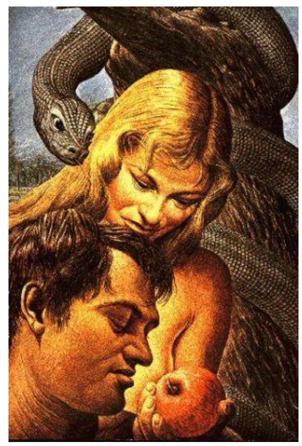
# An Introduction Modeling and Simulation Concepts

Extracted from "Discrete-Event Modeling and Simulation: a Practitioner's approach". G. Wainer. CRC Press. Taylor and Francis. 2009. (Chapter 1)
https://www.amazon.ca/dp/1420053361?tag=duc12-20&linkCode=osi&th=1&psc=1

Human beings are resourceful and curious. Since the dawn of time, we have been interested in solving problems we find around us; sometimes, to improve our quality of life (i.e., by creating devices to reduce effort or improve safety), driven by our curiosity (i.e., learning how plants grow); even just for fun. The need to explore and analyze our surroundings is in our nature (even from very young age). Human kind have been driven by a desire of learning about the universe for thousands of years.



The first technique humans created to learn about (and maybe change) their environment was *experimentation* (for instance, if one wants to learn how much clay and water must be mixed to do pottery, one does different trials until the desired consistency is obtained).


Figure 1 shows a basic scheme for experimentation, in which an experimental frame (for instance, the different trials – experiments – of mixing clay and water) is carried out on an entity (a clay mix that we can mold and solidify in an oven).

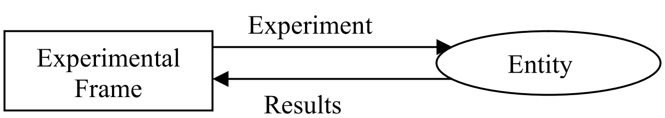



Figure 1. Problem solving through experimentation.

There are two objects under consideration: the *entity* under study (the mix of water and clay), and the Experimental Frame (EF), which defines the conditions for experimentation (percentage of each material, temperature of the mix, maximum amount of time of the experiment, etc.). The EF defines not only how we experiment on the entity, but how we obtain the experimental results (i.e., the expected consistencies and textures expected for the clay mix).

*Example 1:*

> Let us suppose that we want to study the best possible allocation of desks in a classroom, in order to reduce heating costs. We need to decide where to put the desks and the heating/air conditioning sources. An experimentation-based solution would take different groups of students in different positions, and would use a sensor (i.e., a thermometer) to take the temperature in different areas in the classroom. In this case, the Experimental Frame will define by multiple student configurations (Experiments), and the kind of Results expected at the end of each experiment. The Entity is the classroom under study with different desk configurations. The Results are provided by the thermometers used to measure the temperature.

Unfortunately, the problems we need to tackle are usually much more complex than learning how to mix the materials for doing pottery or measuring the number of students and their temperatures in a classroom. In many cases, experimentation is not a feasible solution due to risks (i.e., we cannot study spread of an epidemics or fire evacuation in the classroom) or costs (we cannot study every possible configuration in the classroom, as having a large number of individuals during a long time can be very expensive). In other cases, experimentation is simply not possible (for instance, we cannot manipulate a star to better understand its gravitational field; we also cannot do an experiment on our classroom if the room still does not exist; and in many cases this is a need: we want to carry out our studies before building the room).


*Problems with Experimentation: Ethics; Cost (pictures: Large Haldron Collider; Sudbury Neutrino Observatory), a combination of them (Atomic bomb explosion), impossibility to experiment (black hole, subatomic particles, remote planets)*

Human ingenious found different ways of dealing with these issues. One of them is very old, and it is based on a concept that (if I caught your attention with the classroom example), you might have been doing yourself. As soon as we think about an experiment like Example 1, we start abstracting the problem itself, and we create a model of the problem in our minds. As soon we deal with the entities under study (like our classroom example), one starts having a mind picture of the problem (thinking about possible student distributions, mechanisms to improve the heating according to where they are located, window positions, doors, corridors, building orientation, and room location in the building). You might have even gotten pen and paper to organize your ideas better.

We are well prepared (and have been trained from childhood) to create these mind *models* in a very natural way, and they help us to better think about the problem we want to study. Modeling techniques evolved for centuries, and as of today, we have over 300 years of experience based on the ideas proposed by Newton-Leibniz. *Differential equations*, for instance (Taylor 1996) provide a formal framework to study and analyze problems that has proven to be extremely successful. Using these equations, we could think about the room's temperature using Fourier's law (Bacon 1989), which defines the conduction of heat in a one-dimensional steady state isotropic as $q_x = -k\frac{\partial T}{\partial x}$, where $q$ is the heat flux (W/m$^2$), $q_x$ is the heat flux, $k$ is the thermal conductivity of the material under study (W/m ºC), T is the temperature field in the medium, and $\partial T / \partial x$ is the temperature gradient (the minus sign indicates that the direction of heat flux is opposite to direction of increasing temperature). When we try to solve problems through modeling (using, for instance, differential equations), the scheme previously presented in Figure 1 must be extended as follows:

