9-2000

Using Expert Systems for Simulation Modeling of Patient Scheduling

Charles R. Standridge
Grand Valley State University, standric@gvsu.edu

Duane Steward
Massachusetts Institute of Technology

TECHNICAL ARTICLE

ISSN 0037-5497/00
Printed in the United States of America

Using Expert Systems for Simulation Modeling of Patient Scheduling

Charles R. Standridge
Padnos School of Engineering
Grand Valley State University
Eberhard Center, 301 West Fulton
Grand Rapids, Michigan, 49504-6495
E-mail: Standric@gvsu.edu

Duane Steward
Clinical Decision Making Group
Massachusetts Institute of Technology
NE43-415, 545 Technology Square
Cambridge, Massachusetts, 02139
E-mail: duane@medg.lcs.mit.edu

Modeling the scheduling of patient appointments is an important issue in simulating a health care delivery facility. A simulation model must include the control logic of appointment scheduling software and the explicit and implicit decision rules used by the human scheduler in selecting an appointment time. Expert systems provide one way of modeling such control logic and decision rules. We describe a structure for an expert system that models patient appointment scheduling and the integration of such an expert system within a simulation model. An example expert system for a small animal veterinary clinic is presented.

Keywords: Expert systems, scheduling, health care delivery

1. Introduction

In a health care delivery system, patients often are scheduled to arrive according to their appointment times. The time between appointments may be sufficient to characterize the arrival process if the care delivered, as well as the delivery system resources employed, can be modeled as homogeneous among the patients served.

Such homogeneity is not always the case. Patient visits may include initial examinations and annual wellness checks, as well as follow-up examinations and procedures for previously seen problems. Different types of examinations or procedures may require different work areas, equipment, or health care providers. Time requirements may vary significantly as well. Follow-up examinations and procedures need to be scheduled within a specified time interval after preceding examinations and procedures. Each type of examination or procedure may be clustered in a particular time period.

A health care delivery system meets these requirements for determining when patients arrive by using a scheduling system. This system may be completely manual or semi-automated. Computer software could keep track of what times health care providers and physical system resources are available and when within these times appointments are currently scheduled. The operator of the scheduling system, typically a receptionist, is responsible for selecting an appointment time from among those available that meets the requirements of the examination or procedure that will be performed. These requirements include assuring that needed equipment and the appropriate health care provider are not scheduled for conflicting tasks.

A simulation model of such a health care delivery system must incorporate the scheduling system. Thus, the model must include the control logic of the scheduling protocol and the explicit and implicit decision rules used by the receptionist in selecting an appointment time as well as keeping track of current appointments and available times.

Expert systems provide one way of modeling control logic and decision rules such as those employed in an appointment scheduling system. Existing simulation languages support the modeling of examinations and procedures as well as the tasks performed by health care delivery personnel and the use of equipment. Integrating an expert system into a model developed using an existing, commercial language supports the inclusion of an appointment scheduling system within a simulation.

Patient scheduling in a companion animal veterinary medicine practice is of particular interest. A great variety of examinations and procedures are performed in such a practice. Physical resources, such as surgery space, are few. Staff-to-patient ratios are low. Often a single veterinarian is required to perform at least a significant part of each procedure. Such a resource-constrained situation enhances the importance of patient scheduling for the efficient and economical operation of the practice.

We discuss a technique for modeling a patient scheduling system using an expert system and how such an expert system can be integrated into a simulation model. We present a general structure showing how an expert system can model a patient scheduling system.

The patient scheduling problem meets two criteria for using an expert system:

1. An inherent IF-THEN knowledge structure, and
2. A loosely structured constraint context.

Solutions to the patient scheduling problem are not unique. The expert system is used to select and recommend one solution from among those that are possible. If the constraint context is more highly structured, or unique solutions can be generated from a decision tree, a static procedure based on explicit IF-THEN statements may be employed with smaller overhead than an expert system.

We show the application of this structure within a generic simulator. The simulator was constructed to assist in the design and operation of small-scale companion animal veterinary medicine practices. The use of a small expert system within this simulator demonstrates how such a mechanism can be used for patient scheduling within a simulation model.

