2 Framework for Modeling and Simulation

This chapter is devoted to establishing a Framework for Modeling and Simulation (M&S). The framework defines entities and their relationships that are central to the M&S enterprise. Understanding these concepts will help everyone involved in a simulation modeling project – analysts, programmers, managers, users – to better carry out their tasks and communicate with each other. Terms such as “model” and “simulator” are often loosely used in current practice but have a very sharp meanings in the framework we will discuss. Therefore, it is important to understand what is included and excluded by the definitions. (This is especially true if you have some experience in M&S and are likely to associate (or prefer) meanings that are different from those developed here.) As illustrated in Figure 1, the basic entities of the framework are: source system, model, simulator and experimental frame. The basic inter-relationships among entities are the modeling and the simulation relationships, The entities are defined in Table 1. This table also characterizes the level of system specification that typically describes the entities. The level of specification is an important feature for distinguishing between the entities, which is often confounded in the literature. You can return to Figure 1 and Table 1 to keep an overall view of the framework as we describe each of the components in the following presentation.

Based on this framework, the basic issues and problems encountered in performing M&S activities can be better understood and coherent solutions developed.
2.1  The Entities of the Framework

2.1.1  Source System

The *source system* (we will omit the ‘source’ qualifier, when the context is clear) is the real or virtual environment that we are interested in modeling. It is viewed as a *source of observable data*, in the form of time-indexed trajectories of variables. The data that has been gathered from observing or otherwise experimenting with a system is called the *system behavior database*. As indicated in Table 1, this concept of system is a specification at level 0 (or equivalently, Klir’s source system) and its database is a specification at level 1 (or equivalently, Klir’s data system). This data is viewed or acquired through experimental frames of interest to the modeler.

Applications of M&S differ with regard to how much data is available to populate the system database. In *data rich* environments, such data is abundant from prior experimentation or can easily be obtained from measurements. In contrast, *data poor* environments offer meager amounts of historical data or low quality data (whose representativeness of the system of interest is questionable). In some cases it is impossible to acquire better data (for example, of combat in real warfare); in others, it is expensive to do so (for example, topography and vegetation of a forest). In the latter case, the modeling process can direct the acquisition of data to those areas that have the highest impact on the final outcome.

<table>
<thead>
<tr>
<th>Basic Entity</th>
<th>Definition</th>
<th>Related System Specification Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>source system</td>
<td>real or artificial source of data</td>
<td>known at level 0</td>
</tr>
<tr>
<td>behavior database</td>
<td>collection of gathered data</td>
<td>observed at level 1</td>
</tr>
<tr>
<td>experimental frame</td>
<td>specifies the conditions under which system is observed or experimented with</td>
<td>constructed at levels 3 and 4</td>
</tr>
<tr>
<td>model</td>
<td>instructions for generating data</td>
<td>constructed at levels 3 and 4</td>
</tr>
<tr>
<td>simulator</td>
<td>computational device for generating behavior of the model</td>
<td>constructed at level 4</td>
</tr>
</tbody>
</table>

Table 1  Defining the Basic Entities in M&S and their usual Levels of Specification

2.1.2  Experimental Frame

An *experimental frame* is a specification of the conditions under which the system is observed or experimented with. As such an experimental frame is the operational formulation of the objectives that motivate a modeling and simulation project. For example, out of the multitude of variables that relate to a forest, the set {lightning, rain, wind, smoke} represents one particular choice. Such an experimental frame is motivated by the interest in modeling the way lightning ignites a forest fire. A more refined experimental frame would add the moisture content of the vegetation and the amount of unburned material as variables. Thus, many experimental frames can be formulated for the same system (both source system and model) and the same experimental frame may apply to many systems. Why would we want to define many frames for the same system? Or apply the same frame to many systems? For the same reason that we might have different objectives in modeling the same system, or have the same objective in modeling different systems. More of this in a moment.
There are two equally valid views of an experimental frame. One, views a frame as a definition of the type of data elements that will go into the database. The second views a frame as a system that interacts with the system of interest to obtain the data of interest under specified conditions. In this view, the frame is characterized by its implementation as a measurement system or observer. In this implementation, a frame typically has three types of components (as shown in Figure 2): generator, that generates input segments to the system; acceptor that monitors an experiment to see the desired experimental conditions are met; and transducer that observes and analyzes the system output segments.

