Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Discrete, continuous and combined simulation

Paul F. Roth
Department of Computer Science and Engineering
Bi-County Center for Engineering
University of South Florida at Sarasota
Sarasota, Florida 34243-2197 U.S.A.

ABSTRACT

This tutorial paper introduces the major types of simulation models: discrete, continuous, and combined. It concentrates on illustrating the models by describing their variable characteristics, through modeling a liquid waste disposal system, first as a continuous sub-model, next a discrete sub-model, finally as a unified combined model. Methods for implementing these models using various hardware and language tools is discussed and specific reference to certain languages is made. It is hoped that this material will increase the modeler's understanding of the basic, dynamic nature of modeled systems and their variables so as to enhance the initiation phase of a simulation project.

1. INTRODUCTION

Simulation is a way of making experimental measurements on a system represented by a computer-implemented model. This process, in effect, substitutes a simulated, or "target", system for the real system, and permits the experimental situation to have a laboratory context. There are many justifications for using a simulation.

The type of model reflects the type of measurement; it also dictates the method of computer implementation.

In a discrete-events model, measurements are made on variables which are affected by instantaneous changes of system state.

In a continuous-events model, measurements are made on variables which reflect system continuous ("transient") state changes.

In a typical combined - or "hybrid" - model, measurements are made on the above variables where the system is affected by both discrete and continuous state changes.

This tutorial paper introduces the various formulations by discussing a specific model, its behavior, and its variables. A narration of its operation provides further understanding.

2. SYSTEM STATE CHANGES

System state changes can best be illustrated and explained by a specific system example involving the dynamic behavior of liquid waste storage. Two subsystems, or sub-models, comprise the system. One subsystem, involving the tank volume dynamics, encompasses continuous changes; the other, involving truck unloading, encompasses discrete changes. The total combined system exhibits both kinds of behavior.

2.1 Continuous

Given a liquid waste storage system, consider a holding tank with input and output flow orifices. Flow into and out of the system via these orifices is continuous; the rates of in- and out- flow during some time interval can cause variation in the tank volume or level. Thus volume or level is continuous function of input and output. When the input/output exhibit variations, the level will show a corresponding transient behavior; when the input/output is stable, the level will stabilize. Measurements of interest might be the continuous tank level, or might be the time at which the transient level increases or decreases through some threshold level, or reaches stability.

Figure 2 illustrates the continuous model. The variables depicted at the bottom of the figure represent the input and output flow rates: input is positive in magnitude; output is negative. Assume both are of the same magnitude. The variable at the top depicts the level of the tank, which starts at an initial level and then begins to fall at a fixed rate as it is perturbed by the step like outflow rate. After a time the outflow forcing function is removed, causing the tank level to respond by settling to a constant, but lower, level. The next transient behavior depicted is the result of applying a step of input flow rate, causing the level to rise. At some time after inflow commences, a step-like application of outflow is introduced, which causes an opposing transient response in the tank level variable resulting in a stable level when the opposing rates cancel each other out. Finally the inflow is terminated, resulting in a net outflow rate, which results in a decreasing level once again.

2.2 Discrete

To illustrate a discrete-change model, consider the liquid waste tank-loading operation, functionally external to the storage tank-level-dynamics model. This sub-model comprises a system of "honey-dipper" trucks and unloading docks, or bays. Each truck carries a specific load volume and must enter and use a bay for a specific time to execute unloading. If a truck enters this system, requests use of a bay and finds all bays occupied, it must wait its turn, i.e., must join a queue, or waiting line. Therefore, each truck which enters the system can be described as going through various changes-of-state as it traverses the unloading operation:

Occurrence, or arrival
Request for bay (concurrent with arrival)
Acquisition of bay (may be concurrent with request): initiation of service
Release of bay: completion of service
Departure (concurrent with release)

Each of the state-changes is a discrete event, i.e., happens instantaneously, and leads to a constant state for each truck:

Waiting (if no bays available)
Unloading

Of major impact, therefore, are the following times for each instance, or occurrence, of a truck in the system:

Time of arrival
Elapsed waiting time (bay allocation time - minus arrival time)
Elapsed service time (bay release time - minus allocation time)

Measurements of interest might be the average time (over all trucks) spent in queue; average number in queue; average utilization of bays. Control variables might be: the rate of truck arrivals; the time spent in truck unloading; the number of bays available. Note that the measurements only involve the states themselves: change-of-state transients are not considered.