2. Background

Little work has been reported concerning the application of simulation to veterinary practice design. Steward and Standridge [1, 2] discuss and illustrate potential benefits for the design and operation of the practice of a single veterinarian. Benefits identified include estimating the amount of physical space, pieces of equipment and number of staff needed to meet the demand for patient care. Alternative uses of space and scheduling policies can be evaluated. The economic feasibility of practice expansion can be assessed. Illustrative practice design cases included simulation experiments conducted using a generic simulation model, or simulator. The effect on the number of patients seen, the number of late patient discharges, and the average minutes late for late discharges are measured for differing numbers of technicians, numbers of phone lines, and intervals between the last appointment and closing time. Using simulation in the design of a new veterinary practice is demonstrated. Technical aspects of the simulator are presented by Steward and Standridge [3]. However, only a brief overview of the modeling of patient arrivals is given. The reader who is primarily interested in modeling and experimentation for veterinary practice design is referred to the papers discussed in this paragraph.

There is significant work concerning the concurrent use of artificial intelligence, including expert systems, and simulation. Some work seeks to provide a unifying framework between the two fields. Fishwick [4] seeks to find a common terminology and taxonomy between artificial intelligence, software engineering, and simulation to facilitate the inter-working of these specialties. O'Keefe [5] presents a taxonomy for combining simulation and expert systems. One component of the taxonomy provides for the parallel and interactive operation of expert systems and simulation, the approach followed in this paper.

Some work has focused on modeling the dynamic behavior of human beings and other system resources using artificial intelligence techniques. Two papers are typical. Nadoli and Biegel [6] present a knowledge representation scheme to achieve modular modeling. Blackboards are used to model how intelligent agents make complex decisions in the operation of a manufacturing system. The manufacturing system is represented by classes of queueing networks. Burns and Morgeson [7] present a simulation modeling procedure for representing intelligent decision making entities, called actors, who respond to changes in the state of the system. The decisions that each actor must make are defined, along with the event that triggers the decision making. The actor must choose from among a set of actions. The action set may be modified in the course of the simulation. This may take the form of a choice of one action from a database of options guided by a loosely structured list of rules. More complex scenarios may arise that include the retraction of earlier assertions and dependent inferences. This is referred to as non-monotonic reasoning in the artificial intelligence literature. In a similar vein, Robinson, Edwards, and Yongfa [8] describe the use of an expert system to model human decision making. Two possible approaches are identified: (1) elicit the decision rules from the expert and represent them within an expert system, and (2) use the simulation model to prompt the expert to make decisions, building up a set of examples from which an expert system could learn. The expert system is subsequently linked with a simulation model as the means to include human decision making.

Expert system techniques have been developed to assist the modeler in designing simulation experiments as discussed by Taylor and Huron [9] as well as Tao and Nelson [10]. Others have applied expert systems to the statistical analysis problems associated with simulation experiments: Mellichamp and Park [11] and Ramachandran, Kimbler, and Naadimuthu [12].

No work discussing the use of these techniques for modeling the scheduling components of systems has been located.

3. Expert System Structure for Patient Scheduling

Modeling receptionist decision making for patient appointment scheduling was of primary importance in designing and building the veterinary practice simulator. A general expert system structure for representing such human decision making was developed.

We believe by building a general structure initially, as well as by constructing one typical application using the structure, that subsequent applications will require less time. The structure and previous applications provide guidance. Small animal veterinary practices are similar to each other. Thus, rules developed in modeling one practice will likely serve as the starting point for rules needed to model other practices. In addition, the same rules may be used in multiple models.

A general structure for an expert system for modeling a patient appointment scheduling system is shown in Figure 1. The expert system consists of a working knowledge base of currently scheduled appointments (appointment calendar) and a rule base specifying how appointments are scheduled, as well as an inference engine. The latter uses the rule base, the appointment calendar facts, and information about an appointment to be scheduled to determine one or more appointment times consistent with all requirements.

The rule base has four types of rules expressed in IF-THEN statement form: appointment-type-compatibility, appointment-time-determination, temporal-constraint and current-time. Appointment-type-compatibility rules help manage the assignment of health care delivery system resources to patient care tasks. These rules seek to maximize resource utilization while avoiding conflicting assignments to concurrent tasks. For example, an outpatient examination cannot be scheduled to begin concurrently with an outpatient surgery since the veterinarian requires a significant amount of time to perform both. However, an outpatient examination can be scheduled to begin concurrently with an outpatient non-surgical treatment if significant parts of that task can be performed by a technician.

