Characteristics of simulations for instructional settings

Wouter R. van Joolingen
Eindhoven University of Technology, Department of Philosophy and Social Sciences

Ton de Jong
Eindhoven University of Technology, Department of Philosophy and Social Sciences

This paper discusses the internal characteristics of simulations. The major part of it is concerned with models and their relation with the domain. Some central concepts regarding modelling and simulation are defined. These include concepts regarding:

- the structure and characteristics of the model;
- the relationship to the system that is being modelled;
- the interaction of the learner or other agents with the model.

A classification of model types is presented, accompanied by a first idea on the representation of the several types of models. The classification includes the distinction between qualitative and quantitative models. Models can further be classified into dynamic and static models, determined by the time dependency of the model. The basic elements of any simulation model are the state of the model, describing the properties of the system that is modelled, and a set of rules determining the possible development of the model state. State space is the collection of all possible states.

In quantitative models the basic elements of the state are variables, which can be dependent or independent. Dependent variables are variables of which the value is determined by the independent variables. The model rules are equations, determining the development of the values of the variables. Quantitative models are classified into discrete and continuous models, depending on the structure of the state space. Qualitative models have a state space consisting of propositions about the modelled system. In this case, the model rules have a more descriptive character.

A brief discussion of the relationship between the model and the corresponding real system is given. Three types of real systems are distinguished: physical, artificial and abstract. The main criterion for a distinction between these types of systems is the possibility of constructing a model that describes the system completely (a base model).

The interaction of the learner with models and simulations is described by introducing the concepts of interaction and scenario. The interaction describes the sequence of operations that are performed upon the model, the scenario includes the interaction and the agents who take part in the interaction.

Classifications of instructional simulation environments (often just called: instructional or educational simulations) are discussed. The usefulness and features of these classifications are investigated. Many of the existing classifications do not distinguish very well between relevant aspects of simulation learning environment.

Three sections describe the relationship between the internal characteristics of simulations and the four themes introduced in de Jong (this volume): domain models, learning goals, learning processes and learner activity. Because simulation models are discussed extensively in the first section of this paper, the section on domain and simulation models gives an overview of domain aspects that are not explicitly referred to in the model. Here, an additional knowledge base, called the cognitive model will be introduced. For each type of learning goal the relation with the domain model or scenario is elaborated. The relationship between learning processes and learner activity and domain models is discussed by relating the possible types of learner activity with the model and scenario elements, resulting in demands for the structure of the model or scenario.

1. Introduction

This article reflects an attempt to make a classification of computer simulations and their related domain aspects on the basis of the internal characteristics of simulations. The overall view taken is that the domain aspect of simulations consists of three parts:

- The model underlying the simulation. This model makes the simulation work and is a representation of some real system.
- The interaction with the model, determining the development of the model state. This can mean both the interaction of the learner with the model and the interaction of other agents with the model.
- Relevant additional knowledge about the model, the system that is modelled and about the relation between these two. This can be knowledge about the meaning of model entities in terms of the real system, e.g. knowledge about critical values, but also an alternative description of the knowledge contained in the model, for purposes of better understanding.

We start with defining the concept of simulations. Numerous definitions are available and we will try to derive a common denominator. Here we also describe the boundaries between the concept simulation and concepts such as system and models.

The second major part of the article starts with a gross classification of simulations into quantitative and qualitative simulations. For quantitative simulations a host of classic literature exists. An over-
view, treating concepts such as discrete and continuous simulations, is given in Section 3. Qualitative simulations have a shorter history. They will be described in Section 4.

The interaction aspect is described in Section 5. In this section the interaction concept is given a more precise definition. Two other concepts: procedure and scenario are introduced and discussed.

Besides the more ‘classical’ classifications of simulations, other classifications (often bearing a more practical character) exist. These classifications are explicitly related with the use of simulations in instructional settings and can be found in Section 6.

In de Jong (this volume) we introduced the concept of an ISLLE (Intelligent Simulation Learning Environment). The contents of such ISLLEs are described in this volume in terms of four themes. Sections 7, 8 and 9 relate the topic of the present paper to these themes. The first of these sections discusses the relation domain - model. Models are extensively discussed in the first and second major parts of this paper. For this reason we will discuss here the relation of models with (other parts of) the domain. Section 7 therefore discusses the third part of the domain: additional domain knowledge needed in order to understand more deeply the relations laid down in the model. In this section the term cognitive model will be introduced.

Section 8 is concerned with the relation between learning goals that can be obtained with computer simulations embedded in a simulation learning environment and the internal characteristics of the simulation. Section 9 deals with learning processes that can be initiated with computer simulations, and with the activity the learner may show when working with a simulation. The handles that can be set on the model (points at which the learner may manipulate the model) are investigated here.

2. Concepts of modelling and simulation

2.1. Models

A model is a representation of a certain system. A model is designed to be an analogy of the system under study. This means that elements of the modelled system will have counterparts in the model and vice versa. An important feature of a model is that, because of the analogy between the model and the modelled system, it is possible to obtain information about the modelled system by studying properties of the model. This characteristic of models can be used for a definition of the concept ‘model’: Any system (A) that is used to obtain information about another system (B) is a model of system B (Bertels & Nauta, 1976).

Bertels and Nauta (1976) distinguish many kinds of models by developing a multitude of classifications for models. For the current purpose only two classifications are important: empirical vs. formal models and quantitative vs. qualitative models. Empirical models are themselves physical systems which have some resemblance to the modelled system. Formal models are a set of relations between formal concepts. For our purposes only formal models are of interest since only formal models allow implementation on a computer. When in this article we use the term model we intend it to refer to formal models. Qualitative and quantitative models are both formal models.

In the case of quantitative models the relations between the several model components are described by quantitative mathematical relations, in the second case this is not true. For performing a qualitative simulation with a computer some formal representation of model concepts will be necessary. Artificial intelligence techniques are frequently used in qualitative simulation (Kuipers, 1986; Fishwick, 1989a,b).

2.2. Modelling concepts

Zeigler (1976) distinguishes a number of concepts in relation to modelling. First there is the real system which is to be modelled. In principle there can be a base model of this system, which describes every aspect of the complete real system. From these base models lumped models can be derived.

The term real system must be understood in a broad sense as the system that is modelled. In practice that can be a real, physical system but also a hypothetical system, like a non-existing apparatus or an ideal world, e.g. a world without friction. For this latter kind of system the term ‘real system’ is not really applicable but the term will be used for all systems that are modelled.
which describe only a subset of the real system. This subset is defined by an experimental frame. This experimental frame serves as a filter which selects information that is relevant for a specific purpose. In this sense the experimental frame also filters the data that will be reconstructed by the lumped model from the total amount of data that can be obtained from the real system.

Modelling a system always implies simplifying it, and in this respect the concept of an experimental frame is a useful one. The experimental frame restricts the view of the observer and the modeller to that part of the real system that is interesting for a specific purpose. It also defines limits on the experimental instruments used to retrieve information from the real system. For instance, an experimental frame can determine that measurements can only take place at sharply defined, discrete time points, or that some details of the system cannot be perceived by the observer, in which case it is not necessary to model those details. It is possible that a lumped model be derived from a more general base model or that it is constructed from scratch, only with the goal of reconstructing the information that comes through the experimental frame (see Figure 1).