Figure 1 depicts the discrete-event phenomena described in the discussed truck-unloading sub-model. The row of arrows dispersed in time represents the truck arrival and departure events which denote the system's changes of state: truck arrival instances caused by a random process; truck departures the result of the end of each unloading operation. Events of the arrival type are termed "exogenous"; of the departure type, "endogenous". The system contains one unloading bay.

Also shown in Figure 1 in the appropriate sequence-related to the state changes - are the states of individual trucks (system users) and the state of the unloading bay (system resource). Of special importance is the "waiting" state of Truck 3 caused by the overlapping of the Truck 3 arrival event and the "unloading" state of truck 2. In effect, Truck 3 goes into a queue, or waiting line, to await the release of the bay by the completion event of Truck 2.

To compare with the continuous model type, what is not important in a discrete-event model is any transient behavior which might be induced by the abrupt state changes. If such actually exist, they are assumed to be outside the dynamic range of the discrete-event model, or otherwise uninteresting for the purposes of this formulation.

2.3 Combined

To illustrate a combined - or hybrid - model, consider the behavior of the truck unloading subsystem which unifies the discrete and continuous sub-models.

In this case, assume as subsystem linking mechanisms:

Tank input (continuous) on-and-off controlled by truck off-loading events (discrete);

Truck bay closing (discrete) affected by sensing tank threshold level (continuous);

Tank output on-and-off controlled by sensing tank threshold level.

Each truck in an unloading bay unloads at a constant flow rate for a time duration which reflects its load volume. This acts as a perturbation input of volume-rate to the tank dynamics model; the total input flow is the sum of rates produced by all the trucks unloading at a given time, in the case of a multi-bay model. Thus the discrete-event unloading subsystem model acts as a driving function for the tank dynamics subsystem model.

The tank outflow is controlled by a valve which is caused to be open or closed based upon some operating policy. Thus the tank level reflects the balance (as before) between input and output flows, but in the combined case the dynamics of the truck unloading subsystem interact with the tank subsystem.

A mechanism linking the two subsystems might be to sense the tank volume as it increases through some threshold level to effect the cessation of truck unloading (close the bays) and vice versa to commence the unloading.

Measurements of interest might be the tank volume profile given a certain ensemble of control variables: truck arrival rate, number of bays, increasing and decreasing tank threshold levels, etc.

Figure 3 depicts the resulting combined continuous/discrete event system phenomena for the one-bay model. The topmost display presents the time history of the tank level, a continuous variable which is perturbed by discrete events and state changes. Dashed lines indicate important, sensed threshold levels. The bottom display presents a row of arrows indicating, as before, truck arrival and departure events. In the middle, time bars depict the state of various trucks.

Initially, level is a at steady-state value. At some time later, a discrete event occurs as Truck 1 captures the bay and enters the "unloading" state. This imparts an input-flow rate perturbation to the tank level sub-model, which results in a time-wise increase of level. Meanwhile Truck 2 arrives and, finding the bay occupied, goes into queue, or "waiting" state. Truck 3 does the same shortly thereafter which results in a waiting line of two trucks.

Eventually Truck 1 finishes unloading and leaves, allowing Truck 2 to occupy the bay and commence unloading. We assume, for purposes of this model that the transition between the two trucks is instantaneous, causing no observable transient behavior of the tank-level variable. Meanwhile, Truck 3 continues to await its turn in the bay.

When the tank control system senses that Level 2 has been reached on an increasing slope, the system bars further access to the unloading bay by trucks in the waiting line, without interrupting the currently unloading truck.

When the tank control system senses that Level 1 has been reached on an increasing slope, it opens the outlet valve causing an equal and opposing flow rate to be imparted. This has the initial effect of causing the tank level to attempt to reach a new steady-state, but this is overtaken by the completion of off-loading of Truck 2, which results in a net outflow rate perturbation.

The resulting steady decrease of level continues until the control system detects that Level 2 has been crossed in a decreasing direction, which causes the system to reopen the unloading bay. At this point, Truck 3 which has been waiting - both for Truck 2 to finish and for the bay to reopen (dashed lines) - acquires the bay and starts unloading. This action imparts a positive flow perturbation thus negating the outflow rate and the tank level attains a steady state value.