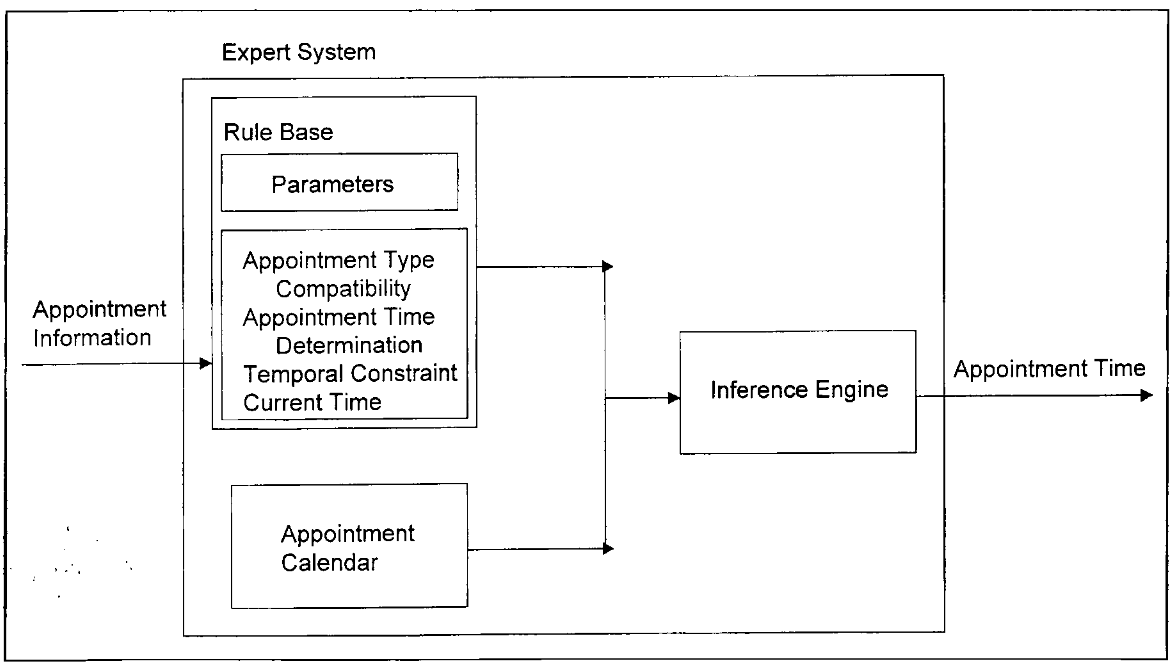

Figure 1. General structure of an expert system for appointment scheduling

Appointment-time-determination rules specify how to search the appointment calendar to determine an appointment time. The search strategy tries to minimize the difference between the desired appointment time specified by the health care provider and the actual appointment time. An appointment time is feasible if no incompatible appointments exist during the interval: appointment time + duration of appointment. The interval should not contain the closing time of the facility, that is, the appointment can be completed before closing.

Appointment-type-compatibility rules and appointment-time-determination rules model the explicit decision making procedure by which an appointment time is selected. The individual performing the appointment scheduling function also employs implicit rules that constrain when appointments can occur. Such rules must be made explicit in the expert system.

The health care facility may be closed during certain time intervals each day, such as 6:00 p.m. to 7:00 a.m. the following day, and on specified days of the week such as Sunday. These times may be initialized in the appointment calendar as appointments of a null type. Closed-facility rules define null appointments as incompatible with all other appointment types. Current-time rules forbid the scheduling of an appointment prior to the current simulation time.

Contextual parameters are stored as facts within the knowledge base. These are used to provide operational quantities that can be changed for simulation experimentation without modifying the IF-THEN statements. Typical parameters include the minutes before closing after which appointment times are not allowed, as well as the daily opening and closing times.

Appointment information characterizes each individual appointment. The inference engine uses this information to enforce the appointment-type-compatibility rules and to select feasible appointment times using the appointment-time-determination rules. Typical appointment information includes the target starting time, duration, and type of appointment, as well as the current simulation time and a unique patient identifier.

4. The Veterinary Practice Simulator Scheduling Expert System

Employing an expert system for modeling patient scheduling using the structure shown in Figure 1 requires the following:

1. Specifying the appointment information,
2. Specifying the rules,
3. Specifying contextual parameter facts,
4. Implementing the appointment calendar and the inference engine.