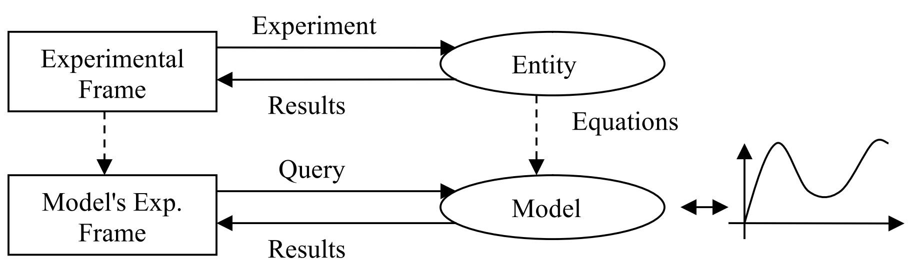
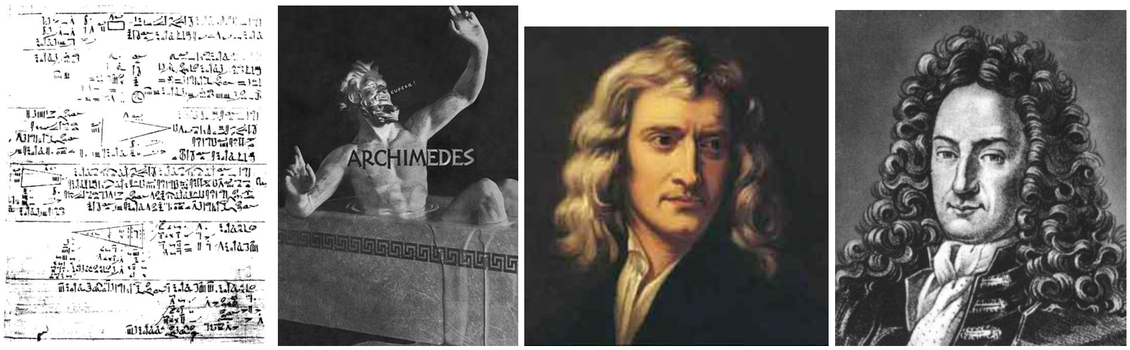


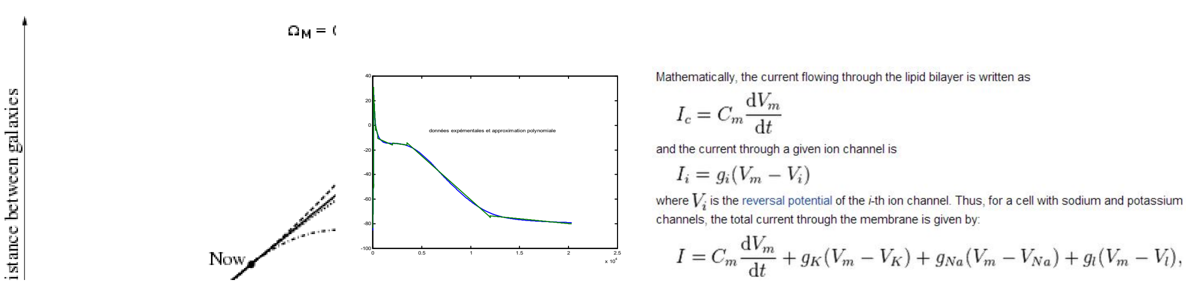

Figure 2. Problem solving through Analytical Modeling.

In this case, we still do some experimentation to obtain data about the entity under study, but the immediate step is to create a model of the entity through the use of equations (or any other analytical technique). We use the Model's Experimental Frame to put a context on the model's creation. In this case, the Model's EF will indicate the kind of questions we can query to the model (for instance, we cannot find the average weight of the students using the Fourier's equations we have defined for finding the room's temperature), and it gives a context for the assumptions one might take (i.e., we will use a 1D equation to approximate the temperature in the room). The Model's EF also tries to mimic the experiments carried out with the original entity (i.e., we are sensing of temperatures, and we are not interested in the social interactions between the class students), in order to be able to obtain the desired Results (in this case, the effect in the room's temperatures according to the number of students and their distribution). The results observed are used, in turn, to modify the original entity (in this case, temperature).




*Formal modeling in history: A'h-mosè papyrus (1650 BC), Archimedes (250 BC), Newton-Leibniz (1670 AD)*

These problem-solving techniques are analytical, in the sense that they are symbolic and based on reasoning, and they try to provide general solutions to the problems to be solved. The idea is that we abstract what we learned about the entity through experimentation (in this case, the temperature in the room) into a Model that represents the Entity under study. This abstraction implies a loss of information, but simultaneously, it allows one to describe the behavior of the entities, and to prove properties of them (for instance, controllability and stability in a control system). We use equations to represent the entity and the experiments, and if we are able to solve them, we will know the results for every possible experiment to be carried out. The solution is built using inference rules (which should be correct in the paradigm chosen to describe the model). If we need to obtain particular solutions, we just replace the symbolic values by their corresponding numerical counterparts. Once obtained a solution, one can apply a value to each of the variables of the model obtained, and one can find particular solutions. In Example 1, the Fourier's equation allows us to find the solution to the problem for every value of *x*; if we are able to solve the equation, we will know the exact temperature in every single point (with infinitesimal scale).




*Analytical modeling: finding the equations ("Is the Universe ending? How?). Solving the equations found (Hodgkin-Huxley model of Action Potential in neurons)*

Figure 2 shows an important concept: the model and the original entity should match. That is, the results given by the model should be valid. This requires a *Validation* phase, in which one must check that the results given by the model match what we see in the original entity. This is one of the most complex tasks in our study, moreover considering that throughout history, many theories have been developed by creating abstract mathematical models based on analytical models that had no available entity to experiment with (i.e., atoms, quarks, chemical reactions in

ADN or black holes). In some cases, the model was built first, and only much later technology permitted to build experimentation facilities to validate the proposed theories, which in some cases were correct, but in others required rework based on the experimental results.

Although in many cases we can solve these equations, we usually need to do several simplifications to be able to succeed. In our Example 1:

a. We use a single-dimension model, and we are interested in a 3D model. Adding a second equation to solve the problem in 2D (which is yet a simplification of the 3D problem), adds extra complexity. In this case, we need to add a new equation $q_y = -k \frac{\partial T}{\partial y}$, and modify the temperature field in the medium $T = T(x, y)$ to be a function of both $x$ and $y$. Obviously, solving this equation is more complex.
b. These equations do not consider transient behavior: what happens if students move? What if a window is opened? What if the heating is turned on? How the position of the heating device and the flow of heat from it are going to affect the equation definition?
c. The equations do not consider combinations of the different possible transients, the materials being used in the room's walls and floors, influence of air flowing in the room, and many other simplifications.
d. We need to be able to *solve* the equations of interest, which can be unfeasible or very complex (moreover considering that, in many cases, just finding the equations to describe the entity under study in detail is impossible).

