LA-UR- 99-3049

Approved for public release; distribution is unlimited.

Title: WHAT MAKES A SIMULATION USEFUL?

Author(s): STEPHEN G. EUBANK

Submitted to: IEEE Systems, Man, & Cybernetics
1999 Conference
Tokyo, Japan
October 12-15, 1999

Los Alamos
NATIONAL LABORATORY

DISCLAIMER

DISCLAIMER

What Makes a Simulation Useful?

Stephen G. Eubank
Simulation Applications Group
Technology and Safety Assessment Division
Los Alamos National Laboratory, Los Alamos, NM 87545
Email: eubank@lanl.gov

ABSTRACT

Modern computers make possible a new blending of systems, man, and cybernetics in the detailed simulation of large sociotechnical systems. Several such simulations are currently under development at Los Alamos National Laboratory and elsewhere. When deployed, they will affect the daily lives of hundreds of millions of people and the allocation of billions of dollars. *Whether* they are deployed depends entirely on their perceived usefulness, which in turn depends on answers to the following:

- What kinds of questions does the simulation address and what kinds of solutions does it provide?
- How can the solutions be validated?
- Is simulation more cost-effective than other methods?

Answers to these questions lead us to define a useful simulation as one which efficiently provides correct, robust estimates required by decision-making needs, together with well understood variability for the outcomes in hypothetical situations.

This paper examines the implications of this criterion for the design of TRANSIMS, a regional transportation network simulation, and by extension, for simulations of other sociotechnical systems.

INTRODUCTION

The Transportation Analysis Simulation System (TRANSIMS) is a large-scale, detailed simulation of urban transportation networks developed as a decision support tool.[1] Urban planners in the United States are legally required to forecast the effects on various subpopulations of changes in the network infrastructure, but up till now there has been no tool available for making these forecasts.

TRANSIMS simulates the second-by-second movement of millions of individuals making their way over the course of a day through a multi-modal regional transportation network resolved down to roughly 10 million 7.5 meter cells. For input, it requires samples of a population's demographics, regional land use patterns, and a detailed description of the transportation network. From these, it generates a synthetic population, gives each individual a set of daily activities, determines detailed transportation plans (mode choice and route) for each, and follows them through the network as they execute their plans.

TRANSIMS shares the following important properties with other simulations of similarly complex systems such as the electric power grid and communications networks:

- the dynamics depend on decisions made by a large number of independent agents;
- interactions among the agents are primarily short-range or local in space and time, mediated by a network, but they also have a non-negligible long-range component;
- the available resources, demands placed on the resources, and the network itself are non-stationary and often far from equilibrium.

Typically, development of simulations has focused on modeling the underlying physical system - the flow of traffic through the network, in this case - and indeed the model itself is often referred to as the simulation. We argue here that this is not correct. The decision-making problem at the heart of such large scale simulations necessitates wrapping the model in a learning algorithm. It is fair to say that so-called "simulation" software is useless without the learning algorithm.

By itself, the model of a system provides a Newtonian clockwork picture. Getting valid results from the model requires a valid initial state, but detailed information about the initial state is not available as input. Furthermore, validity of the initial condition depends on the use to be made of the results. The learning algorithm must estimate an initial state from the available data, known constraints on the results, and dependencies between them arising from the model.

In the discussion below, we will consider a discrete time simulation from time 0 to a final time T. We distinguish among the input data set, D; the time-dependent state of the dynamical system being modeled, $S(t)$, the individual state of each agent within the model, $s_i(t)$ which together constitute the state of the model, $S(t) = \{s_1(t), ..., s_N(t)\}$; the model M which maps $S(t)$ to $S(t+1)$; the actual joint probability distribution of all the agents' states at each time, conditioned on the input data, $P(S(0), ..., S(T) \mid D)$; and the "results" R, which can only depend on the input data D and all the states $S(0), ..., S(T)$.

For example, in TRANSIMS the input data, D, consists of two qualitatively different kinds of data:

1) information about infrastructure - a detailed road map, transit schedules, etc.
2) information about populations - a set of probability distributions for demographic variables such as income, age, and education.

The first kind of data is in principle readily available, since it is required for design and control of the system under study and varies on a slower time scale than the population data. In contrast, because of privacy concerns and the cost of collecting data, detailed information about the population is not available. In particular, although the marginal distribution of demographic variables is available for the entire population, the joint distribution is available only for a few small sample subsets.

The ith agent's state $s_i(t)$ depends on what sort of entity the agent represents and the level of abstraction in its representation. In TRANSIMS, the most obvious agents are individual travelers. Each traveler's state includes dynamical variables such as position, and velocity as well as other state variables such as the traveler's next destination or aversion to traffic congestion.