A formal model that can be used for simulation has at least two components: a state, representing the state of the real system and a set of rules, characterising the possible changes of the state, depending on time or other internal or external influences. At this point we will not be more specific in relation to these concepts, because both state description and rules strongly depend on the type of the model. These concepts will become clearer when specific model types are discussed.

2.3. The relation with the real system

The value of the modelling structure, introduced by Zeigler and depicted in Figure 1, is dependent on the usefulness of the concept of base model, apart from the fact that a lumped model can be constructed, using only the filtered data, as noted in the previous section. The existence of a base model determines if and how the experimental frame can be specified. In order to make some more precise statements about the possible existence of base models we must first distinguish between several types of systems that can be modelled. It appears that base models cannot exist, in the strict sense meant by Zeigler, for all domains. We distinguish between physical, artificial and hypothetical or abstract systems.

Physical systems, which could also be called natural systems, are systems that occur in the natural world. Models of physical systems do not represent the system directly but refer to observed characteristics. Since these systems are complex in most cases, and observations are always incomplete and inaccurate (to a certain level), it is impossible to construct a complete base model of these systems (in Zeigler's sense). The term physical should not be taken too literally. For example social and biological systems are also observable, and can therefore be regarded as physical systems. The objective of scientific research is to construct models that produce a faithful reproduction of observations, and to test these models in new situations².

In the modelling process—the creation of models to be implemented into simulation learning

²For the use of models and simulations in instruction the above discussion is a minor issue. In most (probably all) cases only validated models of physical systems will be used. The learner will investigate the properties of the real system by investigating the model. The question how the model is originally constructed is then unimportant.
environments—one can regard the most accurate model of this reality as the base model. In most cases this is not very helpful, because the distance between the observation level that is needed for the model that will be implemented on the computer and the most advanced physical model is so large that there will be no useful relation between the two. For example when one wants to implement a model of the ecological processes in a pond one does not start at the subatomic level of the entities that are present in the pond.

Summarizing, we can say that for physical systems it is impossible to construct a base model and that the models that are created for physical systems can be compared with observations. This introduces the concept of validation of the model: comparing the model results with real observations and recognizing (or not) that the model is a more or less faithful representation of the real world. Of course this validation has to be done by the author of the model. However, the learner who is working with the simulation might also need to validate the model because, in order to learn something about the domain that is simulated on the computer s/he must recognize the model as a representation of an external system.

An artificial system is a system that is created by human beings. This can be some artefact, like a machine, or some artificially created situation, like a post office. The base model of an artificial system is always known, indeed: the artificial system is constructed using a model that was created before the system itself. This ‘model of construction’ is the base model for the artificial system.

The base model of the artificial system is not the ‘deepest’ level model that is possible. There can be (and in practice there are) models which describe the artificial system at a lower and, consequently, more detailed level. However, these models will in essence not be models of the artefact but models of a physical system which underlies the artefact. For the purpose of modelling the artefact just the behaviour of the system, as described in the model to create the artefact, is complete and sufficient information to describe it. Any deeper level model will not be a model of the artificial system and modelling at such a deeper level does not make sense. For artificial systems almost the same validation arguments hold as for physical systems except that validation has no real purpose since the system has been constructed using the model.

A hypothetical or abstract system is a system with no counterpart in the real world. Hypothetical systems are used, for example, to illustrate some effects that are not clearly visible in the real world, or to show how the world would look if certain things were to differ from reality. An example is a world without friction. This world can be used to demonstrate the effects of force. In practice it is not possible to create situations where only one force acts: the friction force will always be a second actor. Therefore, a hypothetical frictionless world can be useful to show the acting of one single force, which can contribute to a better understanding of the force concept (Van ’t Hul, Van Joolingen, & Lijnse, 1989; DiSessa, 1982).

Since hypothetical systems do not have a real counterpart, their only existence lies in their formal model. One cannot identify the real system and the base model. This means that concepts such as validation and fidelity do not have a meaning in relation to hypothetical systems. Sometimes, the difference between hypothetical and artificial can be problematic. As an example we can take models of an economy. The concepts that are used in these kinds of models are very abstract. On the one hand the model is designed to describe processes in the economy and should therefore be validated. In this sense the model would be a physical model. On the other hand, validation is most of the time a difficult issue for many types of economics models, implying that many of these models are only used as an organizer for thoughts about economy. This would make the system under study an abstract or hypothetical system.

2.4. Simulation

Simulation is a way of experimenting with a model of a system in order to retrieve information about the model and the modelled, real, system. Sometimes performing simulation experiments is called modelling. However, it should be noted that the term modelling implies constructing and investigating the properties of a model, whereas the term
simulation only refers to the second of these two activities.

Simulation is carried out by applying the model rules to the model state. This application strongly depends on the type of state and rules, but also on the purpose of simulation. The latter depends on the kind of information that is to be obtained from the simulation.

Simulations are often carried out by means of electronic computers, for reasons that are clear. This implies that the model must have a formal representation, which can be interpreted by the computer and can be used to calculate the development of the state in an unambiguous way.

3. Quantitative models

3.1. Types and representations of quantitative models

3.1.1. Variables

The model is a representation of the modelled system. For the purpose of modelling, the modelled system is nothing more than a set of input and output quantities, with a certain input-output behaviour (Zeigler, 1976). The most important concept in relation to quantitative models is the concept of variables. Variables can take values from a predefined set of possible values, depending on the model. In this way variables represent the system’s state.

A variable can be an input or an output quantity or a quantity that is attributed to the modelled system. The latter kind is part of the model but no ‘real world’ quantity that can be directly measured can be associated with it, only some hypothetical quantity can be associated with these variables. The purpose of a quantitative model is to describe the input-output behaviour of the system by defining relations between the various variables and thus defining the development of the model’s state. The experimental frame then defines which states of the real system correspond with the model’s states.

An (often used) example of a quantitative model is a model of a rabbit population, where the variables are, for instance, the number of rabbits and time. The model then describes the behaviour of the number of rabbits as a function of time. Another quantitative model is a post office model, where the variables are time, the number of people waiting in each queue and the number of tellers who are actually serving people.

A distinction must be made between dependent and independent variables. Independent variables are not influenced by other variables in the model. When one of the independent variables is time the model is called dynamic, otherwise static. It should be noted that time can be present implicitly in a model, making the model dynamic, without explicit time dependence. For example, when the only independent variable in a model is the amount of water added to a vessel, and the flow of water is a function of time the model is dynamic, without an explicit time dependency. A model is static only if the independent variables can be tuned back and forth, independent of time. Mostly, the number of independent variables is small; the complexity of the model rises strongly with the number of independent variables.

Dependent variables are a function of the independent variables and/or each other. The number of dependent variables is unlimited, in principle. In practice numbers of one to several thousands occur.

In the case of a dynamic model the notion of dependent and independent variables is less clear. In this case the development of the system is often described by differential equations. In mathematics the independent variables in differential equations are the variables on which an integration can be performed. In the case of ordinary differential equations only one independent variable is present, but the choice of initial values of the ‘dependent’ variables is still free for those dependent variables that are mutually independent.

3.1.2. Parameters

Other quantities that are important in modelling and
simulation are parameters. Parameters are quantities that depend on external circumstances and are not influenced by variables. The input-output behaviour described by the model depends on these parameters, e.g. the fertility of rabbits may depend on the environmental temperature, influencing the growth rate of the rabbit population.