If one is interested in solving a simple (and harmless) problem like the one in Example 1, most of the details in reality will probably be ignored. But if, instead of the temperature of a classroom we are interested in studying the heat conductivity of a new material with manufacturing purposes, we need to pay attention to these details (otherwise the model will lose precision, and the results of the study would be incorrect).

Thus, in order to deal with higher complexity, *difference equations* and other *numerical methods* were introduced (Brenan, Campbell, and Petzold ; Lapidus and Pinder 1982). The idea of these methods (which are almost as old as the differential equations) is to approximate the equation by discretizing the time evolution (which, instead of being continuous and thus computable at every point in time, it is now calculated at predefined timesteps). The result is numerical, and it will lose precision (as we cannot obtain solutions for every possible combination of the model's variables), but it will provide approximate values that are close enough for the problem under study. For instance, in our previous example, we could divide the area of the model in small elements (assuming a linear temperature distribution along its unit length, and a unit area perpendicular to heat flow direction), and obtain the approximated result as:

$$T_1 = \frac{hT_\infty + \frac{K_1}{L_1} T_o}{h + \frac{K_1}{L_1}}$$

where $K_l$ is the thermal conductivity, $L_i$ the length of the element, $T_\infty$ is the fluid temperature, $T_o$ is the inner node temperature inside the material, and $h$ is the heat transfer coefficient. $T_1$ is the surface temperature computed, which will be used in the next computation as $T_o$ (Bacon 1989). The problem solving activities can now be described as follows:

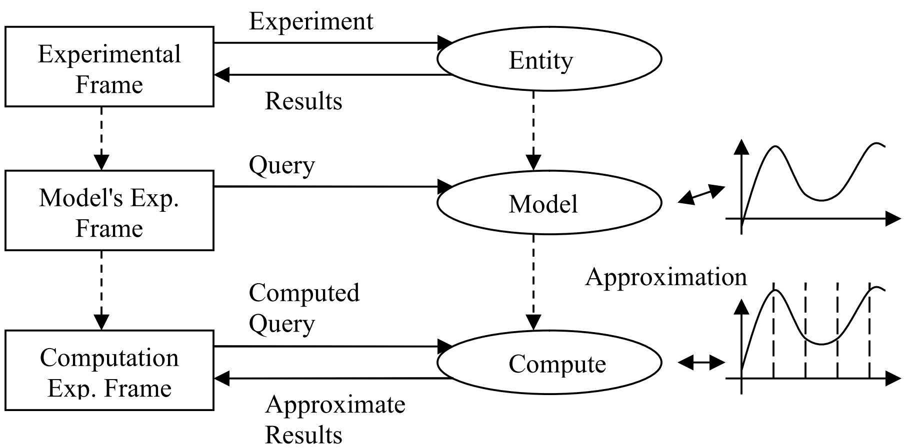



Figure 3. Problem solving through Computation

In this case, we also obtain experimental data from the entity, and then we create the equations to model the observed behavior (within the corresponding experimental frames). Nevertheless, as in this case, we cannot find a solution for the equations, we use a numerical approximation. The computation is based on a recursive method (that will calculate the values of the state variables at a given time, and it will convert the value as the basis for the next computation). The Computation EF in this case is derived from the Model's EF, and within the same frame, the computation model should be able to answer the same queries (with a lost of precision due to the approximation). In this figure a new step is introduced, which means we need to carry out an extra check in order to verify the correctness of the results. In this case, we will say we need to do *Verification* of the results obtained by numerical approximation. The idea is that the values we computed need to be checked against the experimental data, while trying to mimic the model's description as accurately as possible.

This was the main approach of problem in the last few centuries, which gave excellent results for every field in science and engineering. In fact, this model made science and technology advance tremendously. Although the associated mathematical techniques evolved, the complexity of the problems to tackle also grew steadily. One of the main reasons is that many of the unanswered questions to problems in nature were solved through these methods, and thus new questions arose (which were, of course, much more complex). Likewise, the elaborate devices humans developed during the 20th century (e.g., automated digital control, intelligent manufacturing, traffic monitoring, etc.) made it difficult or impossible to continue using "pencil-and-paper" methods for their analysis. When we consider such problems (with a few exceptions) they are analytically intractable and numerically impossible to evaluate (unless one simplifies the model, which, in most cases, results in solutions that are far from reality).

The advent of computers in the 1940's provided scientists and engineers with alternative methods of analysis. Computers are well suited to deal with approximation techniques, reducing human compute errors, and being able to solve the problems at much higher speed. Thus, since the early days of computing, numerical models were converted into computer-based solutions (called *computer simulation* studies). The cycle for creating simulation studies can be also represented as in Figure 3, with the difference that now, the computation model is carried out by a specialized device (in the beginning of simulation history, analog computers that could be used for numerical approximation, (Coward ); today, digital computers). The Computation EF is now a program that generates test cases for the computational approximation of the model. In this case, the verification activities will also check the accuracy of the results, while adding an extra step (as limited precision in computers can create erroneous results which can even diverge from the expected solutions).

Computer simulation enabled scientists and engineers to easily experiment with "virtual" environments, elevating the analysis of natural and artificial applications to a new level of detail unknown in earlier stages of scientific development, and providing great help in the design and analysis of complex applications. Simulated models also can be used for training, as they provide cost-effective and risk-free solutions. In simulation-based techniques, one finds solutions to a particular problem (opposed to the general solutions found by analytical methods) using a device (in general, a digital computer) for controlled experimentation and time compression. The constant reduction of the cost of computers (in hand with graphical interfaces, advanced libraries, languages and other facilities), allowed simulation to become an easy-to-use and flexible technique. Nowadays, M&S is a well-developed, well-proven approach to problem solving, which advances steadily as more computing power becomes available at less cost.