However, there are other entities in the model, with different state variables capturing other kinds of time dependence. For example, the network topology includes time-dependent traffic signals, reversible lanes, and costs associated with resource usage, such as parking or tolls. All these elements can be included in our description as agents with associated states.

A typical question which might be posed to the TRANSIMS simulation is: what is the effect of a change in parking fees on mass transit ridership, and what subpopulation shows the largest change? The result of the simulation in this case is *not* the final state of the model, $S(t)$. Instead, the result, R, must be deduced from all the intermediate states of the model, specifically, the demographics of travelers who choose to use mass transit at each time step.

QUESTIONS AND ANSWERS

Simulations are not magic. It would be incorrect to claim that TRANSIMS contains the behavior of specific individuals. Only the ensemble of agents' states has any meaning. This ensemble is just one sample from the underlying joint probability distribution P over all the agents' states conditioned on the input data, Each run of the model generates a sample from this distribution. Statistics can be evaluated on each sample and an estimate of the distribution P quantifies the variability in those statistics. That is, if q denotes the estimator used to evaluate the statistic Q from the states of the model, so that $Q = q(S(0), ..., S(T))$, the simulation is used to construct $P(Q \mid D)$ induced from $P(q(S(0), ..., S(T)) \mid D)$.

The distribution P can be used for system identification, answering the question "How does the distribution of Q compare to the distribution expected for this system?". It can also be used for extrapolation – "As the system evolves, how does Q change?". And it can even uncover control parameters not explicit in the input data – "How does Q change for different realizations of the initial state, all equivalent with respect to the input data?".

Sometimes $P(Q \mid D)$ is all that is required for decision-makers. When q is a function which accumulates distance traveled by each vehicle driver, for example, $P(Q \mid D)$ will describe how proposed infrastructure changes affect the total number of vehicle-miles traveled. Other kinds of questions require a sightly different analysis. For example, one might want to know the fraction of emissions accounted for by vehicles which travel farther under the infrastructure changes. This depends on the type and condition of the vehicle, as well as on how it is driven - all factors which are likely to be related to the demographics of the owner and driver. In this case, q would be an indicator function determining membership in Q, the set of drivers who drove farther under the hypothesized scenario than under current conditions. The demographics would be obtained by evaluating the demographics of the sub-population Q to find $p(D(Q) \mid D)$.

This view has several obvious implications for the design of a simulation.

Stochasticity
There must be a stochastic element in the simulation or else the sampling process will yield only a single value. The only places to insert randomness into a model are in the dynamics or the initial condition. For the simulation as a whole, this translates to the agents' decision-making processes and the learning algorithm referred to in the introduction.

Experimental Design
With N on the order of 10^6 and T on the order of 10^5, as in TRANSIMS, P depends on an enormous number of variables. It is not feasible to run the model often enough to sample P in any detail. Careful analysis is required to

identify important sources of variability in the distribution and design experiments to probe them.

Agent Identity

Some agents must maintain their identity in terms of the input data. That is, from the input data we create synthetic agents. each of which is associated with a set of demographics drawn from the input distributions in such a way that all constraints provided by the input are respected. Then, given a subset of agents which meets certain criteria, we can infer the distribution of input variables over that subset.

Efficiency and Agent Abstraction

Decisions generally have a fixed time horizon beyond which they are irrelevant. A typical decision will be based on running the model not just a single time, but many, many times. For agent-based simulations, this constraint places limits on the complexity of the agents and their interactions. Only those aspects of state which are essential to making the final decision should be included. Research is often required to understand exactly what is essential. For example, including a complicated psychological model for drivers in TRANSIMS would slow down the simulation unacceptably. Instead, we have shown how to generate correct distributions of traffic behavior from extremely simple cellular automata (CA) models

The result of a simulation is a set of joint probability distributions for the agents' states which can, in turn, be related to the input data. The state of any individual agent is unimportant - it is like a channel creating dependence between input and output. As long as the channel transmits information with the required fidelity, details of the channel's representation are not important.Furthermore, just as in communication theory, the degree of surprise in the result is a measure of the information content of the message, or in this case the simulation. A simulation which provides low information content is not likely to be worth the effort it takes to develop. On the other hand, too much information overwhelms the decision maker.

In summary, simulations answer questions framed in terms of a correlation structure in the output data or between input and output data. The answer they provide is a set of statistics evaluated on the joint probability distributions of agents' states, conditioned on the input data.

