ENTITIES WITH COMBINED DISCRETE-CONTINUOUS ATTRIBUTES IN DISCRETE-EVENT-DRIVEN SYSTEMS

Kristina Dammasch, Graham Horton[(a)]

[(a)]University of Magdeburg, Department of Simulation and Graphics

[(a)]kristina@sim-md.de, graham@sim-md.de

ABSTRACT

Whether a real system is approximated in a discrete or continuous simulation model depends on the more predominating type of variables determining the system's behavior over time. If both discretely and continuously changing variables have to be considered, hybrid simulation techniques have to be applied, where discrete events have an influence on the otherwise continuously changing system state. But in hybrid as well as continuous models the involved entities are usually seen as some kind of evolving fluid. If the individual entities of the system and their attributes are of particular interest this approach might be inappropriate. The paper presents a different approach to hybrid simulation based on colored stochastic Petri nets. It enables the modeling of distinguishable entities with both discrete and continuous attributes in an otherwise discrete-event driven system. An example is used to demonstrate the principles and show possible application fields.

Keywords: Discrete-Event Systems, Hybrid Simulation, Colored Stochastic Petri Nets, Psychosomatic Attributes

1. INTRODUCTION

Computer simulation is the emulation of a real or planned system in a computer program for studying the functionality and effectiveness of the system. In most simulation studies, the system's behavior over time is of particular interest. It is described by the alteration of involved variables considered as relevant for the system's behavior and thus determining the system state. These state changes are either modeled discretely, if the involved variables change at countable points in time, or continuously, if they change continuously over time. If a state change is caused by the occurrence of discrete events, the system is referred to as discrete-event system. Furthermore, it is possible that a system state is determined by both discrete and continuous variables. In that case hybrid models combine the features of discrete and continuous systems.

The description of a system also includes the involved entities. Entities in discrete-event systems are single objects that can move around within the system boundaries. They can either have no particular characteristic or be described by discrete attributes. In industrial applications where simulation is wide-spread, material and equipment involved in the production or distribution processes that has to be modeled are often represented by entities. Implemented attributes are usually simple discrete characteristics such as availability, destination in the process or the probability of defects. In continuous systems the involved entities are usually seen as some kind of evolving mass changing or moving fluidly.

But there are simulation studies requiring a more detailed model of the individual entities and all their relevant attributes. This might be the case if the entities themselves and their behavior in the system environment are of particular interest as it is often found in applications fields such as industrial psychology or health care. Furthermore, the modeling of not only several discrete but also continuously changing entity attributes might be necessary. Considering industrial psychology, continuous attributes could be psychosomatic parameters such as fatigue, alertness or performance. These can have influences on global system variables and therefore on the quality of the process and, eventually, on the overall system output. In health care applications, patients play a decisive role and, depending on the objectives of the study, their characteristics have to be taken into account for a system analysis. This might also include not only discrete parameters such as diagnosis or length of hospital stays but also continuous ones such as mood or compliance with treatment.

If these influences have to be regarded, we need a combined discrete-continuous simulation. But existing techniques only include hybrid system variables and not hybrid entity attributes. For that reason we would like to introduce a new approach to hybrid simulation models enabling the modeling of entities with both discrete and continuous attributes in an otherwise discrete-event system.

2. METHODOLOGICAL APPROACHES

Discrete-event simulations approximate systems as they evolve over time, where the system state changes instantaneously at countable points in time. These points are determined by the occurrence of discrete events that may change the state of the system. This state is described by all variables and attributes required for giving a complete image of the system at a particular time relative to the objectives of a simulation study. For each objective only the relevant and necessary details have to modeled (Banks, Carson, Nelson and Nicol 2001). The time when each type of event will occur next has to be stored in an event list. The simulation run is continued as long as there are scheduled events in the list.

Petri nets are a common paradigm for modeling such kinds of systems (Wang 1998). The complete model is represented as a graph consisting of places (states) and transitions (state changes due to occurring events) that are connected by directed arcs. The places of the net can contain any number of mobile elements of the system, referred to as tokens. These tokens are moved from place to place by the "firing" of the transition representing an event occurring in the system. So called immediate transitions fire instantaneously when becoming enabled, whereas timed transitions fire with a certain time delay. Colored Petri nets extend the concept by adding attributes to the tokens (Jensen 1997). The firing of a transition not only changes the distribution of tokens but can also modify the attributes of the tokens. With colored Petri nets, discrete-event systems can be modeled containing distinguishable mobile entities with specified attributes.