The advantages of simulation are multiple:
- one can reduce the development time of a given application,
- decisions taken can be checked artificially,
- the same model can be reused multiple times,
- simulations are easier to create and use than many analytical techniques, and they need less simplifications,
- during simulation studies, one can easily modify the rules used to define the model's behavior,
- during execution one can experiment varied special cases that cannot be carried out in a lab,
- the user can interact with the simulator, allowing analysis of such interactions,
- shorter design-cycle times and reduced requirements for initial resource investment,
- economic benefits: companies and institutions can improve their research and development cycles,
- the entity under study is not affected, and it can continue to be used; etc.

### *1.5. Some definitions*

According to Systems Theory, a *system* is a natural or artificial entity, real or abstract, which is a part of a given reality constrained by an environment. It can be seen as an ordered set of related objects that evolve through different activities, interacting to achieve a goal. It is also called the *real system* (or the *System of Interest*), and it is seen as a source of observational data, which is viewed through an *experimental frame* (EF) of interest to the modeler (Zeigler, Praehofer, and Kim 2000). The system's observational data is used to define the model's *behavior*, which is defined as a specific form of data observable in a system over time within an experimental frame.

A *model* is an understandable representation (abstract and consistent) of a given system that we use to better understand it. Models can be built in a variety of ways, and they have different meaning according to the individual doing the modeling. For Architects, a model can be the drawing of a floor map or a maquette; for a Chemist, a 3D representation of a molecule, etc. In this book, we are interested in models that have *dynamic behavior* (i.e., models that models that exhibit time-varying changes). We are not interested in static models: instead, we want to study how the model organizes itself over time in response to imposed conditions and stimuli. The objective is to predict how the system will react to external inputs and proposed structural changes. The model's behavior is generated using specific rules, equations or a modeling formalism with the goal of generating behavior that should indistinguishable from the one of the system within one or more model's EF. The EF defines the conditions under which a system or a model are observed or experimented with, thus, problem solving is related to an experimental frame in which the model is analyzed (Zeigler, Praehofer, and Kim 2000).

The process of thinking and reasoning about a system in order to abstract the description of the model from reality is called systems *modeling.* We call *paradigm* to the concepts, laws and mechanisms used to define a set of models. It is important to keep a clear separation between the system of interest and the models that we use to think about them (a model is an abstract representation of the system, not the system itself, although it is easy to mix them as we always think about models when reasoning about the real system).

As we are interested in discrete-event modeling and simulation, we need to precisely define some basic concepts in the field (Nance 1981). An *event* is a change in the state of the model, which occurs at a given *instant* (called the *event time*), causing the model to *activate*. The model's activation will produce a *state* change (i.e., at least one attribute in the model will change). Finally, the model's *state* is the set of values of all the attributes of the model at a given instant.

A model is an *abstract* representation of the system of interest, as the model is an aggregated elaboration of the information provided by the system (with a format based on rules, equations, and relations between components). We define *abstraction*: as the basic process we use when modeling in order to extract a set of entities and relations from a complex reality (Zeigler, Praehofer, and Kim 2000). When one abstract from the system, information is lost, but a higher level of abstraction allows one to better define the model's behavior and to prove properties of the system by manipulating the abstract model definition.

We can now say define *simulation* as the reproduction of the dynamic behavior of a System of Interest with the goal of obtaining conclusions that can be applied to the system.

A *simulation study* refers to a carefully designed set of simulation-based experiments. Typically the model associated with a simulation study is implemented as a computer program (also called a *simulation model* or a *simulator*). The *simulation experimental frame* contains information about the experimental conditions, parameter values, and model behavior generation mechanisms. A simulator may interface with real-world entities (e.g., a moving platform for a driving simulator) or with humans (e.g., with a driver in the simulator). We call the first a simulator with *hardware-in-the-loop*, and the second a simulator with *human-in-the-loop.*

In order to be able to create a model and a simulator that represent the system with precision, we need to carry out Verification and Validation (V&V) activities. We will call *Validation* to the relationship between the model, the system and experimental frame. Model's Validation answers the question of whether it is impossible to distinguish behavior of model and system within the experimental frame. *Verification*, instead, is the process of verifying that a simulator of a model correctly generates the desired behavior.

*Example 8:*

Let us employ Example 1 to explain some of the definitions just presented. The *experimental frame* consists of different classroom configurations (students entering/leaving and varied heating/cooling levels). We created different models: a *spatial* one in **Error! Reference source not found.**, and a *declarative* one (finite state machine) in **Error! Reference source not found.**. A student arriving is an *event*, which causes the temperature to raise and to have one more student in the room (these two attributes change, and the composite of both is the room's model's *state change*), which occurs at the *instant* when the student enters the room (represented by a Real number). The FSM and the special models introduced are *abstract* representations of the actual system (for instance, the FSM does not say anything about the student's age or gender, which are attributes of the real system that we are not interested in modeling in our EF). A *simulator* for this model would implement the FSM in software, and a simulator's EF would generate a variety of independent tests to run the program. In order to *verify* the simulator, we need to check that the trajectories generated by the simulator coincide with those in the model (for instance, if a new student arrival is simulated, we see the number of students increment, and the temperature increasing, as described in the model). Finally, *validation* activities should ensure that what we see in the model and in the simulation results coincide with reality (for instance, if we see temperatures decreasing when a new student arrives, we know that this behavior is not valid; we have first to check the model, and if its specification is erroneous, we need to fix it. Otherwise, we need to fix the model's simulator).

