Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

MEDMODEL - SPECIALIZED SOFTWARE FOR THE HEALTHCARE INDUSTRY

Lucien F. Keller, Jr.

PROMODEL Corporation
Healthcare Systems Division
1100 N.W. Loop 410, Ste 700
San Antonio, Texas 78213

ABSTRACT

MedModel represents a substantial forward movement in the evolution of simulation software. Designed to be simple to use and tailored to meet the analytical demands of the healthcare environment, it provides a basis for the comprehensive evaluation of large, complex problems representative of healthcare systems in general. This paper serves as a cursory examination of some of the product's characteristics and capabilities designed to solve hospital-specific simulation problems.

1 BACKGROUND

One of the most difficult problems facing a hospital systems analyst - regardless of facility size or type - is the evaluation and analysis of just exactly what occurs in the healthcare process. This means, where delays and bottlenecks occur, what's efficient and what's not and what the overall effect may be of adopting different patient management policies. Generally, it's because there are just too many interrelated and highly varied steps involved in any given healthcare process scenario to watch them all at once. Accordingly, many hospitals continue to rely on massive computer driven databases to "post-operatively" examine the impact of workload, staffing, patient mix, capacity and policy on cost, quality and financial viability. Unfortunately, the correlations between these variables are rarely exact and the ability to "watch" or measure the activity of a healthcare process in general, is constrained when a database alone forms the foundation for analysis.

In fact, it has been difficult to find an analytical tool that would really handle such a variety of issues well. That is, a tool that would allow an investigator to efficiently and quickly model any given healthcare process from the admission to the disposition of a patient; or one that would be every bit as effective for the modeling of a single activity (ward, emergency room, pharmacy, admissions, etc.) as it would for an entire network of activities (hospital clinics, surgical services, an entire hospital, etc.).

Moreover, the problem of finding such a tool actually extends beyond simulation itself since simulation long ago proved its value to the manufacturing sector and has been used to evaluate process problems in healthcare as well. However, pure manufacturing is anything but an accurate reflection of what occurs in a healthcare setting. And, as such, simulators and simulation languages designed with the manufacturing environment in mind rarely contain the constructs and algorithms necessary to handle unique healthcare issues.

2 THE DEVELOPMENTAL FOCUS

Developmental activity was actually focused on two separate planes. The first concentrated on understanding the specific simulation needs of healthcare and the hospital environment. The second on the availability of current and future micro-computer and software technology to improve the operational characteristics of the simulation itself.

2.1 Healthcare/Hospital Systems Needs

MedModel's creation was prompted by numerous requests for algorithms and constructs needed to handle the very unique problems associated with healthcare in general and hospitals specifically. Of singular concern were requests for mechanisms to facilitate model building under circumstances involving extremely complex pathing, transparent relationships, multiple and repetitive activities, and great operational variety - all of which are generally found in the most routine of healthcare systems models.

More importantly however, developmental concern went beyond simply capturing the essence of a healthcare system. Rather, it focused on ensuring that specific constructs represented precisely what was required to model the activity of interest and did so with as much statistical validity as possible. For example, an

initial concern surrounded the fact that patients and their records (doctors notes, laboratory test results, etc.), although often managed through different healthcare channels, were not really separate entities. In fact, each was inextricably linked to the other by "ownership." When one was delayed by a bottleneck in the system, the other generally followed suit. Accordingly, specifically matching one with the other at various times in a simulation was often critical to the performance of a model.

Because these issues are often unique, how MedModel handles them is of specific interest.

2.1.1 Matching entities

As mentioned above, matching patients with their supporting medical files, records and test results holds specific significance since the progress of a patient's records through the complex system of medical reviews and handling may determine the patient's progress from one provider to another. In the case of the Radiology Clinic depicted below, each patient's films are processed through automated developers following exposure while the patient awaits the results before proceeding to another clinic. Although a small example of the need for matching, it has been shown that arbitrarily joining patients with films as they queue for exit from processing produces patient waiting times that are significantly different statistically from those experienced otherwise. As a result, simulation outputs tend to be invalid for the system under study.