In the veterinary practice simulator, these steps are accomplished as follows. A companion animal veterinary practice serves a variety of patients that require a wide variety of services. These services include annual examinations, follow-up examinations for previously provided treatment, outpatient treatments, inpatient treatments one to three times daily, and inpatient and outpatient surgery. Resources required to deliver this health care include personnel such as veterinarians, veterinary technicians, and receptionists. Physical space resources are required: examination rooms, surgical areas, surgical preparation, recovery areas, dental treatment areas, cages, and phone lines.

Appointment information uniquely characterizes each appointment and is as follows:

Target starting time:	The most desirable time for the appointment to begin
Duration:	The time duration of the appointment
Type:	What kind of examination or procedure will be performed during the appointment
Patient ID:	A unique identifier of the patient
Current-time:	The current simulation time

To avoid a voluminous and unnecessarily detailed presentation, only the rules most significant in modeling the behavior of the individual performing the scheduling task are presented. Other rules take care of expert system computer implementation details.

Appointment-type-compatibility rules define the appointment types that may and may not be scheduled concurrently. In the veterinary practice simulator, the following appointment types were supported: examination (Exam), outpatient or inpatient non-surgical treatment (Treatment), and outpatient or inpatient surgery (Surgery). Compatibility rules are processed only when the feasibility of scheduling an appointment concurrently with an existing appointment is assessed.

The compatibility rules used in the veterinary practice simulator are shown in Figure 2. DesiredType is the type of appointment to be scheduled, and CurrentType is the type of appointment already scheduled. The rules return NO if the appointment types cannot be scheduled concurrently and YES if they can.

In summary, the appointment-type-compatibility rules forbid an appointment to start when another appointment of the same type is ongoing. An ongoing surgery or examination excludes starting another appointment. Resource requirements for a treatment are flexible enough to permit starting a surgery or an exam while a treatment is ongoing.

Closed-facility rules augment the appointment-type-compatibility rules. These rules indicate the times that the veterinary practice is closed and unavailable for appointments. In addition, appointments are not allowed in a specified time interval before the practice closes,

```
// Appointments of the same type cannot be scheduled concurrently
if (DesiredType == CurrentType) return NO

// An examination cannot be scheduled to begin concurrently with an ongoing surgery
if (DesiredType == Exam && CurrentType == Surgery) return NO

// An examination can be scheduled to begin concurrently with an ongoing treatment
if (DesiredType == Exam && CurrentType == Treatment) return YES

// A treatment cannot be scheduled to begin concurrently with an ongoing examination or surgery
if (DesiredType == Treatment && ((CurrentType == Exam) || (CurrentType == Surgery)))
  return NO

// A surgery cannot be scheduled to begin concurrently with an ongoing examination
if (DesiredType == Surgery && CurrentType == Exam) return NO

// A surgery can be scheduled to begin concurrently with an ongoing treatment
if (DesiredType == Surgery && CurrentType == Treatment) return YES
```

Figure 2. Compatibility rules used in the veterinary practice simulator

```
// No appointment may be schedule when the clinic is closed or during the pre-closing interval
if (CurrentType == Closed || CurrentType == PreClosed) return NO
```

Figure 3. Closed-facility rules used in the veterinary practice simulator

PreClosed. The closed-facility rules are shown in Figure 3. An appointment of type Closed indicates that the clinic is closed.

Appointment-time-determination rules search the appointment calendar for an appropriate time to schedule an appointment. An appointment time is available if there is no incompatible appointment in the range defined by the starting appointment time and the duration of the appointment. No incompatible appointment means that there is no other appointment scheduled or that the compatibility rules are satisfied (return YES).

The best time for the appointment is specified by Target = CurrentTime plus some Interval. If this time is available, the appointment time is Target. If this time is not available, the appointment calendar is searched both forward and backward in time for the closest available appointment time to Target. The closer of the two appointment times, one found by searching forward and one found by searching backward, is chosen as the appointment time.

The appointment-time-determination rules are shown in Figure 4. This rule set must be reviewed iteratively until an available appointment time is returned, as shown in Figure 5. The iteration ends when the Target appointment time is the appointment time returned by the rules. The iteration is necessary since the computation of a proposed appointment time, given that the Target time is not available, takes into account only the previously scheduled appointments closest to the target. It is possible that other appointments may exist in the time interval, proposed appointment time + Duration, that make the proposed appointment time unavailable.

Current-time rules prevent scheduling appointments before the current time, CurrentTime as shown in the last two rules in Figure 4.