Parameters can be subdivided in internal and external parameters. Internal parameters describe constant system properties that have no direct relationship with variables. Examples are: the volume of a tank, or the length of a pendulum. External parameters represent the relation with the environment. They can be uncontrollable values, such as temperature, or outputs from some other system that is not included in the model.

Parameters can be static or dynamic. Static parameters are constant during the model time (i.e. the value of the independent time variable). Dynamic parameters are, explicitly or implicitly, a function of model time. Usually, internal parameters will be static, whereas external parameters can be both static and dynamic.

Dynamic parameters can be deterministic or stochastic. The latter model some random external process, e.g. the entrance of customers into a post office or a random internal process, e.g. noise. Variables which depend on stochastic parameters will be stochastic themselves.

The difference between parameters and variables is rather subtle. The most important feature of parameters is that they are independent of the development of the model state. This means that they remain the same, whatever the development of the model state. This means that the causes of changes of parameters are never the objects under study, when using a simulation. Of course the effects of these changes can be objects of interest and study.

A quantity that is in one case a parameter can be a variable in another case, depending on the scope of the model. This is defined in the experimental frame. The environmental temperature, for instance, may be a parameter in one case, but when the causes of temperature change are incorporated in the model the environmental temperature will be a variable.

3.1.3. Discrete and continuous models

Systems that can be modelled can be of very different types. Therefore, the models for these systems may have a different nature (Pritsker & Pegden, 1977). An important dimension of models is continuity. This term applies to the variables of the model, which can be continuous or discontinuous (discrete). The dependent variable of a model of a population of rabbits is the number of rabbits, which is always an integer. This implies that in a model the number of rabbits will be a discontinuous function of time. A change in the number of rabbits always implies a discontinuous jump in the rabbit function. On the other hand, the charge of a capacitor is an irrational number. The charge is, in a model, a continuous function.

The independent variables can also be both continuous and discontinuous. For instance when the experimental frame is such that the rabbits are viewed only once a year, the model time is discontinuous. The charge of a capacitor can be measured at any time, implying a continuous time variable. Models can thus be classified as discrete-discrete, discrete-continuous, continuous-discrete and continuous-continuous where the first word applies to the dependent variables, the second to the independent variables. It turns out that the continuity of the dependent variable(s) determines more deeply the characteristics of the model than does the continuity of the independent variable(s). Therefore a major classification can be made between discrete and continuous models, depending on the continuity of the dependent variable(s).

Besides these categories a third one is possible. As an example we take the concentration of a chemical reactant. In general, it changes continuously over time, but when a certain amount is added to the vessel, a discontinuous jump in concentrations is possible. This type of model, which allows both discrete and continuous variables, will be called a hybrid or mixed model.

---

*A dynamic parameter stays the same in a relative way: its time dependence does not change, though of course its value can change.*
3.1.4. A note on continuity of models.

It needs to be stressed that the continuity of the model is important for simulation purposes, not the continuity of the modelled system. For determining whether the model will be discrete or continuous, two factors are important: the continuity of the real system and the experimental frame. Of these two, the latter is the major factor. For instance, the rabbit population has a discrete dependent variable (the number of rabbits) and a continuous independent variable (time). However, the experimental frame determines that the population is measured only once a year, making the model time variable discrete. Also it is possible that the experimental frame determines that only large populations and averages are to be considered in the model, making the model number of rabbits (and the model as a whole) continuous.

It should be noted that the possible values of a continuous variable can be mapped into a discrete set of values. In fact a computer can only represent discrete values and always performs this kind of mapping to represent real values. However, when this kind of mapping occurs to values of continuous variables this does not make the model discrete. There is an important difference between the interpretation of continuous and discrete variables which is not reflected in this kind of mapping operation. Every real value of a continuous variable has a meaning in the model, meaning that it represents some physical state. This is not true for discrete variables where only values in a discrete set can be assigned a meaning. E.g. in the post office model, the variable which represents the number of waiting customers in a queue can only take integer values. A value of 3.5 customers in a queue has no meaning.

3.1.5. The state

An important concept is the concept of state, as previously noted. For quantitative models this concept can now be defined: A subset of the set of dependent variables is a state when from this subset the values of all other dependent variables can be calculated. Most of the time several choices can be made for defining the state. A variable in the state is called a state variable. A state is called minimal if removing any state variable results in the remaining subset no longer being a state.

State space is the set of all possible values which the state variables can take. When there are N variables in a minimal state, the state space will be N-dimensional. The state of a model can be represented by a single point in state space. A change in the choice of variables in the minimal state corresponds with a different choice of coordinates in state space.

For a simple static model, where the values of the state variables do not depend on previous states, the dimensions of state space (or the number of degrees of freedom) equal the number of independent variables. But for a system in which the state development depends on the current state (in practice almost all systems are of this type), the number of degrees of freedom equals the number of initial conditions that can be chosen freely. The independent variables are not considered to be state variables in this case.

The concept of state is especially important in the case of dynamic models. In this case only the time development of the state variables needs to be calculated, because all other variables are dependent on the state. This saves calculation resources, because the values of variables that are important for a specific purpose need to be calculated only at interesting time points.

Because of the reason given above it is important for the modeller to pay attention to the choice of state variables. The state chosen should always be minimal. Those variables should be chosen as state variables which make the representation of the time development of state variables the most efficient. This may, as an example, result in a different choice of coordinates.

3.2. Discrete models

3.2.1. Different viewpoints for discrete simulations

As was previously noted, the dependent variables of discrete models only take discrete values. In other words, the state space is discrete. For discrete models an event occurs when the value of one of
the state variables changes. When the events occur only after fixed time intervals, the model is discrete-discrete, otherwise it is discrete-continuous. In the latter case, the events can occur at any time. This is the case at a post office, where customers can enter, wait and be served at any time.

In discrete models, objects like customers and tellers are often described independently of each other. This means that many objects may exist at one time, each having a simple structure. This is in contrast to a continuous model with fewer objects with a more complex structure, resulting in more complicated relations between variables.

An object has a limited life-cycle within the model. It is being created (the customer enters the post office...), processed (...he is being served...), and deleted (...and exits the office). This is called a process. A part of this process is the activity that is undertaken with the object. A third concept is the event. In the example of the post office there are three events: the entrance, the start of service and the end of service/exit. When a model of a post office is represented in a symbolic system (e.g. a simulation language) it is possible to do this from three viewpoints or approaches, depending on the type of elements that are regarded to be central: events, activities or processes. The symbolic system chosen depends on the choice of approach, or, the choice of approach depends on the symbolic system selected. In the following sections each of these approaches will be discussed in more detail.

3.2.2. The event approach

When events are the central concepts of a model, the process of simulation can be described as follows, in pseudocode:

```
central processor
begin
  initialize system
  while not end of simulation do
    t := time of next event;
    check event list;
    handle next event
  od
end
```

An important element in this approach is the event list, which holds a record of all future events. At every simulation step the next event time is looked up in the event list and this event is handled. Simulation time increases with this process. The handling of an event influences the state of the system. Also the event list can change under the influence of an event handling. The post office simulation (a post office with a single teller and a single queue) is an example:

```
handle arrival event
begin
  if teller idle
    then
      make teller busy;
      determine service time;
      schedule end of service event at t + service time
    else
      queue := queue + 1
    fi
end

handle end of service event
begin
  if queue = 0
    then
      make teller idle
    else
      queue := queue - 1;
      determine service time;
      schedule end of service event at t + service time
    fi
end
```

It is not necessary to specify a start of service event because this always coincides with an arrival event or an end of service event. The arrival events are generated by some external source. An event oriented simulation language generally has the central processor included, together with some facilities to monitor the process. The user limits him/herself to specifying the state, the types of events and the event handling procedures, for which the simulation language offers some facilities, like a standard procedure library.