**Objectives and Experimental Frames**

Objectives for modeling relate to the role of the model in systems design, management or control. The statement of objectives serves to focus model construction on particular issues. It is crucial to formulate such a statement as early as possible in the development process. A firmly agreed upon statement of objectives enables project leaders to maintain control on the efforts of the team. Once the objectives are known, suitable experimental frames can be developed. Remember, that such frames translate the objectives into more precise experimentation conditions for the source system or its models. A model is expected to be valid for the system in each such frame. Having stated our objectives, there is presumably a best level of resolution to answer the questions raised. The more demanding the questions, the greater the resolution likely to be needed to answer it. Thus, the choice of appropriate levels of abstraction also hinges on the objectives and their experimental frame counterparts.

![Figure 2](image.png)  
**Figure 2** Experimental Frame and its Components.

Figure 3 depicts the process of transforming objectives into experimental frames. Typically modeling objectives concern system design. Here measures of the effectiveness of a system in accomplishing its goal are required to evaluate the design alternatives. We call such measures, outcome measures. In order to compute such measures, the model must include variables, we’ll call output variables, whose values are computed during execution runs of the model. The mapping of the output variables into outcome measures is performed by the transducer component of the experimental frame. Often there may be more than one layer of variables intervening between output variables and outcome measures. For example, in military simulations, measures of performance are output variables that typically judge how well parts of a system are operating. For example, the success of a missile in hitting its target is a performance measure. Such measures enter as factors into outcome measures,
often called *measures of effectiveness*, that measure how well the overall system goals are being achieved, e.g., how many battles are actually won by a particular combination of weapons, platforms, personnel, etc. The implication is that high performing components are necessary, but not sufficient, for highly effective systems, in which they must be coordinated together to achieve the overall goals.

![Figure 3 Transforming Objectives to Experimental Frames.](image)

Forest fire management is an interesting application domain. There are two quite different uses of M&S in this area: 1) that of fighting fires when they break out and 2) that of trying to prevent them from breaking out in the first place, or at least minimizing the damage when they do. Formulated as objectives for modeling these purposes lead to quite different experimental frames. Let’s look at each of these frames.

In the first frame, real-time interdiction, which refers to on-the-spot fire fighting, we require accurate prediction of where the fire will spread within a matter of hours. These predictions are used to allocate resources in the right places for maximum effectiveness. Because humans may be placed in great danger, highly reliable short-term predictions are required. A typical question asked here would be: is it safe to put a team of fire fighters on a particular ridge within reach of the current fire front for the next several hours? To improve the ability to make accurate predictions, the model state may be updated with satellite data to improve its correspondence with the real fire situation as it develops.

In fire prevention and mitigation the emphasis is less on short term prediction of spread than on answering “what-if” questions for planning purposes. For example land use planners might ask what should be the width of a fire break (area cleared of trees) around a residential area bordering a forest so that there is a less than 0.1% chance of houses catching fire. Here the model needs to be capable of working with a larger area of the landscape but the resolution it needs may be considerably less in order for useful comparisons of different planning alternatives to result. Indeed, a model might be capable of rank ordering alternatives without necessarily producing fire spread behavior with high accuracy.
As suggested the experimental frames that are developed for these contrasting objectives, interdiction and prevention, are quite different. The first (interdiction) calls for experimenting with a model in which all known prevailing fuel, wind and topographic conditions are entered to establish its initial state. The output desired is a detailed map of fire spread after say five hours within the region of interest.

The second experimental frame (prevention) calls for a wider scope, lower resolution representation of the landscape in which a range of expected lightning, wind, rain and temperature regimes may be injected as input trajectories. The model may then be placed into different states corresponding to different prevention alternatives, e.g., different fire break spatial regions might be investigated. The output for a particular run might be as simple as a binary variable indicating whether or not the residential area was engulfed by fire. The output summarized over all runs, might be presented as a rank ordering of alternatives according to their effectiveness in preventing fire spreading to the residential area (e.g., the percent of experiments in which the residential area was not engulfed by flame).