In continuous simulation models, the state of the system is changed by continuous processes, this means that the state variables change continuously over time. Typically, the dynamics of continuous state variables are modeled by differential equations specifying the rates of change of the variables over time. If these equations are simple the whole system can be solved analytically without a simulation. But for most continuous models numerical-analysis methods such as Euler or Runge-Kutta integration have to be used for solving the differential equations numerically (Law and Kelton 2000).

In hybrid simulations, combining discrete and continuous approaches, discrete events can have an effect on the otherwise continuously changing system state. Continuous or hybrid models can be used for modeling discrete-event systems if the number of involved entities is very large. In this case an aggregated mass of entities is “flowing” through the system. Fluid Petri nets are an example for such kind of modeling technique. Instead of discrete tokens one or more places can hold fluid tokens (Horton, Kulkarni, Nicol and Trivedi 1998) that may be used to approximate a large amount of discrete tokens. The main disadvantage is that entities are no longer distinguishable from each other and individual attributes and behavior are not observable.

One of the most important advantages of Petri nets in general is the graphical representation giving a clear overview of the system. Not only relations and interdependencies between involved events and elements are illustrated but also the movement of entities through the system during the simulation run. But the existing approach to a hybrid Petri net disregards individual entities which is insufficient for some kinds of applications.

Agent-based simulation considers entities with several related discrete or continuous attributes, but the concept of intelligent agents covers much more functionality than a detailed description of the entities' characteristics (Macal and North 2005). Other functions such as decision making or machine learning often do not have to be considered when observing a discrete-event system and its behavior. As colored Petri nets capture a more appropriate amount of detail, we would like to establish a new approach for hybrid simulation based on this paradigm but focus on a more detailed modeling of the mobile entities.

3. ENTITIES WITH HYBRID ATTRIBUTES

For modeling a discrete-event system with mobile entities described by hybrid attributes we extend the existing concept of colored Petri nets by a new token definition. Instead of implementing only discrete attributes we would like to add continuous parameters. Therefore we now refer to the extended token as a hybrid token. In addition, the token has to have functions computing the continuous change of these attributes, typically, implemented by differential equations describing the rates of change over time. That way, our modeling approach combines the idea of hybrid simulation with still distinguishable entities moving around in the system.

In order to explain the idea in detail we have to take a closer look at colored Petri nets and their basic components. The most important modification is done to the colored tokens of the net. As shown in Figure 1 the new defined hybrid token contains not only discrete and continuous variables but also functions describing the changes in the values of continuous attribute. Therefore a hybrid token has its own attribute dynamics which are independent of system events. Regarding that the continuous attributes are not to be precomputed the integration steps have to be included in the global event list for proceeding the simulation run.

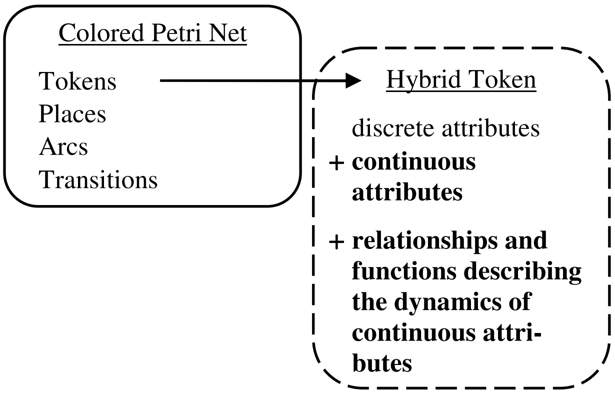

Figure 1: Components of Stochastic Petri Nets with Hybrid Tokens

Considering the hybrid token extended by continuous attributes as well as dynamics, there are several interactions possible between different token attributes. They are no longer only influenced by occurring events but by changing values of other token attributes. The following interdependencies are possible:

- A change in the value of either discrete or continuous attributes may cause a change in the value of a continuous attribute.
- A change in the value of either discrete or continuous attributes may cause the relationship governing a continuous attribute to change at particular time.
- A continuous attribute achieving a threshold value may cause a change in the value of another continuous attribute.
- A continuous attribute achieving a threshold value may cause the relationship governing another continuous attribute to change at particular time.