3.2.3. The activity approach

The activity approach implies that the simulation system has to scan all possible activities at all possible times to see which activities are currently active. Events are implicitly present as the start or end of an activity. While in the event approach the waiting queue was a central object, in the activity approach the teller is. This approach is often called
the activity scanning approach for obvious reasons. Because all activities have to be scanned at every possible simulation time this approach results in inefficient programs and is not used in practice (Bratley, Fox, & Schrage, 1977).

3.2.4. The process approach

The process approach takes the objects as central elements of the simulation. The division of the larger system into smaller objects is most clear in the process oriented approach. Every object is now depicted as a kind of mini-simulation of a subsystem. There is no description of a global state but every object has its own 'private' state variables. A central processing procedure keeps all these processes in pace. The post office problem is now coded like this:

```
customer
begin
  enter queue at entrance time;
  wait until teller idle and you are the first in the queue;
  exit queue, begin service;
  be served until end of service time
  exit;
end;

teller
begin
  while not closing time do
    become idle;
    wait until queue not empty;
    become busy;
    remove first from queue;
    serve until end of service time;
  od
end;
```

No specifications are needed for separate events, these are described in every separate process description. Of course some global system characteristics need to be described, but many of these global characteristics are defined implicitly in the several process descriptions.

3.2.5. Comparison of the approaches

Each approach has its merits. The process and event approaches in particular are good for describing discrete simulations. There are of course some differences.

The greatest virtue of the process approach is its clear structure. By dividing the larger system into small entities, for which the simulation process can be described separately, a clear overview is supported. The process view is suitable for representation as a graphical network. Some simulation languages rely on such a representation (for example SLAM and GPSS, see Pritsker & Pegden, 1977). The network representation is often carried into the computer by translating the network into a directly corresponding text language.

The event approach has a simple structure. It is easy to envisage the processes in the computer, represented by an event oriented language. This can be an aid in developing a model. However, when the state of the system becomes more complex and the number of different types of events increases, the clarity of the system decreases accordingly. This is because every event has its own effect on the state of the system, which can be the cause of some interference between events, unnoticed by the author of the model.

An advantage of the event approach is that it is very clear how an interface to some external process can be realised. The external process can put an event on the event list and, on the other hand, some events can influence the external process. This is all very transparent. It is also possible to create some interfacing from a process oriented approach, in this case the external process should address the individual objects. This is somewhat less transparent.

3.3. Continuous models

In a continuous simulation the process is described in a different way. Now the state variables are more explicitly present. The description of the system is explicitly in terms of these state variables and related variables. The time development of the system's state is controlled by differential or difference equations. Time is divided into small slices, depending on practical conditions and the required calculation accuracy. The modelling system calculates the state variables and other values as a function of time, depending on the given equations. Mostly, several integration techniques are available.

Take the example of the growth of a rabbit popu-
The number of rabbits is described by a continuous variable, called RABBITS. The system is described in terms of difference equations, in the language of the very simple continuous modelling system DMS (Ogborn, 1985; 1988).

\[
\begin{align*}
\text{FERTILITY} &= (1 - \text{RABBITS}/\text{MAX}) \times \text{F0} \\
\text{BIRTHS} &= \text{FERTILITY} \times \text{RABBITS} \\
\text{RABBITS} &= \text{RABBITS} + \text{BIRTHS} \\
T &= T + 1
\end{align*}
\]

The modelling system calculates from the values at time \( T \) the values at time \( T + 1 \). The user has to supply values for the constants \( \text{F0} \) and \( \text{MAX} \) and initial values for \( T \) and \( \text{RABBITS} \). The only state variable is the variable \( \text{RABBITS} \). The same system can be modelled in a system which takes differential equations. In this case the third line would read as (in some arbitrary notation):

\[
\text{D(} \text{RABBITS} \text{)} = \text{BIRTHS}
\]

and the last line would vanish because the system itself takes care of time progress. This leaves open the choice of integration technique and suggests more strongly that the system is continuous. In the case of the rabbit population that may not be a wise choice, when the real system is monitored from generation to generation. In this case the system is continuous-discrete and times between two measurements have no meaning because they cannot be observed. \( T \) takes the function of a generation counter. This is true in general: a continuous-discrete model can not be described in terms of differential equations, only difference equations can be used. When a continuous-continuous model is represented in terms of difference equations, in effect the model will be continuous-discrete. However, the interpretation will be different from a 'real' continuous-discrete model, since time points between two calculation steps have a meaning in continuous-continuous models. Values for variables at such time points can be obtained by interpolation.

3.4. Mixed models

Within one model, both discrete and continuous processes can take place. In a chemical plant the concentration of a certain substance can be described by a continuous variable but some discrete events can take place, for instance the adding of a great amount of substance to the vessel. These events can even be generated by the continuous variables. The decrement of the concentration below a certain critical value can trigger an automatic addition of a new amount.

It is not possible to describe these models with any of the types of simulation languages characterized before. One solution to this problem is to model the continuous variables as discrete objects. This approach has some major drawbacks. Accuracy is not easy to control and the process of solving the differential equations has to be modelled in a discrete modelling system. The other way around, modelling discrete variables in a continuous modelling system also doesn't work very well: the modeller cannot use the specific features of the event and process approaches. A better solution is to make use of a simulation system which can combine both types of variables.

Both process and event description approaches can be combined with continuous variables. In both simulations, time is running in the background. This makes it possible to calculate the current values of the continuous variables at any event. In this type of description the differential/difference equations and the process/event descriptions are given separately. Take as an example the chemical reaction:

\[
\text{handle add_event}
\begin{align*}
&\text{begin} \\
&\quad \text{concentration} := \text{concentration} + \text{amount}/\text{volume} \\
&\text{end;}
\end{align*}
\]

\[
\text{state (of continuous variables)}
\begin{align*}
&\text{begin} \\
&\quad \text{reaction\_speed} = f(\text{concentration}, \text{other variables}) \\
&\quad D(\text{concentration}) = \text{reaction\_speed} \\
&\quad \text{increase time to current time} \\
&\quad \text{schedule add\_event when concentration < critical value} \\
&\text{end;}
\end{align*}
\]

The procedure \text{state} is now called at every event in order to calculate the current concentration. Within a process oriented approach there are two possibilities for the continuous state variables. They can be an assigned property of an object, in which case they are \text{local}, or they can represent a global aspect of the system (e.g. the temperature), in which case they are \text{global}. The integration of local continuous
variables is done within the object descriptions. The state evaluation of global variables is done by a procedure which is called every time an object requests the values.

Mixed systems occur very often. In industrial practice almost every model that is used has a mixed nature. When large systems are modelled both discrete and continuous variables will almost always occur. 'Pure' continuous or discrete models only occur as small models and, more importantly, as submodels of larger systems.

4. Qualitative models

For several reasons it might be useful to describe models not in terms of quantitative relations but in terms of qualitative relations. Hartog (1989) states that in the context of instruction and learning qualitative models are extremely important because they are closer to the mental model the student is to acquire (see also Section 7).