The expert system was implemented in the C programming language. Rules were transliterated into IF-THEN-ELSE statements and associated logic. This C program served as the inference engine. The appointment calendar was stored as a random access binary file. Relevant pieces of the appointment file were read into memory to search for appointment times.

5. Expert System – Simulation Model Integration

The scheduling expert system operates concurrently and in parallel with the simulation model as shown in Figure 6. The simulation model invokes the expert system whenever an appointment must be scheduled, either as a follow-up to a previous appointment or as

```
// Duration = Expected time needed for the appointment
// TimeBetweenAppointment = minimum interval between the start time of two successive appointments = 15
minutes
// BestTimeForward == Earliest time after Target that appointment can begin
// BestTimeBackward == Latest time before Target that appointment can begin

//Appointment time slot availability rule
if (there is an incompatible appointment in the interval Target+Duration) return(NotAvailable)
else return(Available)

// ClosestAppointmentForwardStartTime == Starting time of appointment that begins the soonest
// at or after Target
ClosestAppointmentForwardStartTime = Minimum time >= Target when an appointment starts

// ClosestAppointmentForwardEndTime == Ending time of appointment that begins the soonest
// at or after Target
ClosestAppointmentForwardEndTime = Ending time of the appointment whose start time is the Minimum time >=
Target when an appointment starts

// ClosestAppointmentBackwardStartTime == Starting time of appointment that begins the latest
// before Target
ClosestAppointmentBackwardStartTime = Maximum time < Target when an appointment starts

// ClosestAppointmentBackwardEndTime == Ending time of appointment that begins the latest
// before Target
ClosestAppointmentBackwardEndTime = Ending time of the appointment whose start time is the Maximum time
< Target when an appointment starts

// Use Target as the appointment time if possible
if (Target == Available) return {Target}

// Find closest available appointment time following target
if (Target != Available)  BestTimeForward = ClosestAppointmentForwardEndTime

// Find closest available appointment time preceding target
if (Target != Available &&
        ClosestAppointmentForwardStartTime – Duration >= ClosestAppointmentBackwardEndTime)
        BestTimeBackward = ClosestAppointmentForwardStartTime – Duration

if (Target != Available &&
        ClosestAppointmentForwardStartTime – Duration < ClosestAppointmentBackwardEndTime)
        BestTimeBackward = ClosestAppointmentBackwardEndTime

// Make sure BestTimeBackward is not the same as the Target
if (BestTimeBackward == Target) BestTimeBackward = Target – TimeBetweenAppointments

// New Target is the closer of BestTimeBackward and BestTimeForward to the current Target
if (Target – BestTimeBackward <= BestTimeForward – Target) return BestTimeBackward

if (Target – BestTimeBackward > BestTimeForward – Target) return BestTimeForward
```

Figure 4. Appointment-time-determination rules

```
while (Target != Appointment Time Returned by Appointment-time-determination Rules)
        {
        Target = Appointment Time Returned by Appointment-time-determination Rules
        }
```