All other components of the Petri net remain unmodified but their functionality has to take the attribute dynamics of the hybrid token into account. Transitions can be enabled either by values of discrete or continuous token attributes and in the same manner change all attribute values as well as attribute dynamics when an event occurs.

There are also several types of possible interactions between system events and hybrid tokens. The definition follows the existing concept of combined discrete-continuous simulation (Pritsker 1995):

- A discrete event may cause a discrete change in the values of discrete or continuous token attributes.
- A discrete event may cause the relationship governing a continuous token attribute to change at particular time.
- A continuous token attribute achieving a threshold value may cause a discrete event to occur or to be scheduled.

The modeling approach of hybrid tokens adds attribute dynamics to the Petri net but keeps it separated from already existing dynamics due to the occurrence of system events. The advantages and disadvantages of the approach will be discussed using an example where human beings have to be modeled as entities in a dynamic environment.

4. MOTIVATING EXAMPLE

Assuming that our modeling approach is of particular interest for systems where human beings and their attributes have a noticeable effect on the system state, we would like to present a small illustrative example describing the behavior of human beings in their environment. First, the underlying Petri net is presented followed by details on how the idea of hybrid tokens had been implemented.

4.1. Petri Net of a Simple Queue-Server Model

The Petri net describes a simple queuing system consisting of one queue and one server illustrated in Figure 2. It is intended to model a simplified waiting behavior of customers in a queue and under which conditions a customer will leave before being served due to a loss of patience.

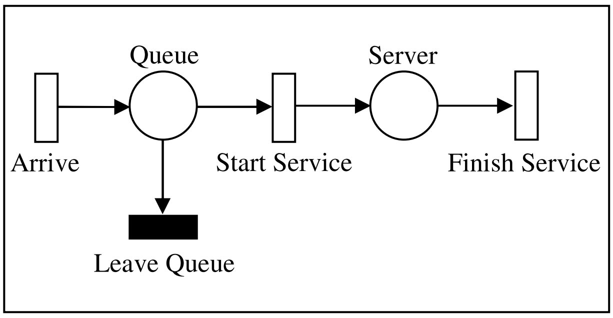

Figure 2: Petri Net for a Simple Queue-Server Model

The places “Queue” and “Server” both can contain hybrid tokens of type “Customer”. A customer in this system is characterized by the following attributes:

- patience (continuous)
- threshold (discrete)
- loyaltyToServer (discrete)
- isLeaving (discrete)

The parameter “threshold” specifies the lowest value “patience” is allowed to reach before the customer will leave the queue early. The logical attribute “isLeaving” marks this state and is set to “TRUE” whenever this case occurs. The loyalty of the customer to the server has an influence on the value of the threshold. A high loyalty (parameter is set to “TRUE”) is associated with a low threshold that will cause longer waiting times, and a low loyalty (parameter is set to “FALSE”) is associated with a high threshold that will cause shorter waiting times. Furthermore, functions need to be defined governing the evolution of “patience” over time as well as the relationship between “loyaltyToServer” and “threshold”. For the rate of change of the patience we assume a simple ordinary differential equation modeling a slow but continuous decrease of the value. An example behavior of the variable over time is shown in Figure 3.

The transition “Start Service” is enabled when “isLeaving” is “FALSE”, that is when the customer enters the queue. But as “Start Service” is a timed transition, the event marking the beginning of the service will not occur immediately but with a stochastic time delay. While time is advancing it is possible that the value for the customer’s patience will fall below the defined threshold and the attribute “isLeaving” is set to “TRUE”. This enables the transition “Leave Queue” that will fire immediately and thus remove the according token from the queue. As a consequence the scheduled event for the beginning of the service has to be removed from the event list.

