

## A DISCRETE EVENT MODEL OF CLINICAL TRIAL ENROLLMENT AT ELI LILLY AND COMPANY

Bernard M. McGarvey  
Nancy J. Dynes  
Burch C. Lin  
Wesley H. Anderson  
James P. Kremidas  
James C. Felli

Eli Lilly and Company  
Lilly Corporate Center  
Indianapolis, IN. 46285, U.S.A.

### ABSTRACT

Clinical trials constitute large, complex, and resource intensive activities for pharmaceutical companies. Accurate prediction of patient enrollment would represent a major step forward in optimizing clinical trials. Currently models for patient enrollment that are both accurate and fast are not available. We present a discrete event model of the patient enrollment process that is accurate and uses relatively small CPU times. This model is now being used on a regular basis to predict the enrollment of patients for large trials with around 13,000 patients and has led to significant reduction in the time it takes to make these predictions.

### 1 INTRODUCTION

The management of clinical trials used to determine the safety and efficacy of new pharmaceutical products represents a major investment in resources for pharmaceutical companies. Such clinical trials consist of a number of complex and interdependent tasks, including: protocol development and approval, trial site selection, and patient identification, selection, and enrollment. Clinical trials also represent a major opportunity for pharmaceutical companies to optimize the overall new drug approval process. Some of the issues that can occur in the clinical trial patient enrollment process include:

- Delay of study completion due to poor enrollment. Approximately 80% of clinical trials across the industry do not complete enrollment as planned, resulting in increased clinical operations expenses, cancelled trials, and loss of future revenues from delayed submissions.
- Over-enrollment of patients to provide a safety factor. Each extra patient that adds to the cost of a trial without improving the statistical value of the

analysis results in excess and unnecessary expenses. Subsequently such patients may be excluded from the trial and so the resources used to enroll these people are wasted.

- Mismatch of supply and demand for resources used in the trial. When resources such as clinical trial material are not synchronized with the availability or location of enrolled patients, wasted materials or delayed study completion can result.

Felli et al. (2007) give a good discussion of current efforts to improve the patient enrollment process for clinical trials. In particular, the ability to predict the enrollment rate of patients in a clinical trial is presented as an important element of improving the clinical trial process overall. A theoretical model based on semi-Markov chain analysis developed at Eli Lilly (Felli et al. 2007) showed that modeling patient enrollment can lead to useful predictions. This model allowed the prediction of total patient enrollment rates based on the rates of enrollment at individual sites, anticipated patient drop out rates and so on.

After this initial success, focus groups were created at Eli Lilly to evaluate the usefulness of the general model as a practical tool for the ongoing prediction of clinical trial enrollment. The main issue identified by these groups concerned the model execution time. Even for a relatively small clinical trial involving 700 patients over a 300 day simulation time horizon took about  $2\frac{3}{4}$  hours on a desktop computer. Since the original vision for using the prediction model would allow users to do what-if scenario analysis as they sat at their desktop computers, such a long execution time was clearly unacceptable. Based on discussions with users the following goals for execution times were set:

- small trials (<1000 patients) should take on the order of 5 minutes or less to simulate
- large trials (>10,000 patients) should take on the order of 30 minutes or less to simulate.

In the rest of this paper we describe the patient enrollment process, the model we developed to address the problem of execution time, the results obtained, and the future plans for the model.

## 2 THE PATIENT ENROLLMENT PROCESS

A brief description of the patient enrollment process now follows. The main components of the process are shown in Figure 1.

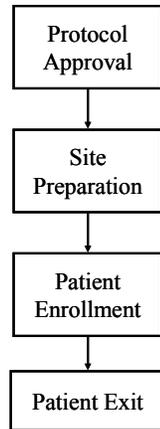


Figure 1: The main components of the patient enrollment process.

The process has three main steps:

- The process begins by having a protocol approved. This protocol must be approved before any site in the trial can begin any task associated with the trial. The protocol approval date is uncertain so a range of protocol approval dates must be specified. Once the approval date is reached the process sends a signal to the site preparation step that the protocol is approved and the trial sites can begin their preparation process.
- Once approved, this protocol triggers all the sites that are part of the trial to begin the appropriate preparation processes so that they can begin to enroll patients. Once protocol approval is obtained each site independently completes its preparation process. This preparation step involves each site creating its own local plans to ensure the successful enrollment of patients and a site may fail to execute these tasks adequately. If a site fails its preparation process, it is excluded from the remainder of the process. If a site is successful, it can begin to enroll patients and so the process moves to the patient enrollment step. The key data for this part of the process includes the time it takes a site to get prepared and the probability that a site will be successful at this preparation step.