Once the study has finished, and the final results are obtained, we need to use them on the original system of interest. According to the objectives for the study, and the decisions to be taken on the system of interest once it has finished, we can consider the following kinds of simulation models:

- *Exploration*: they are used to better understand the operation of the system;
- *Prediction*: they are used to predict the future behavior of the system.
- *Improvement*: they are used to optimize the performance of the system, through the analysis of different alternatives;
- *Conception*: the system does not exist yet, and the model is used to test different options prior construction.
- *Engineering design:* they can be used to design devices in engineering applications (ranging from bridges to electron devices). Simulation permits exploring different design options and help to choose the best.
- *Rapid prototyping:* they permit to quickly obtain a working model that can be used to test ideas and get early feedback from stakeholders.
- *Planning:* they provide a risk-free mechanism for thinking about the future in different fields of application (ranging from manufacturing to governance).
- *Acquisition*: very large pieces of equipment (i.e., helicopters, airplanes, submarines) are extremely expensive. M&S can help to decide in the purchasing process, enabling the customer to exploring different alternatives without the need of constructing the equipment prior to take the decision.
- *Proof of concept*: they can be used to test ideas and put them to work before creating the actual application.
- *Training*: they can be used with training purposes, providing controlled experiments to enhance the decision making skills in defense (called *constructive* simulation), including also emergency management. In these cases, simulation can be used to teach fundamental concepts and enhance relevant decision making abilities. Other training examples include *business gaming* and *virtual* simulators (which are usually human-in-the-loop simulators to learn and enhance motor skills when operating complex vehicles).
- *Education*: they can be used in different sciences to provide insight into the nature of dynamic phenomena as well as the underlying mechanisms.
- *Entertainment*: games and animations are the two most popular applications of simulation.

### *1.1 Phases in a simulation study*

There has been different kinds of lifecycles proposed for studies in modeling and simulation (Balci 1994; Law and Kelton 2000; Lutz 1999; Sargent 2005; Zeigler, Praehofer, and Kim 2000). In this section we include the basic steps that should be considered in doing a simulation study. The life cycle does not have to be interpreted as strictly sequential; it is iterative by nature and sometimes transitions in opposite direction can appear. Likewise, some of the steps can be skipped according to the complexity of the application. It is highly recommended to use a spiral cycle with incremental development for steps a) – h), which can cause going back to earlier phases.

a) Problem formulation: the simulation process begins with a practical problem needed solving or understanding. It might be the case of a cargo company trying to develop a new strategy for truck dispatching, or an astronomer trying to understand how a nebula is formed. At this stage, one must understand the behavior of system of interest (which can be a natural or artificial system, existing or not), organizing the system's operation as objects and activities within the experimental framework of interest. Then, one needs to analyze different alternatives of solution, investigating other previously existing results for similar problems. The most acceptable one should be chosen (omitting this stage could cause the selection of an expensive and/or wrong solution).

One also must identify the input/output variables and classify them into **decision** variables (controllable) or **parameters** (non-controllable). If the problem involves performance analysis, at this point is when one can also define performance **metrics** (based on the output variables) and an **objective function** (i.e., a combination of some of the metrics). At this stage, one can also do risk analysis and decide if it is worth it to follow or discard the project.

b) The *conceptual model* must be defined. This step consists on building a high level description of the structure and behavior of the system, identifying all the objects with their attributes and interfaces. One also must define what are the state variables, how are they interrelated and which ones are important. In this step, key aspects of the requirements are expressed (if possible, using a formalism, which introduces a higher degree of precision). During the definition of the conceptual model, on needs to reveal features that are of critical significance (e.g., possibility of instability, deadlock or chaotic behavior). One must also document non-functional information, for instance, possible future changes, non-intuitive (or non-formal) behavior, and the relation with the environment. At this point, one also must attack the complexity of the system by partitioning it into subcomponents.

c) Collection and analysis of input/output data: in this phase, one must study the system to obtain input/output data. To do so, one must observe and collect the attributes chosen in the previous phase. When the system entities are studied, one tries to associate them with a timing value. Another important issue during this phase is the selection of a sample size that is statistically valid, and a data format that can be processed with a computer. Finally, one must decide which attributes are treated as stochastic and which as deterministic. In some cases, there are no data sources to collect (for instance for non-existing systems). In those cases, one need to try to obtain data sets from similar systems (if available), or to use a stochastic approach trying to cover the data generation through random number generation.

d) Modeling: in this phase one must build a detailed representation of the system including based on the conceptual model and the I/O data collected. The model is built defining objects, attributes and methods using a chosen paradigm. At this point, a specification model is built, including the set of equations defining behavior/structure. Having finished this definition, one must try to build a preliminary structure of the model (possibly relating the system variables and performance metrics), carefully describing any assumptions and simplifications, and collecting them into the model's EF.

e) Simulation: during this stage one must choose a mechanism to define the model (using a computer and a given language and tools), and a simulation model is constructed. During this stage it might be necessary to define simulation algorithms and to translate them into a computer program. In this phase, one also must build a model of the EF for the system.

f) V&V: during the previous steps, three different models are built: the conceptual model (specification), the system's model (design) and the simulation model (executable program). We need to verify and validate these models. Verification is related with the internal consistency between the three models, and verification activities try to answer the question: did we implement the model correctly? Validation, instead, is focused on the correspondence between model and reality. Validation activities try to respond the question: is the implementation consistent with the system being analyzed? Did we build the right model? Based on the results obtained during this phase, the model and its implementation must be refined. As we will discuss on the next section, the V&V process does not constitute a particular phase of the life cycle, but it is an integral part of it. This process must be formal and must be documented correctly since later versions of the model will require another round of V&V, which is, in fact, one of the most expensive phases in the cycle.

g) Experimentation: one must execute the simulation model, following the goals stated in the conceptual model. During this phase one must evaluate the outputs of the simulator, using statistical correlation to determine a precision level for the performance metrics. The phase starts with the design of the experiments, using different techniques: sensitivity analysis, prediction, optimization, determination of functional relations, response-surface methodologies (to find sets of data that maximizes or diminishes output), factorial designs (which determine the effect of input variables on output variables), variance reduction (to optimize the results from a statistical point of view), ranking and selection (comparison with alternative systems), etc.

h) Output Analysis: in the last phase, the simulation outputs are analyzed in order to understand the system behavior. These outputs are used to obtain responses about the behavior of the original system. At this stage, visualization tools can be used to help with the process. The goal of visualization is to provide a deeper understanding of the real systems being investigated, and to help in exploring the large set of numerical data produced in the simulation execution, which is a concern for model validation.