VALIDATING SOLUTIONS

A simulation's validity is often defined as how well it models the dynamics. Not only is this extremely difficult to determine for simulations of complex systems, but also it is far too limited to ensure that the final product is useful.

For Complex Systems

There is a large and growing literature which attempts to classify the kinds of behaviors arising when a large number of simple systems are connected into a complex network. Sometimes it can be shown that the variety of behaviors depends on only a few aspects of the component systems. In these cases it may be possible to validate a simulation, given that its components capture those aspects correctly. In general, though, all that is known is that the behavior a complex system exhibits can be very different from that in any of its components.

Hence there is not yet any clearly superior systematic method for validating the simulation of a complex system. One common approach is to validate system components in isolation, validate the interactions among small subsets of components, and hope for the best when they are all combined.

For example, transportation engineers publish observations of traffic on roadways. These observations are reduced to a set of expectations for traffic on a generic, isolated stretch of roadway, encoded in what is called the "fundamental diagram" of traffic flow. A great deal is known about this diagram and the essential features of models which replicate it.

Clearly, a necessary condition for a traffic flow simulation to be valid is the ability to reproduce this diagram. But this is certainly not a sufficient condition. We can validate the behavior of TRANSIMS on isolated road segments, or include one or a few intersections, or investigate the effect of synchronizing controls across several intersections, but none of this addresses the question of how faithfully the model reproduces dynamics of a large-scale network. [3]

Sub-optimal Input Data

This difficulty in validating a model of a complex system is a well-known problem and must be addressed. But the problem of robustness is even more relevant to the usefulness of a simulation.

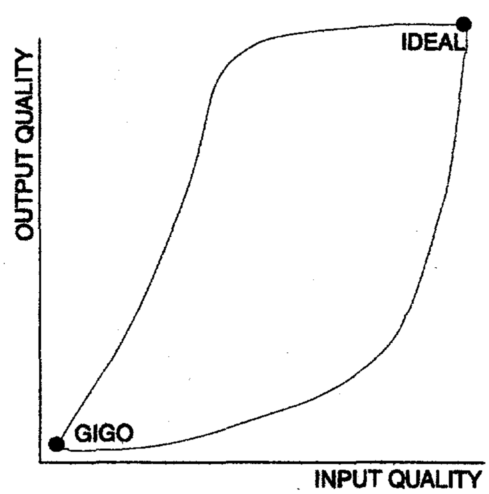

Fig. 1 Example variation in the quality of output as the quality of input changes for two different simulations.

The figure above sketches the operating characteristics of two different simulations in terms of input and output quality. By output quality, we mean how well the simulation reproduces the behavior of the system it is supposed to simulate. By input quality, we mean both the completeness and correctness of what is available.

The ideal operating point for a simulation is in the upper right corner of the figure, where high quality inputs lead to high quality output. A simulation is usually deemed valid if it can operate at this point. The output quality for less-than-optimal input data is ignored under the slogan "garbage in, garbage out", represented here by the point at the origin. But there are important differences among "garbage" input, ideal input, and input available in real world applications. Likewise, there is a real difference between two simulations with the operating characteristics sketched in the figure, even if they provide *identical* output at the ideal operating point. The difference is that only one of the two is useful.

Robustness to less than perfect input data is often an afterthought in the design of a simulation. In a large-scale simulation, it must be among the first design criteria. In the case of TRANSIMS, it has required coupling the traffic model to a learning algorithm.

Together, the two determine sets of self-consistent initial states for the model. There are many open questions about the learning algorithm in TRANSIMS, and these are in fact the focus of most of the research in TRANSIMS. None of these questions can even be posed in the context of the model alone.

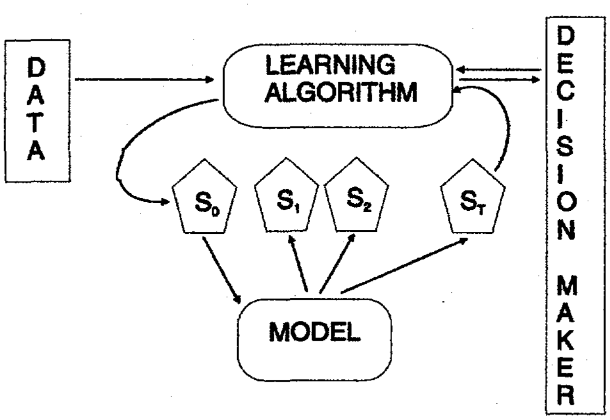

Fig. 2 Architecture of a simulation.