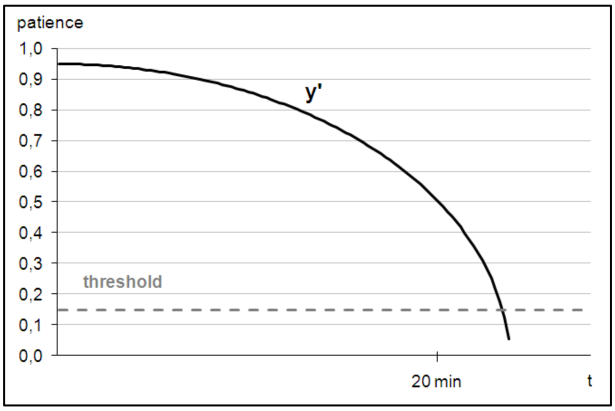

Figure 3: Example Behavior of the Continuous Variable “patience”

If the transition “Start Service” can fire, the token will be removed from the queue and added to the place “Server”. After a certain amount of time the service will be finished and by the firing of “Finish Service” the token will be removed from the place “Server”. The four token attributes have no influence on the timed transition.

4.2. Details on the Implementation

All attributes of the customer are initialized when the transition “Arrive” fires and creates a hybrid token. The logical attribute “isLeaving” is always initialized as “FALSE” so that the customer will never leave immediately after entering the queue. All other attributes are randomly distributed since the individual customers are to have different characteristics.

The loyalty to the server is simplified set to “TRUE” if there is some kind of loyalty, as it is observed in regular customers (Brown et al. 2003). The variable “loyaltyToServer” is set to “FALSE” if there is no particular loyalty. The ratio between being loyal and not being loyal is one to one. The initialization of this attributes also determines the threshold for the patience of the customer: In the case of loyalty the threshold is uniform distributed between 0.0 and 0.25. If the customer is disloyal the threshold is uniformly distributed between 0.26 and 0.5.

The continuous attribute “patience” is also normalized between zero and one and its initial value is assumed to be uniform distributed with a lower boundary of 0.4 and an upper boundary of 1.0. The ordinary differential equation governing the evolvement of the parameter “patience” over time is defined as

$$\frac{dy}{dt} = -c * \frac{t}{y} \qquad (1)$$

with y defined as the value of the attribute “patience”. The coefficient c in the equation is also uniformly distributed with a lower boundary of 0.0002 and an upper boundary of 0.0005 influencing the amount of decrease of the customer’s patience.

This differential equation describes a steady decrease over time and we choose to implement Heun’s integration method that is sufficient for this simplified example. The step size used in the integration is constant and has to be defined by the user prior to the simulation run. We set the step size to 5 seconds resulting in an additional entry in the event list of the simulation model: Every 5 seconds the event “update patience” has to be executed where the integration method is executed for every hybrid token waiting in the queue.

After updating each token, a second method checks if the new value of “patience” is below the defined threshold. In that case the attribute “isLeaving” is set to “TRUE” causing the transition “Leave Queue” to fire when the update process is completed for all tokens. The step size has an influence on the accuracy and the error made when detecting the point in time when the threshold is reached.

In addition to the steady decrease of the customers’ patience it is possible that the occurrence of a discrete event may cause the value of “patience” to change. An example for such an event might be the opening of a second server. The modified Petri net is illustrated in Figure 4.

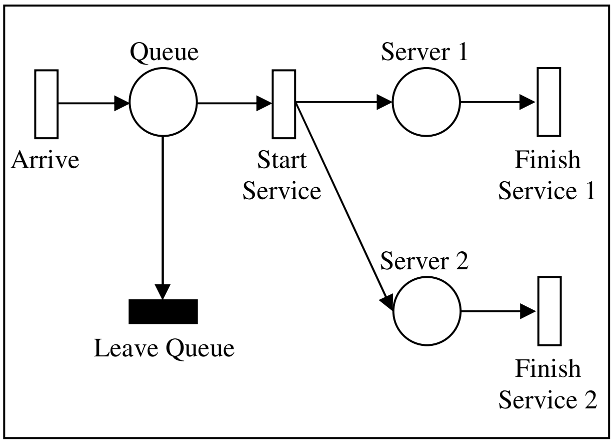

Figure 4: Petri Net with a Second Server

The continuous attribute is changed, depending on the event either immediately or scheduled, and afterwards the integration is continued based on the new value of y. Figure 5 shows a possible influence of the event on the value of “patience”. The opening of the second server leads to an increase in the customers’ motivation to remain in the queue.