It should be noted that there is no sharp distinction between quantitative simulation and qualitative simulation (Fishwick, 1989b). Some modelling methods have both qualitative and quantitative aspects.

In general two types of qualitative modelling can be distinguished. The first one describes quantitative aspects of a system qualitatively, for example: 'The voltage is oscillating' or 'The temperature is monotonically increasing'. The second type of qualitative modelling is where non-quantitative properties of a system (or model) are modelled. An example is a model of an electric circuit where only aspects like on and off, connected and not connected are taken into account. Fishwick (1989b) refers to these two model types as quality-based and abstraction-based respectively. In the present paper we will adopt his terminology.

The experimental frame (see Section 2.2) defines the qualitative or quantitative nature of the model. The experimental frame determines whether quantitative properties of a system will be treated as numbers or as qualitative properties (resulting in a quality-based model), regardless of the system properties. On the other hand, it is possible to map qualitative attributes of a system onto a numerical system, with mathematical relations defining the behaviour of the model. Usually this will result in a discrete event simulation. It will be hard sometimes to tell the difference between a discrete event model and an abstraction-based model. In a certain sense quality-based models can be compared to continuous quantitative models whereas abstraction-based models are comparable to discrete quantitative models.

4.1. Quality-based models

In quality-based models three aspects can be qualitative: the input, output and the model. The term model here applies to a more restricted internal model (sometimes called the core model). The inputs are given in a language describing the qualitative features, with elements like (positive, increasing), the output is given in the same terms. A qualitative model is capable of reasoning with the input language to give a qualitative description of the behaviour of the model.

It is not necessary that all three aspects are of the same kind (qualitative or quantitative). Qualitative input can be fed into a qualitative model giving qualitative output. When two consecutive elements are of a different kind, a translation is needed between qualitative and quantitative representations and vice versa. For example, the qualitative input 'the voltage is oscillating' may translate into \[ V(t) = \sin(t) \], where \[ V(t) \] is fed into the qualitative model. Note that this adds some information to the input. Fishwick (1989b) calls this replacement of a qualitative term with a quantitative function instantiation of the qualitative term, as opposed to constraint propagation where the boundaries for the variables resulting from the qualitative statements propagate through the model. An example may clarify this point. A qualitative statement may be: A is somewhere between 3 and 5, B is somewhere between 8 and 12. The question is: what can we say about A+B. Instantiation methods would make an estimate for A and B, say 4 and 10, and add these giving 14 as an estimate for A+B. Constraint propagation methods would add the boundary values and state that A+B is somewhere between 11 and 17. For a model which contains more complex relations (e.g. non-linear relations) between variables, one needs of course more sophisticated
techniques for constraint propagation and instantiation.

When qualitative output is required from a quantitative model, a reverse translation from quantitative data to qualitative statements is needed.

Note that, in the case of quality-based modelling, there is always a quantitative model underlying the qualitative one, whether known by the author or not. There can be several reasons to decide to perform the simulation qualitatively if the quantitative model is known. If it is unknown, qualitative simulation may be the only way to perform the simulation.

First, qualitative simulation may be desirable because there needs to be a qualitative user interface around the model. For example, this could be a natural language interface. Under certain circumstances, it might be better, for reasons of efficiency and accuracy, to perform the whole simulation qualitatively rather than to translate the qualitative input into quantitative, perform the quantitative simulation and translate the results into the qualitative language.

The second reason to prefer qualitative simulation over quantitative is when the equations describing the system are so complex that it would require large amounts of calculation time to perform all necessary calculations. Qualitative simulation might be more efficient if one is not interested in the quantitative details. This kind of modelling results in less accurate simulations.

A very important reason to perform qualitative simulation is to model human reasoning about a certain system. In this case the main objective is not to build a model which gives highly accurate predictions of the system under study, but to build a model of the expert reasoning about the system. This will bring two systems under study: the modelled system and the human reasoning about the system. Qualitative models of this kind may play an important role in instructional programs e.g. in simulations in which the learner must play a certain role in interacting with other people. The simulation then models the other players. Another application might be to compare learner behaviour with expert behaviour in order to be able to give appropriate feedback to the learner.

Qualitative models can play a role during model development. In this case the relationships in a system are first stated in a qualitative form. During model development several versions of the model may be developed, possibly with the help of qualitative simulation techniques. In the end the qualitative model can be translated into a quantitative model. In this paper we will not go deeper into this matter. Model development is elaborated in Van Joolingen et al. (1990).

4.1.1. A quality-based algorithm

Kuiipers (1986) describes an algorithm for quality-based simulations, QSIM, for systems that can be described by differential equations. In this section this algorithm will be described in brief, to give an impression of the type of reasoning systems that exist in qualitative simulation.

QSIM starts with the differential equations modelling the system under study. From these equations qualitative rules are developed, such as: DERIV(f,g), meaning that g is the derivative of the function f, and ADD(f,g,h) meaning that f(x)+g(x) = h(x), for all x in an appropriate domain.

For each variable in the system, landmark values are defined as those values where something interesting happens, e.g. a maximum of the function or crossing zero or another critical value, depending on the system characteristics. Also some distinguished time points are defined as the time points where (one or more) variables reach landmark values.

The qualitative state of a system is now described by a set of propositions of the form <qval, qdir> indicating the value of the variables, and the direction of change (corresponding to the sign of the derivative), qval can take the form l_i, indicating that the variable is at a landmark value or (l_i, l_{i+1}), indicating that the value is somewhere between two consecutive landmark values. The qdir slot takes values of the set {inc, std, dec} as mnemonics for increasing, steady and decreasing. Time is described in terms of t_i and (t_i, t_{i+1}), where the t_i's are the distinguished time points.

From a certain qualitative state, transitions are possible to other qualitative states. Kuiipers gives 16 possible transitions. From a particular state a maximum of four transitions are possible. The algorithm
builds a tree of possible system development, using the initial state and the possible state transitions. The reasoning mechanism is capable of adding new landmark values to functions if there is a need for this, e.g. when it is discovered that a derivative travels through zero.

The qualitative model rules now constrain the development of the system tree. By an inference mechanism, impossible branches of the system tree are eliminated, resulting in one or few possible system development paths. Using this algorithm, it is possible that there are several ways the qualitative state of the model can develop. The qualitative reasoning mechanism is not deterministic. Kuipers proves that this is true for this kind of reasoning system in general.

4.2. Abstraction-based models

Abstraction-based models model non-quantitative aspects of systems. An example was already mentioned at the beginning of this paper. This system (modelling electronic circuits) was implemented in an instructional program by White and Frederiksen (1987a,b; 1990). Their system contains three, increasingly complex, models of electronic circuits. Using Fishwick’s terminology, these models could be classified as abstraction-based, quality-based and quantitative. The abstraction-based model is capable of generating the new states of every component in the system when something in the system is changed, e.g. when a switch is opened or closed the state of a bulb (alight or not) will be re-evaluated.

In a representation of abstraction-based models the causality of the system must be represented. This can be done by using a set of if-then rules (production rules, see for example Klahr, Langley, & Neches (1987)) or by constructing graphical representations: causal graphs or finite state machines. These techniques are essentially knowledge representation techniques as used in artificial intelligence.