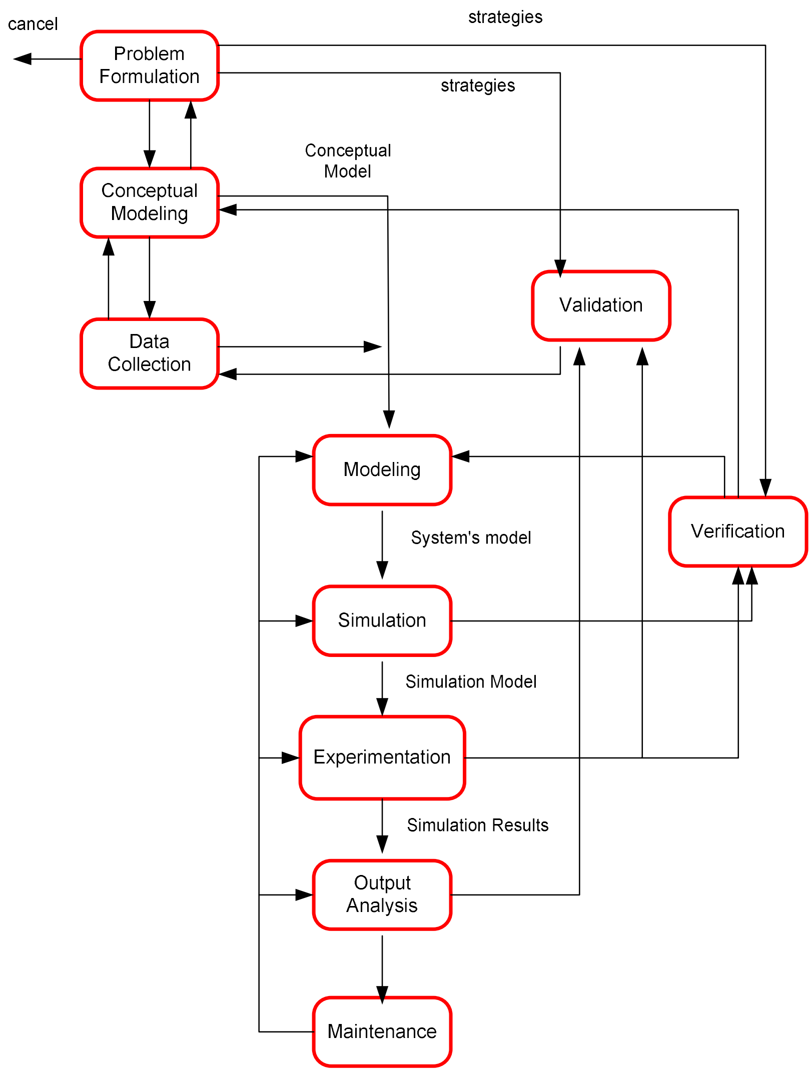



Figure 4. Steps in M&S Studies.

The rest of this book will concentrate mostly on phases d), e) and f), using the DEVS formalism as the chosen paradigm. We will explain how to do advanced visualization for step h), and we will show how to conduct some experiments (g). The focus here is on the practitioner's point of view, and those interested in understanding the details of the underlying theories, should consult (Law and Kelton 2000; Lutz 1999; Zeigler, Praehofer, and Kim 2000). Input/output analysis is exhaustively covered in (Banks, Carson, and Nelson 1996; Cassandras 1993, 790; Kleijnen 2005; Law and Kelton 2000; Leemis and Park 2005) and others. Likewise, we will not cover random number generation (as it is exhaustively covered in the previous references). We will focus on how to create an advanced simulation toolkit based on DEVS modeling and simulation methodology, concentrating on how to reduce the development costs of a simulation (by merging phases b) and d), and reducing phase e), while providing good facilities for incremental development, V&V, experimentation and maintenance.

### *1.2 Verification and Validation (V&V)*

The credibility of the results of the simulation not only depends on the correctness of the model, but also on how accurate the formulation of the problem is expected to be (Pace 1993). Thus, one must use various V&V techniques throughout all the life cycle of the simulation study. As discussed earlier, we call Validation to the process of determining that a model is a correct representation of the real world and its behavior corresponds to the requirements of the model (i.e., validation is related to the formulation of the correct model). We say a model is valid when it is impossible to distinguish system and model within the experimental frame. We can recognize different types of validity (Zeigler, Praehofer, and Kim 2000)

- *Replicative validity*: for every possible experiment within the experimental frame, trajectories of model and system agree within acceptable tolerance.
- *Predictive validity*: It is affirmed if, within an experimental frame, it is possible to initialize the model to a state such that, for the same input trajectory, model output trajectory predicts system output trajectory within acceptable tolerance
- *Structural validity:* the model is able to replicate the data observed from the system, but also mimics in step-by-step, component-by-component the way in which the system does its transitions.

Verification, instead, is the process of checking that a simulator of a model correctly generates its behavior according to the model's specification. A *correct simulation* faithfully generates model behavior in every simulation run.