2.1.3 Model

In its most general guise, a model is a system specification at any of the levels discussed in Chapter 1. However, in the traditional context of M&S, the system specification is usually done at levels 3 and 4, corresponding to Klir’s generative and structure levels. Thus the most common concept of a simulation model is that it is a set of instructions, rules, equations, or constraints for generating I/O behavior. In other words, we write a model with a state transition and output generation mechanisms (level 3) to accept input trajectories and generate output trajectories depending on its initial state setting. Such models form the basic components in more complex models that are constructed by coupling them together to form a level 4 specification.

There are many meanings that are ascribed to the word “model”. For example, a model is conceived as any physical, mathematical, or logical representation of a system, entity, phenomenon, or process. The definition in terms of system specifications has the advantages that it has a sound mathematical foundation and is has a definite semantics that everyone can understand in unambiguous fashion. Like other formal definitions, it cannot capture all meanings in the dictionary. However, it is intended to capture the most useful concepts in the M&S context.

2.1.4 Simulator

As a set of instructions, a model needs some agent capable of actually obeying the instructions and generating behavior. We call such an agent a simulator\(^1\). Thus, a simulator is any computation system (such as a single processor, a processor network, the human mind, or more abstractly an algorithm), capable of executing a model to generate its behavior. A simulator is typically specified at a high level since it is a system that we design intentionally to be synthesized from components that are off-the-shelf and well-understood. Separating the model and simulator concepts provides a number of benefits for the framework:

- The same model, expressed in a formalism, may be executed by different simulators thus opening the way for portability and interoperability at a high level of abstraction.

---

\(^1\) TMS76 referred used the generic “computer” instead of the more specific “simulator”.
Simulator algorithms for the various formalisms may be formulated and their correctness rigorously established.

The resources required to correctly simulate a model afford a measure of its complexity.

<table>
<thead>
<tr>
<th>Basic Relationship</th>
<th>Definition</th>
<th>Related System Specification Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>modeling relation</td>
<td>concerned with how well model-generated behavior agrees with observed system behavior</td>
<td>comparison is at level 1</td>
</tr>
<tr>
<td>replicative validity</td>
<td></td>
<td>comparison is at level 2</td>
</tr>
<tr>
<td>predictive validity</td>
<td></td>
<td>comparison is at level 3,4</td>
</tr>
<tr>
<td>structural validity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>simulation relation</td>
<td>concerned with assuring that the simulator carries out correctly the model instructions</td>
<td>basic comparison is at level 2; involves homomorphism at levels 3 or 4</td>
</tr>
<tr>
<td>correctness</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2 Primary Relationships Among Entities.

2.2 Primary Relations Among Entities

The entities - system, experimental frame, model, simulator – become truly significant only when properly related to each other. For example, we build a model of a particular system for some objective - only some models, and not others, are suitable. Thus, it is critical to the success of a simulation modeling effort that certain relationships hold. The two most fundamental are the modeling and the simulation relations (Table 2)

2.2.1 Modeling Relation: Validity

The basic modeling relation, validity, refers to the relation between a model, a system and an experimental frame. Validity is often thought of as the degree to which a model faithfully represents its system counterpart. However, it makes much more practical sense to require that the model faithfully captures the system behavior only to the extent demanded by the objectives of the simulation study. In our formulation, the concept of validity answers the question of whether it is impossible to distinguish the model and system in the experimental frame of interest. The most basic concept, replicative validity, is affirmed if, for all the experiments possible within the experimental frame, the behavior of the model and system agree within acceptable tolerance. Thus
replicative validity requires that the model and system agree at the I/O relation level 1 of the system specification hierarchy.

Stronger forms of validity are *predictive validity* and *structural validity*. In predictive validity we require not only replicative validity, but also the ability to predict as yet unseen system behavior. To do this the model needs to be set in a state corresponding to that of the system. Thus predictive validity requires agreement at the next level of the system hierarchy, that of the I/O function level 2. Finally, structural validity requires agreement at level 3 (state transition) or higher (coupled component). This means that the model not only is capable of replicating the data observed from the system but also mimics in step-by-step, component-by-component fashion, the way that the system does its transitions.