Figure 5. Iteration logic for firing appointment-time-determination rules

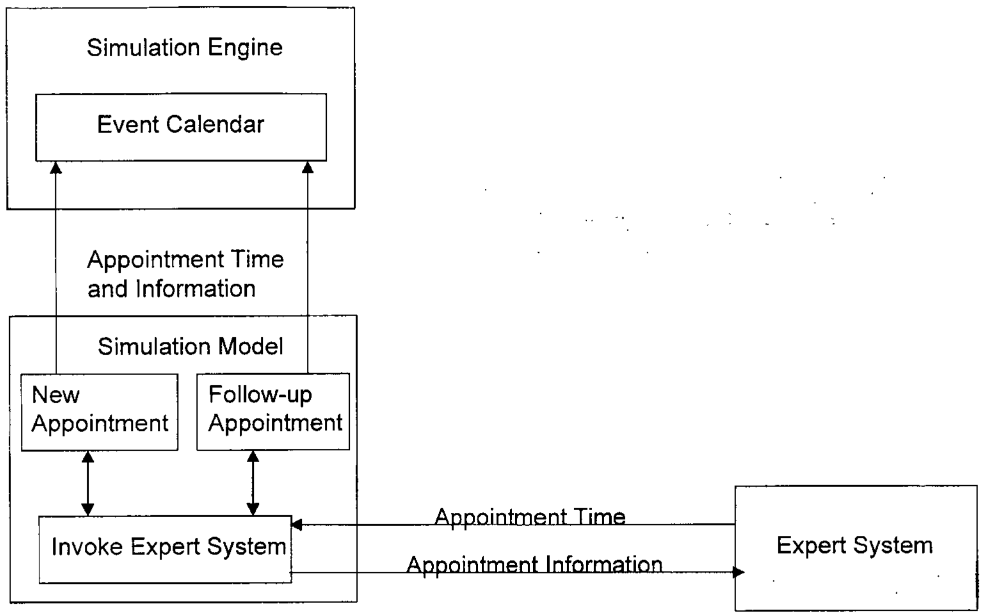

Figure 6. Expert system—simulation model integration

a new appointment. The appointment information is passed to the expert system and an appointment time is returned. The arrival of the patient is scheduled on the simulation event calendar at the appointment time.

Note how the simulation model and the expert system operate in parallel. The expert system has no knowledge of the simulation. It simply uses the appointment information passed from the simulation to determine an appointment time using the rule base and the appointment calendar. The simulation model has no knowledge of the rule base or the appointment calendar. It simply uses the appointment time returned by the expert system.

In the veterinary practice simulator, the interface between the simulation model and the expert system was constructed as follows. The simulation model was implemented in SLAMSYSTEM. Whenever scheduling an appointment was necessary, an event was invoked at the current simulation time. The event routine, coded in FORTRAN, passed the appointment information to the expert system, coded in C. An appointment time was returned. The event routine placed the appointment arrival event on the event calendar at the appointment time.

Time of Day	Appointment Type
Before 9:00 a.m.	Closed
9:00 a.m.	Surgery
10:30 a.m.	Surgery
12:00 p.m.	Closed
1:00 p.m.	Treatment
1:15 p.m.	Examination
1:30 p.m.	
1:45 p.m.	
2:00 p.m.	
2:15 p.m.	Treatment
2:30 p.m.	Examination
2:45 p.m.	Examination continued
3:00 p.m.	Treatment
3:15 p.m.	Examination
3:30 p.m.	
3:45 p.m.	
4:00 p.m.	Pre-closed
5:00 p.m. and after	Closed

Figure 7. Typical veterinary clinic appointment calendar

6. Example Schedule Dynamics

Validation evidence concerning the expert system is obtained if the expert system produces a feasible appointment schedule for a small animal veterinary practice. The production of such a schedule is illustrated by the following.

Figure 7 shows the appointment calendar for a typical day in a small animal veterinary clinic. The clinic accepts appointments from 9:00 a.m. to 12:00 p.m. and 1:00 p.m to 4:00 p.m. The pre-closed period is from 4:00 p.m. to 5:00 p.m.

Suppose an appointment for an examination of a half-hour duration and a target of 3:00 p.m. is needed. First consider the appointment-time-determination rules:

```
//Appointment time slot availability rule
if (there is an incompatible appointment in the interval Target+Duration) return(NotAvailable)
else return(Available)
// Use Target as the appointment time if possible
if (Target == Available) return {Target}
and the relevant appointment-type-compatibility rules:
// Appointments of the same type cannot be scheduled concurrently
if  (DesiredType == CurrentType)  return NO
// An examination can be scheduled to begin concurrently with an ongoing treatment
if (DesiredType == Exam && CurrentType == Treatment)  return YES
```

The new appointment is compatible with the treatment scheduled for 3:00 p.m. However, the Target is not an available appointment time since the examination currently scheduled at 3:15 is not compatible with the new appointment.

A new Target is proposed by searching the appointment calendar forward and backward. The new Target is 3:15 p.m. The relevant appointment-time-determination rules are as follows.

```
// Find closest available appointment time following target
if (Target != Available)  BestTimeForward = ClosestAppointmentForwardEndTime
// Find closest available appointment time preceding target
if (Target != Available &&
   ClosestAppointmentForwardStartTime – Duration < ClosestAppointmentBackwardEndTime)
   BestTimeBackward = ClosestAppointmentBackwardEndTime
// Make sure BestTimeBackward is not the same as the Target
if (BestTimeBackward == Target) BestTimeBackward = Target – TimeBetweenAppointments
// New Target is the closer of BestTimeBackward and BestTimeForward to the current Target
if (Target – BestTimeBackward => BestTimeForward – Target) return BestTimeForward
```

The iteration shown in Figure 5 processes the new target time of 3:15 p.m. This target time is rejected and a new target time of 3:30 p.m. proposed in the same way that 3:00 p.m. was rejected and 3:15 p.m. was proposed. The target time of 3:30 p.m. is accepted based on the rules:

```
//Appointment time slot availability rule
if (there is an incompatible appointment in the interval Target+Duration) return(NotAvailable)
else return(Available)
// Use Target as the appointment time if possible
if (Target == Available) return {Target}
```

7. Summary

An expert system can be used to model the scheduling of patient appointments. Such an expert system can be integrated with a simulation model to aid in the design of a medical facility such as a small animal veterinary clinic. The integration is accomplished by implementing the expert system so that it may be invoked as a subprogram whenever needed by the simulation model.