5. Underlying models, procedures and scenarios: the interaction process

Sections 2 - 4 presented an overview of internal characteristics of models underlying simulations. One of the characteristics of the model is whether one of the independent variables is time, thus defining a difference between dynamic and static models. This concept of time should not be confused with another time aspect frequently found in describing interaction with a simulation: the timing of learner input and system output. This latter interaction aspect describes the sequence of the different inputs by the learner and the different outputs by the system. Mostly the learner has to provide the system with a number of different inputs and the system returns (at least one) output. When there is a normative sequence of input and output associated with the interaction we will speak of the procedure or skill associated with the (underlying) model or the real system that is modelled.

The input sequence is sometimes a free choice for the learner, but in other situations it can be prescribed and is thus fixed. Of course mixed initiative is also possible. Prescribing the sequence can be, amongst other things, the result of instructional decisions. In principle, this timing aspect is independent of the dynamic properties of the model, but in practice a relation will exist.

A third possibility is that the process of interacting is the subject of instruction itself. In other words, mastering the interaction procedure is a learning goal. This is the decisive distinction between learning the properties of the underlying model (both static and dynamic) and learning a procedure or skill: the skill is not only aimed at providing the right input but also at the process of interaction (see van Berkum & de Jong, this volume). However, it will also be clear that for insightful learning of the procedure, knowledge of the related model will be necessary, at least in some qualitative, informal sense.

Figure 2 gives a schematic overview of the interaction sequence aspect in relation to the underlying model. Of course this figure only depicts an example. The underlying model is depicted as a simple box with (input)variables, (input)parameters and output variables indicated. Sequencing these inputs and outputs defines the timing of the interaction. If a choice can be made by the learner when (within which time frame) to give certain input and/or get certain output, this is indicated by grey
areas. The direction of the lines does not indicate input or output relations but gives the relation between internal component of the model and the time axis.

![Diagram](image)

**Figure 2** Schematic overview of adding an interaction element to simulation models.

Decisions on the place of components on the time axis can be based on:

1. Dynamic characteristics of the internal model;
2. Logical relations in the internal model (some input is necessary before some output can be given);
3. Instructional strategies;
4. A (normative) procedure or skill as performed in the real world.

After considering these four elements, free choice for the learner might be left. (In fact, for many simulations there is only a time barrier between input and output. The learner can give all the input in the sequence s/he likes.) This free choice is indicated by grey areas in the figure.

The interaction aspect is one of the elements of what is sometimes called the *scenario* (Reigeluth & Schwartz, 1989). A second element related to scenarios is the concept of *roles*. Roles define what input a learner may give (not the sequence), and what output s/he receives. As an example we might take a simulation model on environmental issues. Roles in this simulation might be *politician, scientist, plant owner,* and *member of Greenpeace*. Each role has different input possibilities and access to different outputs. Scenarios (including interaction and roles) thus define the basic interaction of the learner with the simulation.

![Diagram](image)

**Figure 3** Schematic overview of building a scenario

Figure 3 adds the concept of role assignment to the interaction aspect. The *other agents* in the figure might be other people (even other learners) but it might also involve non-human agents (such as computers that provide random input to a variable). The computer might also be used to simulate another agent.

In summary we can say that in addition to the internal characteristics of a model, a second objective of learning with computer simulations might be the learning of an associated procedure. The procedure concerns the interaction of the learner with an underlying model. The process of interacting with the model may not only be influenced by a possible associated procedure, but also by internal characteristics of the model (dynamic properties, logical relations), and last but certainly not least, the instructional strategy that is applied. A description of the interaction process together with a role assignment is denoted as the scenario of the simulation. For designing simulations in an instructional context, a classification of scenarios would be useful.

6. Classifications of simulation learning environments in the literature

In the preceding sections, two main ingredients of
computer simulations have been described: the model and the scenario. These two main ingredients are, in principle, sufficient to provide a framework for a complete classification of internal characteristics of simulations, in terms of the themes discussed in de Jong (this volume).

Before an attempt is made to develop this classification, some classifications found in the literature will be discussed. An attempt will be made to describe these classifications in terms of the two main ingredients of the internal characteristics of simulations (models and scenarios). As will become clear, most classifications are not based on clear dimensions, which results in quite unstructured and seemingly ad hoc categorisations.

One of the most influential classifications of simulations in education is given by Alessi and Trollip (1985). They distinguish the following types of simulations:

- **Physical simulations.** These are simulations of some physical object, giving the learner an opportunity to use it or to learn about it. Typically these are simulations of machines or laboratory equipment.

- **Procedural simulations.** In this kind of simulation the student must learn certain skills needed to operate the simulated physical device. A typical example is a flight simulator, in which the learner must learn the necessary skills for operating an airplane. *Diagnosis simulations* are a subset of procedural simulations. These simulations are concerned with trouble shooting: the task of the learner is to locate a defective part in a non-functioning device.

- **Situational simulations.** In these simulations the student plays a certain role in a particular situation. For example the student could act as if s/he were a newly starting teacher at a school, confronted with all kinds of problem situations.

- **Process simulations.** In process simulations the learner provides a model with some initial values and watches the state development. There is no possibility of controlling the simulation during run time.

Alessi and Trollip's classification combines a number of the dimensions that we have identified. The first type of simulation, the physical simulation, refers to the relation of the model with the real system. In this, we also distinguished abstract and artificial models. The point of setting physical simulations apart as a separate type of simulation, is that physical systems usually are related to learning some procedure (operational knowledge). The latter category is, however, covered by Alessi and Trollip's second category, procedural simulations. Process simulations are covered by our classification of dynamic simulations; simulations in which one of the independent variables is time. Finally, Alessi and Trollip distinguish situational simulations, a classification that is covered by our scenario dimension. By describing it as a separate category the authors deny, implicitly, the fact that a physical simulation or a procedural simulation and a process simulation, could have related scenarios as well.

Reigeluth and Schwartz (1989) present a classification of simulations, based on the one by Alessi and Trollip and, what they call, the *instructional overlay* of the simulation scenario and model. They separate:

- **procedural simulations** "include both the physical and procedural categories described by Alessi and Trollip (1985)".

- **process simulations** "teach naturally occurring phenomena composed of a specific sequence of events."

- **causal simulations** "teach the cause-effect relationship between two or more changes."

To start with, this classification suggests that a physical simulation will always have an associated (normative) procedure, which of course will generally, but not necessarily, be true. Process simulations should be placed in the context of dynamic simulations. Causal simulations are possibly a good subclass of simulations, because causality is not captured by the classification we have made thus far and it plays an important role in education.

Reigeluth and Schwartz characterize their classification further by describing the level of access to variables and parameters the learner has in each of the types of models (so they set the level of learner activity as allowed for by the interaction model). In
the first category, procedural simulations, most learner activity can be present. The learner has, in principle, access to all (or many) variables and can choose his own scenario. She can really interact with the model. At all times (s)he has access to the variables, unless these are inaccessible due to restrictions imposed by the structure of the model.

Process simulations leave almost no activity for the learner. At the most some initial values may be set, but there is neither freedom of choice for the scenario nor access to variables during simulation run time. These simulations are, in the definition of Reigeluth and Schwartz, almost demonstration devices.

Causal simulations leave only little activity for the learner. The learner may choose variables, and watch the effect of his choices, but is not able to alter the scenario. Also the number of variables that can be accessed will be small ('two or more') and thus does not suggest a large number of manipulable variables.