The term *accuracy* is often used in place of validity. Another term, *fidelity*, is often used for a combination of both validity and detail. Thus, a high fidelity model may refer to a model that is both high in detail and in validity (in some understood experimental frame). However when used this way, beware that there may be a tacit assumption that high detail alone is needed for high fidelity, as if validity is a necessary consequence of high detail. In fact, it is possible to have a very detailed model that is nevertheless very much in error, simply because some of the highly resolved components function in a different manner than their real system counterparts.

### 2.2.2 Simulation Relation: Simulator Correctness

The basic simulation relation, simulator correctness, is a relation between a *simulator* and a *model*. A simulator correctly simulates a model if it is guaranteed to faithfully generate the model’s output trajectory given its initial state and its input trajectory. Thus, simulator correctness requires agreement at the I/O function level (2). In practice, simulators are constructed to execute not just one model but a family of possible models. This flexibility is necessary if the simulator is to be applicable to a range of applications. In such cases, we must establish that a simulator will correctly execute a particular class of models. Since the structures of both the simulator and the model are at hand, it may be possible to prove correctness by showing that a homomorphism relation holds. Recall from Chapter 1 that a homomorphism is a correspondence between simulator and model states that is preserved under transitions and outputs.

### 2.3 Other Important Relationships

Besides the two fundamental relationships, there are others that are important for understanding modeling and simulation work. These relations have to with the interplay and orderings of models and experimental frames.

**Modeling as Valid Simplification**

The inescapable fact about modeling is that it is severely constrained by complexity limitations. Complexity, is at heart, an intuitive concept – the feeling of frustration or awe that we all sense when things get too numerous, diverse, or intricately related to discern a pattern, to see all at once – in a word, to comprehend. Generalizing from the boggled human mind to the overstressed simulator suggests that the complexity of model can be measured by the resources required by a particular simulator to correctly interpret it. As such, complexity is measured relative to a particular simulator, or class of simulators. However, as we will see in Chapter XXX, properties intrinsic to the model are often strongly correlated with complexity independently of the underlying simulator. While computers continue to get faster and possess more memory, they will always lag behind our ambitions to capture reality in our models. Successful modeling can then be seen as valid simplification. We need to simplify, or reduce the complexity, to enable our models to be executed on our resource-limited
simulators. But the simplified model must also be valid, at some level, and within some experimental frame of interest. As in Figure 4, there is always a pair of models involved, call them the base and lumped models. Here, the base model is typically “more capable” and requires more resources for interpretation than the lumped model. By the term “more capable”, we mean that the base model is valid within a larger set of experimental frames (with respect to a real system) than the lumped model. However, the important point is that within a particular frame of interest the lumped model might be just as valid as the base model. The concept of morphism introduced in Chapter 1 affords criteria for judging the equivalence of base and lumped models with respect to an experimental frame. In Chapter 13, we will discuss methods for constructing such morphisms.

![Experimental Frame Diagram](image)

**Figure 4** Base/Lumped Model Equivalence in Experimental Frame

**Experimental Frame – Model Relationships**

Assume that we have a whole repository of models and experimental frames that have been built up over years of experience. Then it is critical to have an ability to ask whether there are any experimental frames that meet our current objectives and whether there are models that can work within this frame. Only those models have a chance of providing valid answers to our current questions. The relation that determines if a frame can logically be applied to a model is called applicability and its converse, is called accommodation (Table 3). Notice that validity of a model in a particular experimental frame, requires, as a precondition, that the model accommodates the frame.
<table>
<thead>
<tr>
<th>Relationship</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental frame applies to a model (or ‘is applicable to’)</td>
<td>the conditions on experimentation required by the frame can be enforced in the model</td>
</tr>
<tr>
<td>Model accommodates experimental frame</td>
<td>frame is applicable to the model</td>
</tr>
<tr>
<td>Experimental Frame 1 is derivable from Experimental Frame 2</td>
<td>any model that accommodates Experimental Frame 2 also accommodates Experimental Frame 1</td>
</tr>
</tbody>
</table>

Table 3 Other M&S Relationships important when dealing with a model repository.