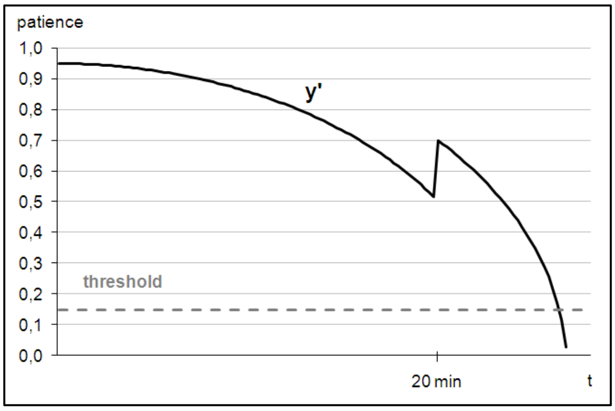

Figure 5: A Discrete Event Causes a Discrete Change in the Value of "patience"

Furthermore, it would also be possible to model the influence of a discrete event by changing the differential equation (1) governing the behavior of "patience". The opening of a second server might instead cause the parameter *c* to decrease. This would result in a slowing of the decrease of "patience". Figure 6 shows a possible development of a customer's patience after the discrete event occurred.

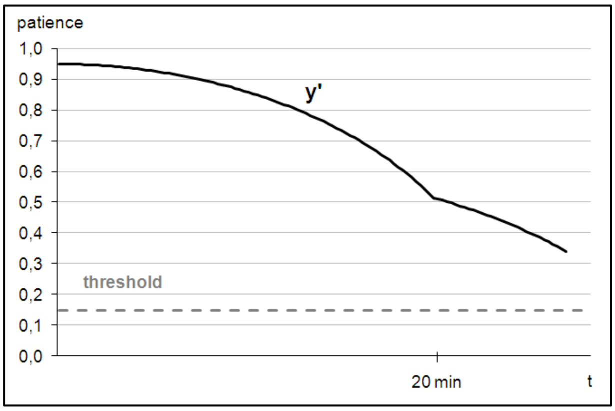

Figure 6: A Discrete Event Causes a Discrete Change in the Differential Equation

Third, it is possible that the occurrence of the discrete event causes the customers' discrete attribute "loyaltyToServer" to change. As the attribute "threshold" is depending on the loyalty it has also be updated immediately.

For our experiment we let the arrival times be exponentially distributed with rate paramter λ = 0.1 [min] and the service times normally distributed with μ = 8 [min] and σ = 5 [min]. In 500 replications around 30 percent of the customers left the queue before being served. The minimum waiting times were at 20 minutes ± 10 seconds. The maximum waiting time a customer spent in the queue was 58 minutes ± 30 seconds.

4.3. Evaluation of the Experiment

A small and simple model like this is well suited for explaining the ideas behind hybrid tokens. But at the same time the question may arise why modeling this system with the aid of hybrid and, with respect to the computational effort, "expensive" tokens. As the system only contains one continuous variable for the patience of the customer it would be easy to precompute the point in time when the customer will leave the queue early before getting served. This point in time could be scheduled in the event list and the modeling of the continuous change of the patience would not be necessary.

But when a larger number of continuous attributes are involved and the equations describing the rates of changes are not particularly simple, the preliminary computation of the values can become very complex. In that case the use of hybrid tokens might be the easier solution. Furthermore, the modeling of all necessary attributes and their dynamics directly within the token is much closer to the natural way of observing and understanding such kinds of systems.

Examples can be found in many different application fields. Considering industrial psychology the human attributes satisfaction and performance are of particular interest (Borman, Ilgen, Klimoski, and Weiner 2003). These are influenced both by several system events and by several other continuous human attributes such as mood, fatigue, or alertness. They are all mathematically related in different ways and have influences on each other as well as on the system output, for example by increasing or decreasing the performance of the human being. One example of use might be the evaluation of different strategies for increasing a worker's performance during days and night shifts.

Hybrid tokens enable the simulation of all relevant discrete and continuous attributes and their dynamics without making non-trivial pre-computations necessary. One further advantage is that the graphical representation of the Petri net is kept simple as no auxiliary transitions have to be added for being able to model all attribute dynamics.