Conkright (1985) orders simulations according to the complexity of the 'path' the learner can take through the simulation. Conkright distinguishes three categories:

- **Linear** - the learner is, at certain critical moments, confronted with a choice between several values for a variable. Only the choice of one specific value is allowed in order to proceed with the simulation. This kind of computer program can hardly be called 'simulation' because the behaviour of the program is predefined. This conflicts with one of the basic requirements we have identified for simulations (a computational model, see de Jong, this volume). It is not even necessary to represent the model in the 'simulation program' because all developments of the state are known at implementation time.

- **Branching** - the learner is again confronted with a choice between several values but now the simulation may proceed with more than one choice of values. This kind of program is already more a simulation than the linear program but the number of possible paths that the learner can take through the simulation is still finite, so there could still be a 'simulation without a model'.

- **Complex** - now the learner is completely free in determining values for some variables. In this case there is an infinite number of possible paths through the simulation. This really requires a model to be implemented in the simulation environment.

Conkright's classification is mainly concerned with the interaction of the learner with the simulation (see Figure 2). Conkright does not say anything on the nature of the manipulable factors. These may be discrete, continuous or qualitative. Also there is no elaboration on the time points at which the learner may change or choose values, especially in the complex simulations. Conkright's classification is thus a very limited one, that allows situations that we would certainly exclude from the definition of simulations, to be called simulations.

A final classification we wish to present was made by Gredler (1986). She developed a taxonomy for microcomputer simulations where the assignments for the learner and the learner activity are the central criteria for the classification of computer simulations. She classifies:

- **Structured questions and graphics 'simulations'** (Gredler's quotes). These are in fact drill and practice or tutorial programs with some graphic demonstrations. Gredler is right in putting the word 'simulation' between quotes (in fact the same situation as with Conkright's linear 'simulations' exists here).

- **Variable assignment exercises**. The scenario for the learner is fixed, but the learner is free to choose values to assign to certain variables or parameters, and to watch the development of the state of the model. This type of simulation is similar to Alessi and Trollip's process simulation and Conkright's complex simulation.

- **Diagnostic simulations**. The learner is confronted with a problem which must be solved, e.g. a trouble shooting problem. The strategy the learner takes is compared with an expert strategy.

- **Group interactive simulations**. A more flexible version of the former category. The learner is
now completely free in determining the scenario.

Gredler does not succeed in making a sharp distinction between these last two categories. Gredler gives a checklist in order to classify a simulation. However the checklist results in yet another classification: a further distinction (between 'drill and practice' and 'tutorial') for the first category is given and the last two categories are combined when using this checklist.

The main problem with Gredler's classification is that it allows for non-computational models to be called simulations. She introduces elements that do not fit in the classifications we have made in this article (giving an assignment, and the context (individual vs. group)), but we would place these extra elements not directly in the simulation itself but rather in the instructional environment that surrounds it (see van Berkum & de Jong, this volume).

It can be concluded from this short overview that most existing classifications of simulations are not based on clearly defined dimensions, that they allow for non-simulations to be classified as simulations and that they tend to include characteristics that do not pertain to the internal structure of the simulations.

In the following sections, the four themes that we found in defining the instructional use of simulations will be related to the simulation model which for the moment we will subsume under the broader heading of domain.

7. Theme 1: Models and domain knowledge

As noted in the introduction to this article (Section 1), the overlap between the theme 'models' (see de Jong, this volume) and the design component 'domains' is very large since the model itself is one of the most important parts of the domain. However, the model that makes the simulation run (the runnable model) does not cover all domain aspects that are important for learning. One of these aspects has been discussed in Section 5: the description of the interaction with the model. In the present section we will pay attention to another issue: the knowledge that is not directly contained in the model but that is needed in order to interact in a proper way with the model or to obtain some deeper understanding of the model (and/or the domain that is modelled). The additional knowledge and the model that drives the simulation together will form the domain representation of the ISLE. This section will give some outlines of this domain representation. An elaborated discussion can be found in Van Jooleinge & De Jong (1991).

It is clear that, when the interaction with the model is at stake, the knowledge that is represented in the lumped model is not sufficient to reconstruct the complete interaction between external agents and the model. Whereas the model contains sufficient information to drive the simulation, additional knowledge of the domain is needed in order to be able to make decisions to interact with the model or to understand the relations in the underlying model more deeply. In fact it will be this type of knowledge that is to be acquired in order to access a certain procedure or skill.

An example might be a simulation of a chemical plant, where one of the variables of the model is the temperature in a certain part of the plant. The model will calculate the value of this temperature but will not assign a meaning to it. For example the temperature can be too high or too low for a good controlled functioning of the plant. Additional knowledge of the possible critical values of the temperature is needed in order to be able to make decisions on when and how to interact with the plant simulation. This is (in principle) the same knowledge that an operator of a real plant should have.

Besides the knowledge when to interact, knowledge is also needed of how to interact. Of course the effects of possible actions are determined by the model, and the model can be used, by performing the simulation, for investigating these effects. But in longer term one expects a learner to anticipate these effects. In order to make this anticipation possible some higher order, more informal and qualitative, knowledge is necessary. This knowledge should be represented in the domain representation and will be termed the operational model.

Another important issue is that the conceptualisation that is used to build the runnable model may not be
the most suitable conceptualisation to understand the model. The variables and rules that constitute the runnable model are chosen to drive the simulation in an efficient way, not to obtain a transparent understanding of the model. Other descriptions of the model may offer a clearer view of the domain. We will refer to these descriptions as conceptual descriptions of the domain.

An example will make this point more clear. A pendulum can be described by physical laws in terms of two variables, the deviation from the equilibrium position and the velocity, and two parameters, the length of the pendulum and the acceleration of gravity. A model consisting of these four elements and two first order differential equations (or one second order equation which is functionally the same) provides a complete model for the pendulum. All possible pendulum behaviour can be described with this model.

However, when an expert is asked to describe the processes that take place s/he will almost never use the terms from this runnable model. S/he will say that at the lowest point the pendulum has a maximum kinetic energy and a minimum potential energy. At the maximum deviation the kinetic energy will be zero, the potential energy will be at a maximum. Without using the state equations directly, the expert can extract the most interesting features from the model.

The expert is able to do so because s/he introduces new variables, which are dependent on the state variables, and uses them for his/her alternative description. The relations that s/he uses for these new variables can be derived from the state equations and the definition of the new variables. With these relations and variables the expert is able to reason both qualitatively and quantitatively about the model and give an adequate description of the behaviour of the simulation.

All knowledge that the expert uses is implicitly present in the runnable model but it is very hard to extract it without using the additional instruments of new variables and new relations. It is not realistic to assume that a possible reasoning system could derive its own conclusions from the rather small information base of the runnable model. This illustrates the need for the inclusion of an explicit conceptual description of the simulation model in the domain representation.

Another kind of knowledge that should be included in the domain representation is teaching knowledge about the domain. This knowledge includes, among other things: information about which concepts are important to learn, logical dependencies between concepts that could influence the order in which they should be taught, and knowledge about typical situations which are to be shown to the learner. (Goodyear, 1990, calls this 'pedagogical content knowledge'.)

Information about possible misconceptions may also be included in this additional knowledge base, as well as domain information relevant for creating a learner interface.