In large organizations (mostly related to Defense), an *Accreditation* phase can be required. This is the official process of gaining approval for model use, which certifies that the model satisfies the specifications. *Specific* accreditation includes the model and its environment (input data, aptitude of the personnel, performance of the simulator, etc.).

The result of the activities of V&V must not be considered in absolute form (i.e., absolutely right or wrong). A simulation model is constructed with certain objectives described in the Conceptual Model and the Experimental Frame. As in any software project, it is recommended that V&V should be executed by an independent team to prevent biased decisions (Pace 1993). Likewise, the software cannot be tested completely, and standard testing techniques should be used, trying to cover the largest percentage of cases within the domain (Beizer 1990). The objective is to increase confidence in the model, based on the study of the objectives. Likewise, the satisfactory test of each submodel (unit test) does not imply the correctness of the whole (integration test).

Following the lifecycle described in Figure 4, we need to carry out the following activities:

a) Verification of the Conceptual Model and of the problem posed: one must determine that the problem one wants to solve corresponds to the real problem, demonstrating that the characteristics of the system have been identified and its objectives have been defined. It is essential to understand the requirements of the system under study, as they form the base upon which the finished product is evaluated. One also must justify that all the assumptions made during the formulation of the conceptual model are appropriate, and that it represents the real system based on the proposed objectives.

b) Verification of the design: the purpose of design verification is to ensure that the specifications reflect accurately the conceptual model.

c) Validation of the system's model: one must verify that the system's model corresponds with the real system within the EF, and it represents the conceptual model appropriately. One must also verify that the model is represented with sufficient degree of accuracy. As part of this process, one must ensure that the data used in the different phases of the model have the appropriate format and range, and they are complete and unbiased.

d) Verification of the simulator: one must ensure that the software implementation of the model reflects its specification. The times and the restrictions of the process and tests are examined and the behavior of the model is tested to cover the maximum percentage of test cases possible.

e) Validation of the experimental results: the credibility of the model is the result of having completed satisfactorily each one of the different activities of V&V mentioned previously, and that the results given by the simulator are compatible with those of the real system.

There is a variety of techniques employed, discussed in detail in (Balci 1994; Beizer 1990; Kleijnen 2005; Pace 1993; Sargent et al. 2000), which can be categorized as:

a) Informal: they lack mathematical foundations and they are based mainly on human reasoning. These are the most common techniques, including inspections, Turing test (i.e., making an expert to interact with the simulator, trying to make sure the expert cannot recognize the differences between the simulator and the real system) and animations (in which the behavior of the model can be represented using animated graphics). Other informal techniques are the comparison with other models (i.e., the experimental results are compared with the results from other existing models), or the opinion of experts.

b) Static: they are used to validate the design of the model and the source code (they do not execute the model). These techniques include the analysis of data flow, analysis based on graphics, structural, syntactic and semantic analysis.

c) Dynamic: they require of the execution of the model and validate its behavior. Some of the techniques include the symbolic execution of the model, cause-effect graphs, regression tests, and stress testing. It also includes reachability analysis, degenerative testing (in which inaccurate or limit input data are used to test the model), sensitivity analysis (which consists of changing the values of the input as well as those of some internal parameters in order to observe the reaction of the model to these variations), and varied statistical techniques (including graphical comparisons with histograms and plots, confidence intervals, or hypothesis tests).

d) Formal: they try to prove the correctness of the model, using mathematical foundations. They present a high degree of difficulty and serve as a base for other techniques of V&V. This can include induction techniques, inference models, deduction logic, predictive calculus, correction proofs, bisimulation and other model checking techniques.

## References


Alur, Rajeev, and David L. Dill. 1994. A theory of timed automata. *Theoretical Computer Science* 126, no. 2:183-235.

Bacon, D. H. 1989. *BASIC heat transfer.* London ; Boston: Butterworths.

Balci, Osman. 1994. *Validation, verification, and testing techniques throughout the life cycle of a simulation study.*Orlando, Florida, United States ed. San Diego, CA, USA: Society for Computer Simulation International.

Banks, Jerry, Carson, John S., and Nelson, Barry L. 1996. *Discrete-event system simulation.*2nd ed. Upper Saddle River, N.J.: Prentice Hall.

Beizer, Boris. 1990. *Software testing techniques (2nd ed.).* New York, NY, USA: Van Nostrand Reinhold Co.

Belina, F., and D. Hogrefe. 1989. The CCITT-specification and description language SDL. *Comput.Netw.ISDN Syst.* 16, no. 4:311-341.

Birtwistle, G. M. 1979. *A system for discrete event modelling on SIMULA.* London: Macmillan.

Brenan, K. E., S. L. Campbell, and L. R. Petzold. Numerical Solution of Initial-Value Problems in Differential Algebraic Equations, Elsevier, North Holland, New York, N.Y.; 1989.

Buxton, J. N., IFIP Technical Committee 2--Programming, and IFIP Technical Committee 2, Programming. 1968. *Simulation programming languages : proceedings of the IFIP Working Conference on Simulation Programming Languages.* Amsterdam: North-Holland Pub. Co.

Cao, X. -., and Y. -. Ho. 1990. Models of discrete event dynamic systems. *IEEE Control Systems Magazine* 10, no. 4:69.

Cassandras, Christos G. 1993. *Discrete event systems : modeling and performance analysis.* Homewood, Ill.: Aksen : Irwin.

Cellier, François E., and Kofman, Ernesto. 2006. *Continuous system simulation.* New York: Springer Science+Business Media.

Coward, Doug. Analog Computer Museum and History Center. [cited September 1 2007]. Available from http://dcoward.best.vwh.net/analog/.

Fishwick, Paul A. 1995. *Simulation model design and execution : building digital worlds.* Englewood Cliffs, N.J.: Prentice Hall.

Glynn, P. W. 1989. A GSMP formalism for discrete event systems. *Proceedings of the IEEE* 77, no. 1:14-23.