Four types of rules are identified for modeling patient scheduling. Appointment-type-compatibility rules prevent concurrent scheduling of appointments with conflicting resource requirements. Appointment-time-determination rules search the appointment calendar for a requested appointment time. Closed-facility rules prevent appointments from being scheduled when the patient care facility is closed and current-time rules prevent appointment scheduling before the current-time.

Specific rules for modeling patient scheduling in a small animal veterinary clinic have been presented. An illustration of the appointment schedule generated by these rules has been shown.

8. References

[1] Steward, D. and Standridge, C.R. "The Application of Simulation to the Design and Operation of Veterinary Medical Practice—Part I." *American Journal of Veterinary Medicine*, March 1996.

[2] Steward, D. and Standridge, C.R. "The Application of Simulation to the Design and Operation of Veterinary Medical Practice—Part II." *American Journal of Veterinary Medicine*, March 1996.

[3] Steward, D. and Standridge, C.R. "A Veterinary Practice Simulator Requiring the Integration of Expert System and Process Modeling." *SIMULATION*, Vol. 66, No. 3, pp 143-159, 1996.

[4] Fishwick, P.A. "An Integrated Approach to System Modeling Using a Synthesis of Artificial Intelligence, Software Engineering, and Simulation Methodologies." *ACM Transactions on Modeling and Computer Simulation*, Vol. 2, No. 4, pp 307-330, 1992.

[5] O'Keefe, R. "Simulation and Expert Systems – A Taxonomy and Some Examples." *SIMULATION*, Vol. 46, No. ##, pp 10-16, 1986.

[6] Nadoli, G. and Biegel, J.E. "Intelligent Manufacturing-Simulation Agents Tool (IMSAT)." *ACM Transactions on Modeling and Computer Simulation*, Vol. 3, No. 1, pp 42-65, 1993.

[7] Burns, J.R. and Morgeson, J.D. "An Object-Oriented World-View for Intelligent Discrete, Next-Event Simulation." *Management Science*, Vol. 34, No. 12, pp 1425-1440, 1988.

[8] Robinson, S., Edwards, J.S. and W. Yongfa, W. "An Expert Systems Approach To Simulating The Human Decision Maker." *Proceedings of the 1998 Winter Simulation Confererrnce*, D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan (eds.), pp 1541-1545, 1998.

[9] Taylor, R. and Hurrion, R.D. "An Expert Advisor for Simulation Experimental Design and Analysis." *AI and Simulation*, T. Henson (ed.), Society for Computer Simulation, pp 238-244, 1988.

[10] Tao, Y.-H. and Nelson, B.L. "Computer-Assisted Simulation Analysis." *IIE Transactions*, Vol. 29, No. 3, pp 221-232, 1997.

[11] Mellichamp, J.M. and Park, Y.H. "A Statistical Expert System for Simulation Analysis." *SIMULATION*, Vol. 52, No. 4, pp 134-139, 1988.

[12] Ramachandran, V., D. L. Kimbler, and G. Naadimuthu, "Expert Post-Processor for Simulation Output Analysis." *Computers, Industry, and Engineering*, Vol. 15, No. 14, pp 98-103, 1988.

Charles R. Standridge is an Associate Professor in the Padnos School of Engineering at Grand Valley State University in Michigan. He has more than 25 years of simulation experience in academia and industry. He has performed many simulation applications, developed commercial simulation software, and taught simulation at three universities. His current research interests are in the development of modular simulation environments (MSE). He is working with industry on the application of MSE to strategic and tactical logistics problems. Dr. Standridge's simulation teaching interests are in the use of computer-aided teaching studios for instruction of introductory undergraduate and graduate courses using a case-based approach. He has a PhD in Industrial Engineering from Purdue University, and is an Associate Editor of *SIMULATION* in the areas of Logistics, Inventory, and Supply Chain Management.