Summarizing, the additional knowledge that should be included in the domain representation of an ISLE should consist of:

- A conceptual/operational model of the domain
- Instruction knowledge
- Learner (diagnosis) knowledge
- Interaction knowledge

Together these four knowledge sources will be called the Cognitive model. The structure of this cognitive model is elaborated in Van Joolingen and De Jong (1991).

8. Theme 2: Learning goals and internal characteristics of simulations

In Van Berkum et al. (this volume) a classification of possible learning goals has been given, identifying three dimensions for the classification of learning goals: a knowledge representation dimension, a knowledge category dimension and a knowledge scope dimension. In the present section we will investigate the consequences of the choice of a certain learning goal for the internal characteristics of the simulation, the model, the scenario and the experimental frame.

The consequences of the knowledge category dimension are clearest for the model and the sce-
nario. The learning of conceptual knowledge is learning about the structure and behaviour of the underlying model. Of course the experimental frame should allow the model entities to be recognized as a representation of a part of the real system. The learning of operational knowledge is concerned with the scenario. Now the model is not as important as the sequence and kinds of operations that are performed with it. However, in order to be able to perform the right operations some knowledge of the underlying model is necessary, at least in a qualitative way.

For the knowledge representation dimension it is harder to see what the consequences are for the internal characteristics of the simulation. It is tempting to state that a declarative knowledge representation is associated with a quantitative model and that a compiled representation is associated with a qualitative model but this cannot be true in general. For instance it is possible (and perfectly natural) to teach declarative knowledge about a qualitative model. Also to get 'a feel for the model' does not require the model to be qualitative. The same applies for an 'automatizing of skill' learning goal, which is often taught by the operation of a quantitative model. Possibly this dimension does not have many effects on the model and scenario, but more on the learner interface and, possibly, learner activity.

However, the knowledge representation dimension does set some demands on the cognitive model. For example, when the learning goal is to obtain compiled knowledge of a (qualitative or quantitative) model, the cognitive model must provide 'shortcuts' for thinking. This means that the model will be described only in quite general terms within the conceptual/operational model, with stress on the relations between important variables. On the other hand declarative knowledge requires a more detailed and elaborate representation.

The knowledge scope dimension has relations with the experimental frame. When the learning goal of the simulation is domain specific, the experimental frame should be such that those domain concepts that should be learned are emphasised.

For a more general learning goal, the experimental frame must be chosen in such a way that the relation of which the general character is to be learned, is emphasised. This means that it must be possible to find more than one real system with experimental frames which produce models which share the general relation. The generality of a relation can be emphasised by showing that in a different real system the same kind of relationship is applicable.

In the case where knowledge acquisition skills, like experimentation, are a learning goal, this argument may no longer hold. But it seems reasonable to say that learning of knowledge acquisition skills is more concerned with general relations than with domain specific relations, putting the same demands on the experimental frame used.


In this section the relations between learner activity and learning processes and the internal characteristics of the simulation are discussed. Since learning processes are cognitive actions which result in some physical learner action, we will restrict ourselves here to a discussion of the possible types of learner activity.

In Van Berkum et al. (this volume) several possible types of learner activity are mentioned. In this section the implications of these possible activities for the model and scenario will be analyzed.

9.1. Handles on input variables and parameters

Setting environmental qualities refers to the possibility the user has of altering the values of parameters, both in quantitative simulations and qualitative simulations, where parameters represent a certain quality of the environment of the real system (e.g. it is warm outside or: it is cold outside). Activities of this kind require a 'handle' on the changeable parameters, meaning that the learner should have access to these parameters. Static parameters only need to be accessible at the start of the simulation because they are not subjected to change during simulation time, but dynamic parameters require a more complex input mechanism. First, it is possible
that they are given as a function, which makes it possible to calculate the values at all times. Second, it is possible that they are controlled by the learner (or other agent) during simulation time. This means that their value can be changed at any moment by the operating agent.

**Defining initial conditions for an experiment** requires the possibility of changing the initial state of the model. In scenario terms, this must be done before the simulation time is started.

**Intervening in the system's state during simulation run time** requires that the state variables are changeable by an external agent (the learner) during simulation run time. This means that, inside the model, there must be a representation of this external agent and that this representation must be connected to the learner's input mechanism.

**Selecting and building experimental equipment** requires that input variables of one submodel can be connected to the outputs of another. E.g., the output of a model of a sine wave generator is connected to the inputs of an electrical circuit. This can be done before the simulation run time but also during the simulation time, e.g. by opening and closing a switch.

**Building of an experimental subject** requires that, using small building blocks, a model can be generated of the built subject. E.g. in an electrical circuit the differential equations of the time development are generated from the circuit characteristics.

### 9.2. Scenario freedom

**Deciding on the next step in the interaction** can result in several actions concerning the model, but has, of course, a stronger effect on the scenario. Performing an experiment requires that, at the moment when this is allowed, the learner has one or more of the possibilities mentioned in the previous section and the freedom to choose which variable or parameter to access.

Two special cases need attention. In the first case, the model is in some predefined state (possibly evolving with simulation time) and the learner is allowed to change this state in order to reach a certain goal. To do this, a certain sequence of actions is needed, each action addressing one or more variables or parameters. This kind of simulation is often important when operational knowledge is a major learning goal. The term 'experiment' is not really applicable here.

In the second case, the initial state of the model is undefined and must be defined by the learner. The learners will choose this state in such a way that they can expect their experimental questions to be answered. Now a certain check is needed to make sure that the initial state is defined in a proper way, in order to make sure the simulation will run smoothly. This does not necessarily mean that the experiment itself is fault-free but only that the simulation is supplied with the necessary inputs.

### 9.3. Metacontrol

Metacontrol has its effects on the independent variable(s) (in the strict mathematical sense, see Section 3.1.1). In most cases this will be simulation time. Several types of time control are possible. First a major distinction must be made between the case in which the simulation is performed completely before the results are presented (e.g. in a graph where the value of one or more variables is plotted against time) and the case in which the simulation results are presented dynamically, changing with simulation time.

In the first case no time control (and also other types of learner activity) is possible during simulation time. The only freedom a learner may have is the choice of the starting and end points of simulation time.

In the second case more control is possible. The learner may start and stop the simulation time when s/he likes. This requires that the state of the model is kept in memory during the (real) time the simulation time is suspended. The learner may also adjust the pace of simulation time. This is important when processes are too slow or too fast to make an observation in real time possible. Finally, the learner may also be able to change the direction of simulation time, in order to watch processes go backwards. When the independent variables represent something other than time, the same types of control are possible in principle, but in practice only the first case is likely to occur.
9.4. Control over output

Control over output is concerned with which data is collected and how data is collected. One way of collecting data is by connecting measuring instruments (in a wide sense of the term) to the modelled system. This requires the possibility of connecting the output variables of the model with the models of the measuring instruments. The choice of instrument determines how the data is collected, the choice of variable determines which data is collected.

References

White B.Y., & Frederiksen, J.R. (1987a). Qualitat-
ive models and intelligent learning environments.
In W. Lawler & M. Yazdani (Eds.), Artificial
intelligence and education, Vol. 1, (pp. 281-305).
Norwood, NJ: Ablex.
Model Progressions as a Foundation for Intelli-
gent Learning Environments (Report No. 6686).
Cambridge, MA: BBN Laboratories.
model progressions as a foundation for intelligent
learning environments. Artificial Intelligence, 42,
99-157.
Zeigler, B.P. (1976). Theory of Modelling and