Grattan-Guinness, I. 2000. *The search for mathematical roots, 1870-1940 : logics, set theories and the foundations of mathematics from Cantor through Russell to Gödel.* Princeton, N.J.: Princeton University Press.

Harel, David, Politi, Michal, and Books24x7, Inc. 1998. *Modeling reactive systems with statecharts.* New York: McGraw-Hill.

Ho, Y. 1989. Special issue on discrete event dynamic systems. *Proceedings of the IEEE*.

Hoare, C. A. R. 1985. *Communicating sequential processes.* Englewood Cliffs, N.J.: Prentice/Hall International.

Hopcroft, John E., Motwani, Rajeev, and Ullman, Jeffrey D. 2007. *Introduction to automata theory, languages, and computation.*3rd ed. Boston: Pearson/Addison Wesley.

Kim, T. G. 1994. DEVSIM++ User's Manual. *, CORE Lab, EE Dept, KAIST, Taejon*.

Kleijnen, Jack P. C. 2005. *Forty years of statistical design and analysis of simulation experiments (DASE).*Orlando, Florida ed.Winter Simulation Conference.

Kofman, Ernesto. 2004. Discrete Event Simulation of Hybrid Systems. *SIAM J.Sci.Comput.* 25, no. 5:1771-1797.

Lapidus, L., and G. F. Pinder. 1982. Numerical Solution of Partial Differential Equations in Science and Engineering. *Wiley*.

Law, Averill M., and Kelton, W. D. 2000. *Simulation modeling and analysis.*3rd ed. Boston: McGraw-Hill.

Leemis, Lawrence M., and Park, Stephen K. 2005. *Discrete-Event Simulation: A First Course.*{Prentice Hall}.

Lutz, Robert. 1999. *FEDEP V1.4: an update to the HLA process model.*Phoenix, Arizona, United States ed. New York, NY, USA: ACM Press.

Manna, Zohar, and Pnueli, A. 1992.
*The Temporal Logic of Reactive and Concurrent Systems.*Springer-Verlag.

Manna, Zohar, and Pnueli, A. 1987. *A hierarchy of temporal properties.* Stanford, CA: Dept. of Computer Science, Stanford University.

Manna, Zohar, Pnueli, A., and Stanford Artificial Intelligence Laboratory. 1973. *Axiomatic approach to total correctness of programs.* Stanford, Calif.: Computer Science Dept., Stanford University.

Manna, Zohar, Pnueli, A., and Stanford University. Computer Science Dept. 1996. *Clocked transition systems.* Vol. STAN-CS-TR-96-1566. Stanford, Calif.: Dept. of Computer Science, Stanford University.

Milner, Robin. 1980. *A calculus of communicating systems.* Vol. 92. Berlin ; New York: Springer-Verlag.

Nance, Richard E. 1981. The time and state relationships in simulation modeling. *Communications of the ACM* 24, no. 4:173-179.

Pace, Dale K. 1993. *Naval modeling and simulation verification, validation, and accreditation.*Los Angeles, California, United States ed. New York, NY, USA: ACM Press.

Papadimitriou, Christos H. 1994. *Computational complexity.* Reading, Mass.: Addison-Wesley.

Peterson, James L. 1981. *Petri net theory and the modeling of systems.* Englewood Cliffs, N.J.: Prentice-Hall.

Pritsker, A. A. B., and Pegden, Claude D. 1979. *Introduction to simulation and SLAM.* New York: Wiley : distributed by Halsted Press.

Inputs

Rolia, J., and K. C. Sevkik. 1995. The Method of Layers. *IEEE Transactions on Software Engi¬neering, Vol, 21*682-688.

Rumbaugh, James, Jacobson, Ivar, and Booch, Grady. 1999. *The Unified Modeling Language reference manual.* Essex, UK, UK: Addison-Wesley Longman Ltd.

Sargent, Robert G. 2005. *Verification and validation of simulation models.*Orlando, Florida ed.Winter Simulation Conference.

Sargent, Robert G., et al. 2000. *Strategic directions in VV\&A; research: strategic directions in Verification, Validation, and Accreditation research.*Orlando, Florida ed. San Diego, CA, USA: Society for Computer Simulation International.

Schruben, Lee W., et al. 2003. *Advanced event scheduling methodology: advanced event scheduling methodology.*New Orleans, Louisiana ed.Winter Simulation Conference.

Taylor, M. 1996. Partial Differential Equations: Basic Theory. *Springer Verlag*.

von Bertalanffy, Ludwig, and von Bertalanffy, Ludwig. 1969. *General system theory : foundations, development, applications.* New York: G. Braziller.

Woodside, C. M. 1988. Throughput Calculation for Basic Stochastic Rendezvous Networks. *Performance Evaluation* 9, no. 2:143-160.

Woodside, C. M., et al. 1995. A Guide to Performance Modeling of Distributed Client-Server Software Sys¬tems with Layered Queueing Networks. *Department of Systems and Computer Engineer¬ing, Carleton University, Ottawa*.

Zeigler, B. P., ed. 1976. *Theory of modeling and simulation.*Wiley-Interscience.

Zeigler, B. P., Praehofer, H., and Kim, T. G. 2000. *Theory of Modeling and Simulation.*2nd ed.Academic Press.

Zeigler, B. 2005. Continuity and Change (Activity) are Fundamentally Related in DEVS Simulation of Continuous Systems. *, LNCS, Vol.3397*.

Zeigler, B. P. 1998. DEVS Theory of Quantization. *DARPA Contract N6133997K-0007: ECE Dept., The University of Arizona, Tucson*.

Zeigler, Bernard P. 1990. *Object-oriented simulation with hierarchical, modular models : intelligent agents and endomorphic systems.* Boston: Academic Press.

Zeigler, Bernard P. 1984. *Multifacetted modelling and discrete event simulation.* London ; Toronto: Academic Press.