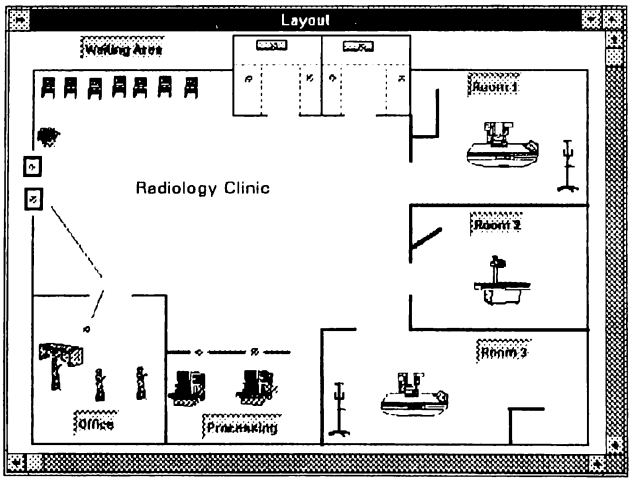

Figure 1: Radiology Clinic with Film Processing

MedModel handles the requirement to ensure that entities are matched by identity (or ownership) by simply relying on a "matching" construct resident within the software. In this case, the specific film(s) associated with each patient are directly matched to that patient based both on the patient's and the film's identity. This patient identity is established early in the model and is automatically transferred as an attribute to any entity that may emanate from the patient.

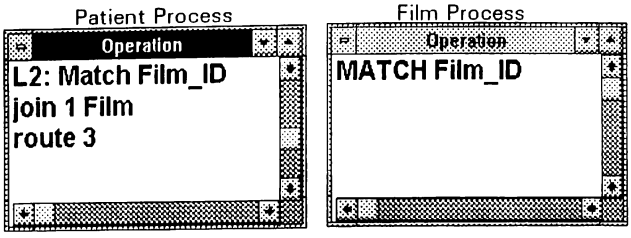

Figure 2: Program Syntax for Film Matching

2.1.2 Reserving Locations

Another process difficult to emulate but essential to the healthcare modeler is that associated with reserving a location. In this case, capturing the notion that an examination room is occupied when in fact the patient has been sent to another area for treatment but will be returning to that specific location.

In the case of the 24-hour HMO Clinic below, patients are first escorted to an examination room, then to a treatment room and finally back to the examination room they originally occupied. As with the actual clinic, no other patient is permitted to enter the reserved room until the patient returns from treatment and finally leaves.

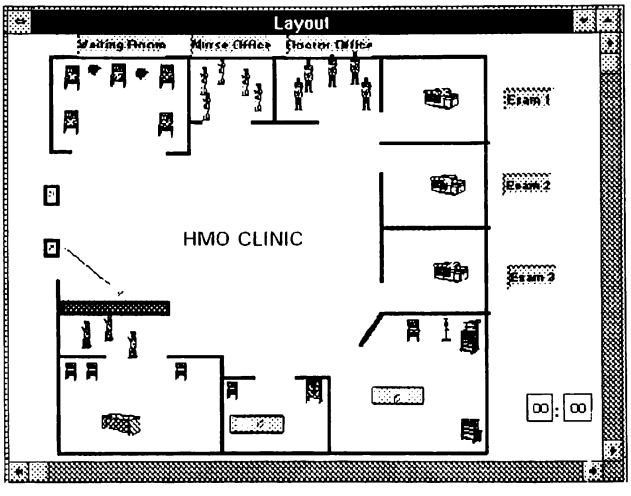

Figure 3: HMO Clinic

To handle this situation, MedModel allows the modeler to use statements that are not only familiar to most people with only a slight acquaintance with programming but generally in plain English. As shown

in figure 4 below, the patient is directed to the treatment room through a "route" (i.e. ROUTE 1) statement associated with that location. However, no statement "ending" the process sequence is issued indicating that the software is to continue reading and executing the remaining statements even though the patient has departed.

