that has no hierarchical structure, and which makes no use of subsidiary *models*, hence the opposite of *macromodel*.
- 3.19 Flow**—A channel for data transfer within or among *models* during *simulation*.
- 3.20 Functional**—Descriptive of a *model* designed to statically or dynamically represent its outputs as functionally dependent on its inputs, without regard for the physical processes involved. See also **Behavioral**.
- 3.21 Generation**—Conversion of some other form of energy to thermal energy.
- 3.22 Heuristic**—An adaptive framework for solving a class of problems.
- 3.23 Hierarchy**—Use of the relationships and *dependencies* amongst the elementary components of a system as a basis for allocating functions and their associated higher levels of abstraction such as *Subsystems*.
- 3.24 High-level**—Refers to levels of abstraction in a *model* that represent reduced *functional* specificity and detail; opposite of **Low-level**.
- 3.25 Instance**—An *instance* is a single use of a *model* in a design, analogous to a subroutine call. Each *instance* identifies its parent *template*, its own unique identifier, the *interface* links to the design, and *values* for its attributes.
- 3.26 Interface**—An element of a *model* which offers transfer of data into and out of the *model* during *simulation*. Compare to **Connection Point**.
- 3.27 Linear**—*Model* characteristics that produce outputs which are directly proportional to inputs and give additive results from additive inputs. See also **Non-linear**.
- 3.28 Low-level**—Refers to levels of abstraction in a *model* that represent increased *functional* specificity and detail; opposite of **High-level**.

- 3.29 Macromodel**—A *simulation model* made up primarily or entirely of subsidiary *models*, with few or no mathematical or logical expressions at the immediate level of *model* definition.
- 3.30 Message**—A text string emitted by a *model* during *simulation* to signify the occurrence of some significant event.
- 3.31 Model**—A formal representation of a component or *system* which can be used to compute its expected behavior under specified conditions.
- 3.32 Model Construction**—A *model* can be built in a variety of ways:
- a. Write *behavioral* equations or *algorithms* in a modeling language.
 - b. Write calls to existing *models* in a modeling language.
 - c. Draw diagrams in modeling software using predefined symbols linked to *behavioral* descriptions.
 - d. Capture certain behavior *heuristically* from actual parts or from other *simulations*.
- 3.33 Model Producer**—The component or *system Model Producer* who, because of their detailed knowledge of the requested component, is best suited to obtain or develop a *simulation model* in response to the *model request*.
- 3.34 Model Request**—A specification issued by the customer or procuring agent (the *model Requester*) to the component or *system Model Producer* (the *model Producer*). The request details the desired *functional* behavior in a *simulation model*.
- 3.35 Monte Carlo**—A statistical characterization technique that employs repetitive time-simulation of a *model* with a pseudo-random sequence of perturbations applied to *parameters* and *variables* so as to represent their statistical variability, followed by statistical representation of the resulting population of results.
- 3.36 Mutual Thermal Effects**—A modeling technique including both *dynamic thermal effects* within the *model* and external heat transfer between two or more *models*. See also **Temperature-Dependent Effects**, **Static Thermal Effects** and **Dynamic Thermal Effects**.
- 3.37 Net (Nets)**—A description of how components interconnect.
- 3.38 Netlist**—A formatted representation of the *net*.
- 3.39 Non-linear**—*Model* characteristics that produce outputs which are not directly proportional to inputs. See also **Linear**.
- 3.40 Parameter**—A quantity which remains constant within a *model* during *simulation*, but may be changed between *simulations* and modes of *model* operation. See also **Variable**.
- 3.41 Precision**—The numerical fineness with which quantities are treated in a *model* or in a simulator.
- 3.42 Procedure**—A computational process that must be called or enabled from another modeling entity to cause it to execute. In certain cases, *variable* inputs and outputs and/or *parameters* may have to be specified as part of the call.
- 3.43 Property**—A quantifiable element within a *model*, whether fixed, assignable or dynamically changing.
- 3.44 Radiation**—Propagation of electromagnetic energy.
- 3.45 Refinement**—The process whereby higher-level or abstract ideas are progressively reexpressed in terms of lower level or concrete ones.