5. SUMMARY AND CONCLUSION

Systems in practice often cannot be easily classified either as a discrete or a continuous system. But usually the system's behavior is more determined by one of the types than the other having an effect on the choice of modeling approach. If the system's design and the objectives of the simulation study require the examination of both discrete and continuous aspects hybrid simulation techniques have to be considered. But usually, these kinds of techniques do not consider distinguishable entities moving around within the system boundaries.

As we are currently developing a simulation model of the German mental health care system, we are interested in a modeling paradigm allowing the specification of entities in detail, including both discrete and continuous attributes, that are part of a discrete-event-driven system. In mental health care systems, the most important entity is the patient taking part in psychiatric services. An aggregation to a "fluid" mass may be insufficient in matters of patient-related questions, for example problems concerning mood, motivation, or the sensed quality of care. Several human attributes, especially mental ones, have to be considered making the simulation study very complex. Nevertheless, the resulting model will be used by decision makers and planners of psychiatric services who are usually not familiar with simulation techniques.

For these reasons, we introduced a new modeling approach for discrete-event-driven systems with entities that can have both discrete and continuous attributes. The approach is based on the concept of Petri nets and we refer to the entities as hybrid tokens.

This different approach to the concept of hybrid simulation may be useful for those kinds of systems where a detailed examination of involved entities with all relevant attributes is of particular interest and cannot be disregarded. The main advantage is an emphasis placed on the entities as well as their reactions and influences on the system. That might be important for all applications with human beings in dynamic environments, especially if the users of the simulation model are not simulation experts. On the one hand human characteristics can be modeled in a detailed manner but at the same time this information is encapsulated and can be hidden from the simulation user. With the aid of hybrid tokens the system and its entities can be modeled precisely and still be illustrated by a graphical representation that is easy to understand. Additionally, adding changing entity attributes directly to the entity equates more the natural way of thinking than modeling the dynamics separately from the entity.

For the above reasons, we believe that Petri nets with hybrid tokens are able to contribute to an intuitive way of modeling discrete-event systems with a close approximation of the entities involved.

REFERENCES

Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M., 2001. *Discrete-Event System Simulation*. Third Edition, Prentice-Hall, Inc.

Wank, J., 1998. *Timed Petri Nets: Theory and Application*. Kluwer Academic Publishers.

Jensen, K., 1997. *Coloured Petri Nets: Basic Concepts, Analysis Methods, and Practical Use*. Volume 1, Second Edition, Springer.

Law, A.M., Kelton, W.D., 2000. *Simulation Modeling and Analysis*. Third Edition, McGraw-Hill Higher Education.

Horton, G., Kulkarni, V.G., Nicol, D.M., Trivedi, K.S., 1998. Fluid stochastic Petri nets: Theory, applications, and solution techniques. *European Journal of Operational Research,* Volume 105 (1): Pages 184-201.

Macal, C.M., North, M.J., 2005. Tutorial on agent-based modeling and simulation. *Proceedings of the 37th Conference on Winter Simulation,* Pages 2-15. December 4-7, 2005, Orlando, FL, USA.

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., Zhao, L., 2002: *Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective.* Technical report. The Wharton School, University of Pennsylvania, Philadelphia.

Pritsker, A.A.B., 1995. *Introduction to Simulation and SLAM II.* Fourth Edition, John Wiley.

Borman, W.C., Ilgen, D.R., Klimoski, R.J., Weiner, I.B., 2003. *Handbook of Psychology, Volume 12: Industrial and Organizational Psychology*. Wiley & Sons, Inc.

AUTHORS BIOGRAPHY

Kristina Dammasch studied Computational Visualistics at the Otto-von-Guericke University of Magdeburg. She was awarded her German University diploma in June 2006. Since November 2006 she is working at the Research Group "Simulation and Modeling" at the University of Magdeburg. Her email address is: kristina@sim-md.de.

Graham Horton studied Computer Science at the University of Erlangen, obtaining his Masters degree ("Diplom") in 1989. He obtained his PhD in Computer Science in 1991 and his "Habilitation" in 1998 at the same university, in the field of simulation. Since 2001, he is Professor for Simulation and Modeling at the Computer Science department of the University of Magdeburg. His email address is: graham@sim-md.de.