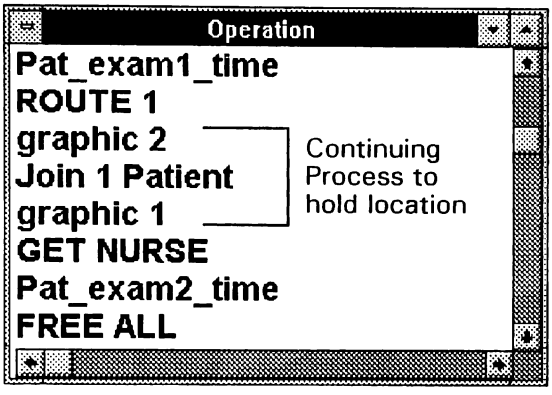

Figure 4: Operation Statement to Reserve a Location

The graphic statement linked with the absence of an ending statement and followed by a join statement automatically reserves the location until the patient that was using it returns.

2.1.3. Independent, Normally-distributed Arrivals

The third example of specialty constructs involves patient arrival patterns. Like other simulation software packages keyed to industrial applications, MedModel is equipped to automatically handle the full range of stochastic arrival patterns associated with most activity functions and entity arrivals.

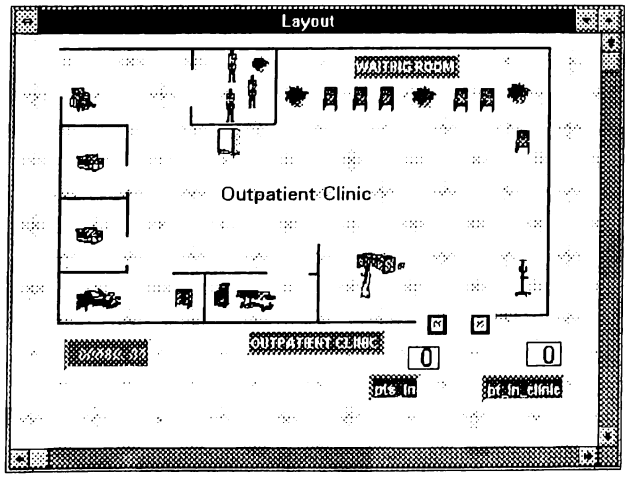

Figure 5: Outpatient Clinic

However, unlike industrial packages, it is also capable of handling hospital patient arrivals that are directly created by an appointment system. In this instance, it is routinely accepted that patients arrive somewhat normally distributed around their appointment times depending on the type of clinic they are scheduled to visit. In the Outpatient Clinic model shown above, patients are scheduled to arrive normally distributed around each 30 minute period. Note that this is not interpreted as a 30 minute inter-arrival rate but rather as 30 minute events that are completely unrelated.

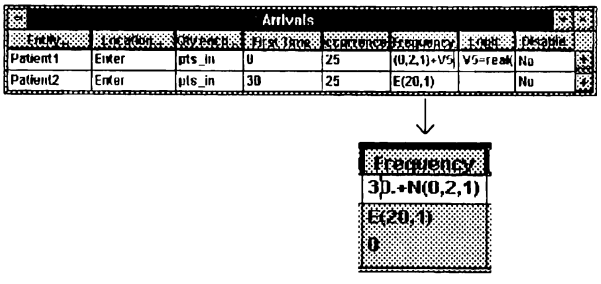

Figure 6: MedModel Appointment Arrivals

MedModel handles this requirement as shown in figure 6 by allowing the modeler to enter an expression as part of the arrival frequency. In this example, 30+(0,2,1)) indicates that patients are to arrive normally distributed within +/- 2 minutes of every 30 minute period.

2.1.4 Patient Acuity

Certainly not the least of the problems handled deftly by MedModel is that concerning the management of patients first on a first-come-first-served basis and then according to acuity.

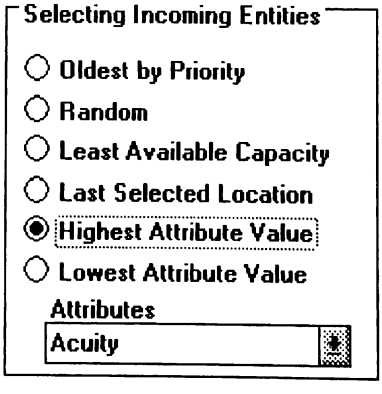

Figure 6: Location Decision Rules

In this case, MedModel defers to its object-oriented capability to assign operating characteristics to not only every location but to every resource used within the model. In one of the decision-rule menus shown in figure 6 above, a location is assigned the requirement to select the next patient from among a group of patients based on which patient has the highest acuity. If all patients have the same acuity, MedModel automatically defaults to that patient that has waited the longest.