As shown in the figure above, the role of learning algorithms in simulation is like that of a controller, and the "plant" under control is the model. The algorithm evaluates the effects of changes in the initial state on model evolution.

With knowledge of the question the simulation must answer, it can determine which input data most strongly affect the simulation results, and which must be improved to meet required tolerances in variability of the results. Like the model, it is thus a special purpose tool incorporating some understanding of the model's properties and the simulation's goals.

For example, one of the central problems in simulating traffic is to generate realistic routes for travelers. Given the travel times on each link of the network and a set of mode choices, it is possible to find the shortest path through the network for any number of travelers. But each new traveler affects the others' travel times in a way which can be determined exactly only by running the full model. This is a nonlinear, coevolutionary problem, and a direct solution is not currently possible.

TRANSIMS's learning algorithm solves this problem through iterative approximation. First, each traveler is [4] routed using the uncongested network travel times. The resulting plans are simulated and new travel times are determined. Then a subset of travelers is rerouted using the new travel times. The travel times form a sort of mean field approximation to the interaction history of each traveler, which is not sensitive to the route of any individual traveler.

The choice of travelers to replan is an essential part of the learning algorithm - it does not work to replan everyone or just to replan travelers who arrive at their destinations late. Moreover, the goal of the study determines which travelers must be routed correctly. In some cases, for example freight traffic can be modeled much less carefully than commuter traffic with no adverse effect on the result.

This iterative approach has an intuitive appeal - people do not re-plan trips based on avoiding interactions with specific travelers, but on avoiding congestion. It is tempting to think of each iteration as a new "day", but that would be a misinterpretation of the algorithm. Each iteration is nothing more than a step in a learning algorithm. The dynamics of this algorithm may have nothing to do with day to day fluctuations in traffic, but the end state of the algorithm should be one of many possible realistic sets of travel plans.

In general, a learning algorithm involves multiple iterations of the dynamical model with feedback for selectively updating initial states. The dynamical model by itself is useless, since it requires complete knowledge of the initial state, which is typically unavailable. The learning algorithm by itself is useless since it requires feedback from the model.

COST EFFECTIVENESS

There are few alternatives which even attempt to support decision-making in large sociotechnical systems. Analytic methods either focus on the components and ignore the crucial problem of scaling to real systems or idealize away important parts of the dynamics in an effort to make the full system tractable.

Decisions in these systems are often made on the basis of intuition or experience with incomplete knowledge, both of the current state of the system and of the true effects of the decision. Simulations break down the steps for rational decision making in the most natural way:

- estimate the current state of the system (or enough of it to base the decision on) from the available data.
- evolve the state to understand the effects of the decision.

Simulations (as the term is used here: model + learning algorithm) solve these two problems in a cost effective way. The only alternatives for determining the initial state are to improve data collection efforts until the system state is directly observable or directly to estimate the system state from the input data. The former is typically more expensive than developing a simulation, and must be repeated for each different decision, as the representation of the system changes. The latter relies on machine learning techniques which are unlikely to work well in the context of sparse data and a high dimensional problem.

Using alternative methods is even harder for hypothetical situations. Direct observations become impossible and the generalization power of machine learning techniques will be poor given only a few training examples. But simulation systems can handle hypothetical cases as easily as real cases.

Gathering data for simulation represents a *capital* investment} which returns dividends in the form of simulating many different scenarios, rather than an *operating* expense, as does gathering data for model building.

CONCLUSIONS

In summary, simulations can be viewed as algorithms which learn to generate joint probability distributions in many variables from a small sample of distributions in a few variables. The quality of the simulation depends on the degree to which the correlation structure in the generated distributions matches that of the system it is intended to model.

A useful simulation couples a dynamical model of the system to a learning algorithm capable of determining a self-consistent initial state for the model from sparse input data. The simulation then robustly estimates the time evolution of system state distributions and provides an understanding of variability in the solution. The whole package is ideally suited for estimating the effects of infrastructure changes or long-term demographic changes in large sociotechnical systems and for deciding what course of action will bring about desired results. Absent any part, however, it is practically useless.

REFERENCES

[1] C. Barrett et. al., Transportation Analysis and Simulation System (Transims), Los Alamos Unclassified Report 99-1658

[2] B. W. Bush, The Transims Framework, 7th TRB Conf. on Transportation planning Methods. Boston, MA. 1999.

[3] K. Nagel et al. TRANSIMS traffic flow characteristics, LAUR- 97-3530

[4] K. Nagel and C. Barrett Using Microsimulation Feedback For Trip Adaptation For Realistic Traffic in Dallas LAUR 97:1334