The next event, the departure of Truck 3, once again restores a negative flow at the tank, causing the level to decrease until the control system senses that Level 3 has been reached during negative flow. This causes the outflow valve to be shut off, allowing the tank to achieve a steady-state level once again.

Note that this combined model has interleaved the effects of discrete events and continuous events to achieve a simulation of the performance of an integrated tank and control system. Variables of interest to the decision maker may be the settings of the sense levels to effect some kind of control policy.

3. MODEL IMPLEMENTATION

There are various ways of implementing the model so that experiments can be run to produce data. Note that the data represents the performance of the model only. It only approximates data which would be realized by the real system, with some degree of difference caused by the simulation process; this is known as the degree of validity.

Most models are implemented by programming digital computers and executing the programs to produce the data. Modeling, programming, execution, analysis - this is the process of simulation.

3.1 Continuous Model Implementation

In the case of continuous models it is possible to use an electronic circuit analogy for the system to be modeled in place of a program; that is, the circuits describe the same dynamic equations as the "target" system. This is called an analog computer and constitutes a special case; otherwise continuous models are usually implemented by performing numerical integration of the dynamic differential equations describing system dynamics. This method of implementation produces results of far greater precision than the analog method, which may be useful where rough approximations are adequate. Analog computers are usually only found in specialized engineering environments. Measurements are made by recording the changing values of appropriate system variables during various time intervals.

3.1.1 Languages for Numerical Integration

Languages appropriate for implementing numerical integration vary over a wide range. In practice, the range goes from standard algorithmic languages, such as Fortran, to specially designed, problem-oriented languages, providing much user support but specific only for this purpose. In cases of large sets of simultaneous differential equations, array processors may be useful for speeding up the processing.

3.2 Discrete Event Model Implementation

In contrast to continuous models, discrete event models are implemented by representing model variables and dynamics by numerical and character-string symbols. In a typical implementation, data structures which represent each instance of a system user contain symbols representing resources required, clock time, activity status, etc. These data structures join and leave link lists which represent queues and event-time-ordered sets as the user goes from state-to-state during its transitional life (from birth to death). Statistical variations are introduced by selecting numerical values selected from random variates generated by user-defined sampling rules. Measurements are made by recording the elapsed times incurred by system objects (users, resources) between state changes, or the event frequencies themselves. Raw data is usually reduced to statistical averages or distribution for evaluation use.

3.2.1 Languages for Discrete Event Models

Once again, a wide range of languages can be utilized. General-purpose programming languages, such as Fortran, provide the modeler with complete flexibility of expression. Problem-oriented discrete-modeling languages are available, but may cause the user to trade flexibility of expression for user-friendly programming features. Discrete-event languages can be classified by their mode of input expression: data files or statements. GPSS and SLAM incorporate a succinct file of command symbols to effect the model; Simscript and Simula require a statement syntax to represent the model. At least one language, Conversim, now under development, facilitates programming of a model via a questionnaire implemented on screens and windows.

3.3 Combined Model Implementation

Combined models require programming features representative of the above types plus the capability to represent the linking mechanisms. Such mechanisms include continuous variable threshold - level - and - direction detectors to initiate discrete state changes, and discrete-event-actuated switches to initiate continuous variable perturbations.

3.3.1 Languages for Combined Model

Combined programming can be accomplished using a wide range of computational techniques. Hybrid computers, combining analog continuous models and language-implemented discrete models can be used, subject to the previously-discussed limitations of analog computation. Basic programming languages can, of course, represent the combined model. There also exists a class of problem-oriented language which greatly facilitate the expression of combined models. The language, SLAM II, is representative of this type, having all the modeling constructs necessary effect the combined model programming job.

4. REFERENCES

Pritsker, A.A.B. (1986). Introduction to Simulation and SLAM II, Third Edition. Halsted Press, New York

Roth, P.F. (1983). Simulation In: Encyclopedia of Computer Science and Engineering (A. Ralston and E. Reilly, eds.) Van Nostrand Reinhold, New York.