2.1.5. In General

Of course, what has been addressed above is only a small part of what MedModel is capable of doing. Because the modeling of healthcare processes is almost always characterized by the need for complex process logic due to the seemingly endless variety of patients and activities, MedModel is equipped with an impressive collection of pre-programmed constructs that handle the vast majority of related problems. In other words, constructs that handle issues like the simultaneous but conditional use of different members of a healthcare team or the requirement to preempt certain medical activities when higher priorities come along, are built in. That makes it easier for the modeler to concentrate on modeling rather than on developing complex expressions to represent common activities.

2.2 Technology

To facilitate ease of use while simultaneously capturing extremely large-scale modeling capability, MedModel is based on object oriented programming using C++ and run entirely under Microsoft's WINDOWS. This means that a model's size is limited only by the amount of memory available and that MedModel takes full advantage of synchronized windowing capability, as well as dynamic data exchange offered by WINDOWS. Of course, this also means that larger and faster machines produce ever more dramatic results when building and running models. In fact, although any standard 486DX machine with VGA graphics and 8 Megabytes of RAM should suffice, a similarly equipped Pentium machine with Super VGA will speed-up work significantly.

Of course there is more to "technology" than the operating system itself. Employment of software features available through existing programs that emphasize the capability to make full use of system capacity is also essential. Accordingly, MedModel avails itself of just such capability by being nested in Microsoft's Windows.

What makes MedModel unique then are the features that it calls on to make healthcare system modeling in general and hospital modeling specifically far easier. Coupled with responsive and versatile animation, MedModel provides a basis not only for rapid model building but ease of validation as well.

2.2.1 "Point and Click" Approach

Because MedModel was written under WINDOWS, model definition depends, to a large extent, on nothing more than using a mouse to identify, select or place appropriate components of the model. In addition, since it is fully menu-driven, the modeler need only "click" on an appropriate field to select an element for entry of information. In essence, a model can be built in its entirety simply by placing representative icons on the screen and then selecting different descriptive fields to define movement, relationships and activities between and among model entities and locations.

2.2.2. Custom Icons

MedModel also comes with an impressive library of colorful, pre-designed healthcare system icons representing everything from hospital-specific patients, staff members, material and treatment fixtures to instruments, used surgical trays and the like. More significant however, is the fact that MedModel is accompanied by a new, state-of-the-art icon editor that enables the modeler to design any manner of icon desired using an almost limitless array of colors and shapes. The value of this capability transcends simple esthetics and allows the modeler to represent, as accurately as possible, specifically what is occurring in the hospital environment. This means the capability to use graphical displays as a means to validate the model as well as the process it represents.

2.2.3. Automatic Processing & Path Entries

Unlike the vast majority of simulation software packages, MedModel presents the modeler with the capability to design and construct a model using nothing more than the mouse. This is especially useful when identifying movement and processing steps. In this case, the modeler need only click on the succession of locations to which a patient may move and the required movement entries are made automatically. This holds true for both resource (i.e. doctor, nurse, technician, etc.) movement as well as processing. To create the required pathing for a clinical patient for example, the modeler need only draw a line from the clinic entry to the different locations the patient might go. MedModel automatically enters all distances required to complete pathing.

3. CONCLUSIONS

MedModel is more than a simple step forward in the world of healthcare simulation. What it actually represents is a new opportunity for all elements of the healthcare sector to get involved with analytical tools that are far simpler, easier to use and yet more powerful than any previously available. Of greater significance however, is the fact that because of the comprehensive nature of the tool, assumptions and short-cuts that have routinely characterized healthcare and hospital simulations, are no longer necessary. Rather, it is now possible to model complex healthcare systems accurately and with confidence in the results.

AUTHOR BIOGRAPHY

LUCIEN F. KELLER, JR. is Vice President and Director of Healthcare Systems Engineering for PROMODEL Corporation. He has a B.S. degree in Psychology, a Masters degree in Logistics and extensive Ph.D. level work in Operations Management. He gained his healthcare experience through twenty four years as a Medical Service Corps officer in the U.S. Army.