- 3.46 Refinement Level**—Refinement Level broadly classifies how a *model* as a whole can be applied in a *system*.
- 3.47 Requester**—The customer or procuring agent charged with specifying and obtaining a *simulation model* for use by the end user.
- 3.48 Sensitivity**—A kind of *analysis* which shows the *dependence* of a *variable* upon a *parameter*.
- 3.49 Simulation**—Computing a *model's* behavior under specified conditions.
- 3.50 Simulation Model**—A logical or mathematical description of the *functional* behavior of a particular device or assembly. When used with an appropriate simulator's engine, the *model* emulates the function of the device in the context of its specific application.
- 3.51 State**—One of a finite set of discrete *values* that an element of a *model* is allowed to have.
- 3.52 States**—A set of internal *variables* of a *model* or *model* component that are related by integration or differentiation, wherein the number of *states* is proportional to the order of differentiation.
- 3.53 Static Thermal Effects**—A modeling technique including both ambient temperature and modifications to that temperature caused by self-heating of a component. Heat transfer out of the device and the effect of self-heating on the device's *functional* behavior are neglected. Thus, although the internal temperature is calculated and reported, there is no *functional* difference between *static thermal* and *temperature-dependent* behavior. See also **Dynamic Thermal Effects**, **Temperature-Dependent Effects** and **Mutual Thermal Effects**.
- 3.54 Steady-State**—Conditions in a *simulation* which can be effectively extrapolated as being directly proportional to time beyond the end of simulated time.
- 3.55 Stress**—A measure of a *property* or *model parameter* exceeding its desired or usual range.
- 3.56 Subsystem**—A *system* that is a part of a larger *system*.
- 3.57 System**—A group of interdependent components which perform a function.
- 3.58 Temperature-Dependent Effects**—A modeling technique wherein certain *model parameters* are represented via user input as being dependent upon ambient temperature. See also **Static Thermal Effects**, **Dynamic Thermal Effects** and **Mutual Thermal Effects**.
- 3.59 Template**—The master copy of a *model* from which *instances* are drawn. The *template* defines the *model's* name, *interfaces*, attributes and behavior.
- 3.60 Tolerance**—The allowable range of values of a *parameter* or *variable*.
- 3.61 Validation**—Testing a *model* via *simulation* to confirm compliance to requirements.
- 3.62 Value**—A single quantity associated with a *model property* at some converged stage of a *simulation*.
- 3.63 Variable**—A *model* quantity that is subject to variation with respect to time or internal *states* of the *model*. See also **Parameter**.
- 3.64 Waveform**—A time series of data *values*. The *values* may be discrete or samples of a continuous *waveform*. The sequence of data *values* may represent a periodic function, a continuously-varying signal without any repetitive *features* or a combination of periodic and non-periodic *features*.

- 4. EE Commodity Analysis and Modeling Considerations**—Automotive EE Commodities are electrical or electronic components of automobiles that are subject to procurement by automobile manufacturers. A structured list of automotive EE commodities is provided in Appendix A as a reference example of the range of electrical and electronic components that are considered to be Automotive EE Commodities as of 1998.

The *Requester* shall specify analysis requirements for the EE Commodity subject to model procurement (and possible eventual commodity procurement). The *Requester's* analysis requirements implicitly define most of the modeling requirements; however, the *Requester* may wish to define specific requirements for how modeling is to be accomplished within the context of the modeling and simulation tools specified in the analysis requirements. The following sections address the analysis requirements first, and then build upon that foundation in discussion of the modeling requirements.

- 4.1 Analysis Requirements**—This section provides guidance as to what analyses the *Requester* will provide for in the model. Other chapters of the Specification define requirements for specifying model composition.

As applicable to a particular model, the analysis requirements defined by the *Requester* shall specify the following (each point is discussed in greater detail elsewhere in this document):

- a. The system context and tool environment in which the model is to be used (4.1.1, 4.1.2).
- b. Which commodity the model is to represent (Appendix A).
- c. Which aspects of the commodity are to be represented in the model (Section 5).
- d. Identifiers such as the filename, model name and graphical symbol representation (9.1.1.4).
- e. The *connection points*, along with the name, type and unit designation of each (6.2).
- f. Model *arguments*, their names, unit designations, default *values* and definitions (6.6).
- g. Existing models that may be used as sub-models within the model (7.2).
- h. Existing models that serve as useful examples in the development of the model (6.3).
- i. *Properties* or behaviors to be characterized within the model (5.2).
- j. The modeling style or methodology that should be followed (9.1).
- k. Any empirical data that should be directly incorporated into the model (8.2).
- l. Any physical principles that should be incorporated into the model (4.2).
- m. Any test data or test conditions to be used in validating the model (Section 8).
- n. The expected range of inputs and physical environments applied during simulation (6.5).
- o. The model's response to overstress conditions (7.3.2).
- p. Expected warning and error *messages* to be issued by the model (7.3).
- q. Any special *features* desired in the model (5.2).

Each component model shall be validated by means of simulations providing deliverable data. These are intended to characterize the model's performance in the context of the actual component's physical environment as well as the model's simulation environment.