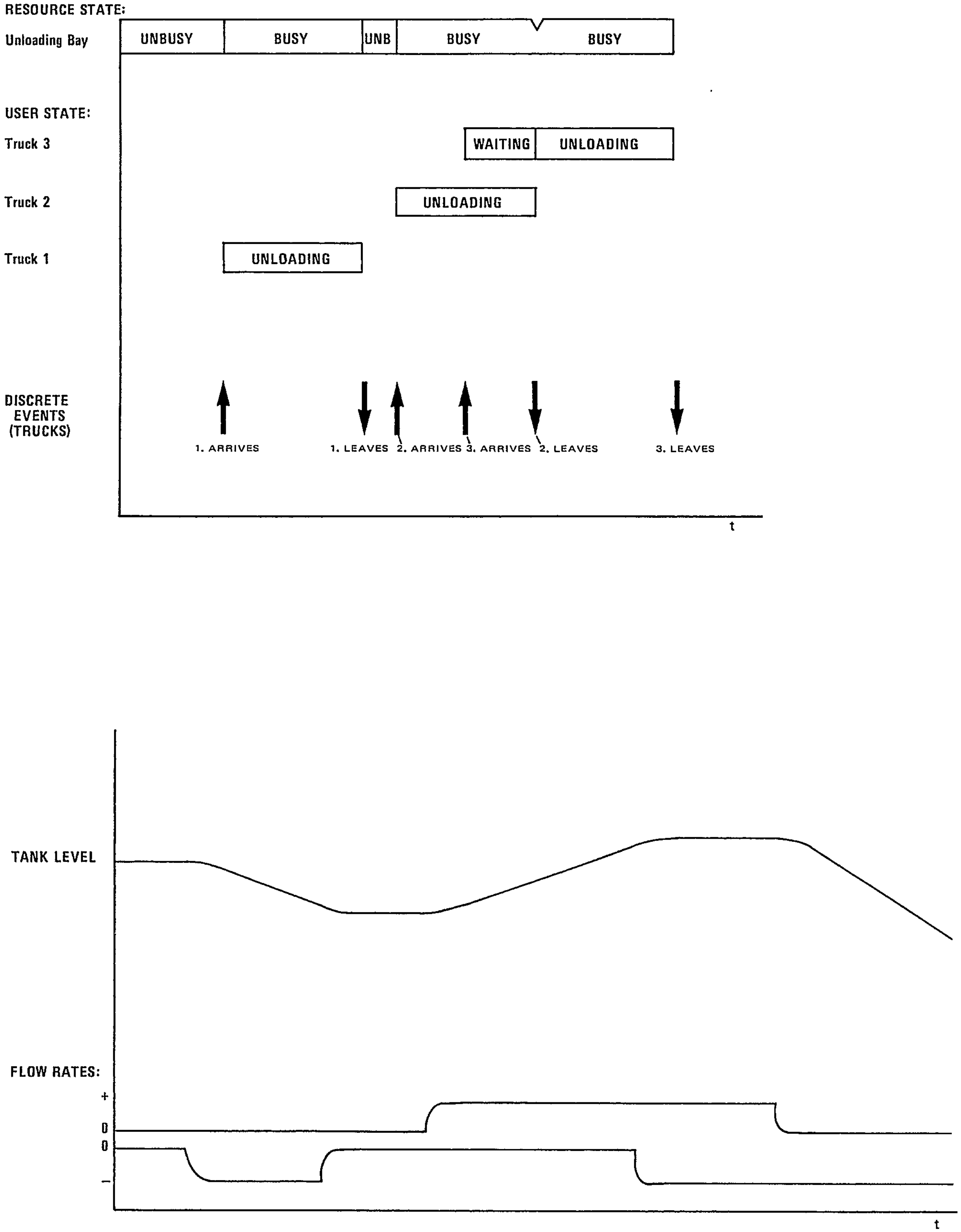

Figure 2 TANK LEVEL MODEL CONTINUOUS STATE CHANGE

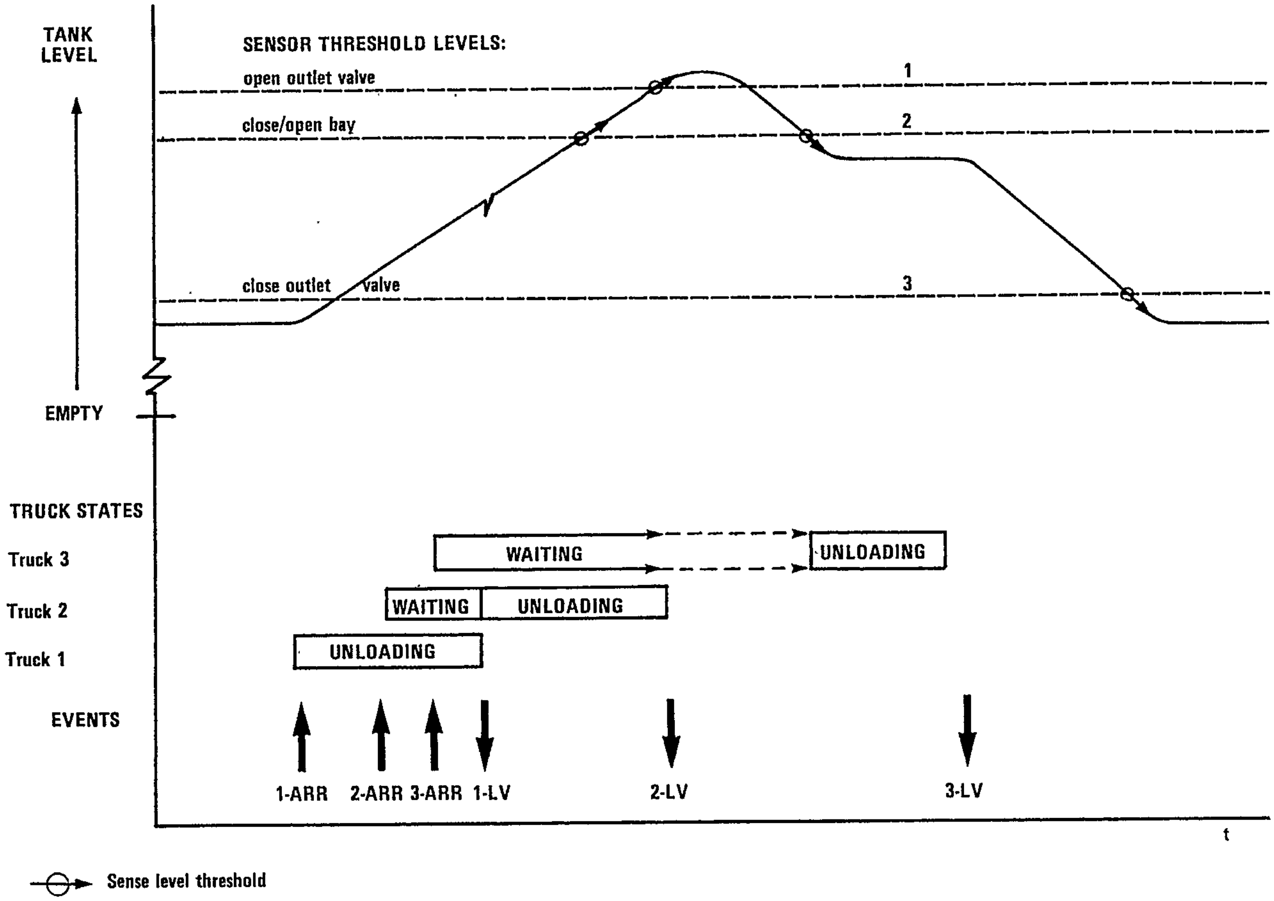

Figure 3 TANK/TRUCK SYSTEM COMBINED CONTINUOUS/DISCRETE EVENTS

Author's Biography

Paul F. Roth is currently Distinguished Lecturer in Computer Science and Engineering at the University of South Florida. He received a B.S. in Physics from the University of Pittsburgh and an M.S. in Engineering (EE) from the University of Pennsylvania. He has been involved in simulation, both continuous and discrete, for over 30 years, with service in industry, government, and academe. His most recent research involves the simulation of computer networks for performance evaluation. He has participated in the development of the discrete-event languages, BOSS and Conversim. He is currently writing a computer - science - oriented book on discrete event simulation. He has twice served as Chairman of the ACM Special Interest Group in Simulation (1975, 1985) and has rendered nearly continuous service in various capacities to the Winter Simulation Conference since 1970. He is a member of ACM and the IEEE Computer Society.