The degree to which one experimental frame is more restrictive in its conditions than another is formulated in the derivability relation. A more restrictive frame leaves less room for experimentation or observation than one from which it is derivable. So, as illustrated in Figure 5, it is easier to find a model that is valid in a restrictive frame for a given system. It turns out that applicability may be reduced to derivability. To see this, define the scope frame of the model to represent the most relaxed conditions under which it can be experimented with (this is clearly a characteristic of the model.) Then a frame is applicable to a model, if it is derivable from the scope frame of the model. This means that a repository need not support both applicability and derivability queries. Only the latter is sufficient if each model has an associated scope frame.

![Figure 5 Illustrating Important M&S Relations Relevant to Model Repositories](image)

2.4 Time

Implicitly, until now, we have assumed that a time base is an abstract way of ordering observations made on a system. In Chapter 3, we will formally characterize a time base as an ordered set for indexing events that models the flow of actual time. If the interpretation of such a time base is left abstract in this manner, we refer to it as logical time. In contrast, when we consider events happening in the real world, in real time, we refer to a time
variable as measured by an actual clock. Thus, physical time, also called metric time or wall-clock time, is measured by ticks of physical clocks, while logical time is measured by ticks of a clock somehow embedded in a model. Also, as relativity theory made clear, time, as perceived by observers at different locations may be different. Based on this distinction, time can be either local and global. The former is valid only within a component of a system; the latter is valid in the whole system. Thus, there are at least two dimensions for classifying time: one along the logical/physical axis and the other along the local/global axis. Consequently, a time base can be interpreted as falling in any one of the four combinations shown in Table 4.

<table>
<thead>
<tr>
<th>Logical/Physical</th>
<th>Logical Time</th>
<th>Physical Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local/Global</strong></td>
<td>Global Time</td>
<td>Global: All components operate on the same abstract time base.</td>
</tr>
<tr>
<td></td>
<td>Local Time</td>
<td>Local: Logical: A component operates on its own abstract time base.</td>
</tr>
</tbody>
</table>

Table 4  A Time Taxonomy

Traditionally, modeling and simulation has considered mainly the first (global, logical) combination. That is, we assume all components of a modeled system have the same time frame of reference and we consider time as an abstract quantity. However, when a model is executing in a simulator, which may be distributed among computer nodes in a network and may also be interacting with the real world, it is hard to maintain this fiction. We will discuss such real-time and distributed simulation approaches later starting with Chapters 10 (real time) and 11 (distributed). For now, we note that synchronization between time bases requires maintaining a correspondence between the two. For example, a distributed simulation protocol synchronizes the local, logical times maintained by the individual simulator nodes. Another example of synchronization occurs in a real-time, human-in-the-loop simulation-based training. Here the simulator employs a physical time base (e.g., computer system clock) to synchronize between a pilot’s physically perceived time base and the logical time of a model of the aircraft being simulated.

2.5  Summary

With this introductory exposition of the framework and the systems foundation in Chapter 1, the pattern of development underlying the book’s table of contents should be readily apparent. The next two chapters of Part I continue, in an informal manner, to present the basic modeling formalisms and their simulation algorithms. Part
II returns, in a spiral manner, to present the levels of system specification and introduce the modeling formalisms and their simulators in a rigorous, but readable, manner. Part III takes the same approach to the system specification morphisms and goes on to tackle the issues of modeling as valid simplification. A theme that this is strongly emphasized in this edition, (one that could only be dimly perceived in TMS76) is that of representing systems, whatever their native formalism, in discrete event form. Part IV starts the process of examining the implications of the framework and its theory for the methodology of modeling and simulation and the computer environments to support it. Indeed, we leave the reader at a superhighway exit ramp with many country roads to explore, guide book in hand, concepts and theory providing the compass.

2.6 Sources

The framework discussed here derives from the one in TMS76. It was subsequently elaborated and extended in [Zeigler 1984] which contains much more in-depth development than can be provided here. The framework was presented in its current form in the book edited by Cloud and Rainey [Zeigler 1997]. Readers may consult the latter book for an integrated approach to the development and operation of models.