- Once a site is prepared, it begins to enroll patients. Based on the specific approaches used by a site, patients will contact the site for the first patient visit. Some patients will not be deemed suitable for the trial and so will exit the process at this point. The remainder enter what is referred to as a washout period. This is a period where some patients decide, for whatever reason, not to continue with the trial and so they exit the process. Those patients who are still in the process at the end of the washout period are now randomized and are brought back in for the first (randomized) patient visit. After this visit some patients may again drop out. The process boundary is considered to be those patients that remain in the trial after the first patient visit. The key data for this part of the process include the rate at which patients initially enroll (which varies from site to site), the time patients spend in the washout period, and the failure probabilities at the first patient visit, the washout period and the first randomized patient visit.

The total patient enrollment across all sites is the process output of interest. This can be described by an enrollment profile over time. Each patient enrollment trial has its own target for the number of patients required to complete enrollment (this number is specified as part of the protocol referenced earlier). The trial is considered complete when the total number of patients that has enrolled equals the enrollment target. Further, each site has its own individual target for patient enrollment so that a site will stop enrolling patients once its individual target has been reached. This aspect of the enrollment process, plus the fact that different sites begin to enroll at different times, give a typical enrollment profile an “s” shape (slow enrollment rate at the start, fast in the middle, slow towards the end), as shown in Figure 2.

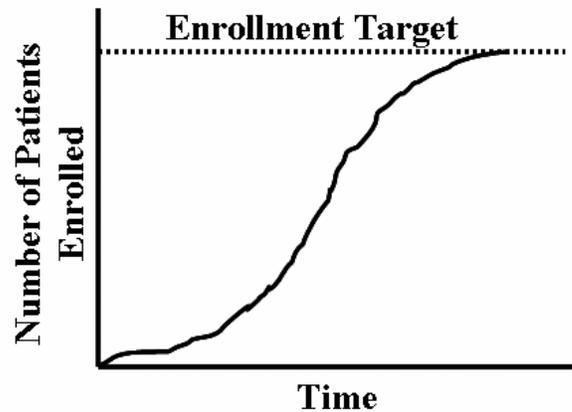


Figure 2: Typical patient enrollment profile

In trying to predict the profile shown in Figure 2, the input data to a prediction model will be uncertain and so will be stochastic in nature. For example, whether a particular site will be successful at the preparation step and will get to enroll patients is uncertain. This uncertainty is expressed by the probability of success for the site. Another example is the time a patient spends in the washout period, which is a random variable and is described by a probability distribution. Because of the stochastic nature of the model, every time the model is run to predict the enrollment profile, a different profile will be obtained. In order to get a good representation of the possible behavior of the predicted profile, multiple runs must be performed. The more runs performed, the better the model covers the possible behavior of the actual trial enrollment. Therefore the total execution time required to make a prediction depends on the time to execute a single run multiplied by the number of runs required to get a good prediction (section 4 quantifies what we mean by a good prediction). In analyzing the current model and looking for improvements, each of these aspects of the model were analyzed separately.

### 3 MINIMIZING EXECUTION TIME PER RUN

The original model was based on a semi-Markov analysis. This type of model involves defining a set of states that each patient can have at any given time and then to model how each patient moves from state to state. A model is considered to be Markov if the times that patients spend in a given state are independent, identically distributed random variables drawn from an exponential distribution. A semi-Markov process is a Markov process where the restriction to exponential distributions is relaxed. A semi-Markov process therefore allows us to model actual distributions found from historical data. Either approach is represented by a state transition model. A typical state transition model S is shown in Figure 3.

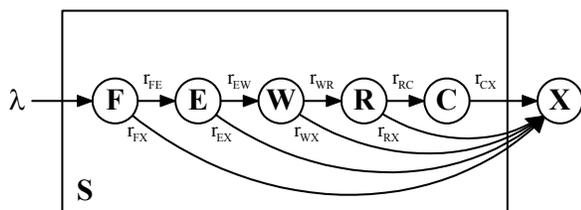


Figure 3: State transition model for the semi-Markov model

This transition model has the following components:

- The allowed states (F, E, W, R, C, and X) of the system. X is an absorbing state as it represents the exit of patients from the process.

- The transition probabilities  $r_{S1,S2}$ . These values specify the probability that a patient in a given state S1 will transition to a state S2 on a given day.
- The arrival rate  $\lambda$ . This represents the rate at which patients enter the process.

The semi-Markov model updates the states on a given frequency. For the original model this was set at a daily frequency, so that the transition probabilities  $r_{S1,S2}$  based on this daily update frequency. Clearly if the update frequency were weekly instead of daily, the  $r$  values would be 7 times larger. Since the model must update the state of all patients at every update time, the choice of the update frequency has a direct impact on the execution time for a single run of the model. We could minimize the execution time by making the update frequency large. However, this will reduce the level of detail within the model – a patient that should leave a state in the middle of the week cannot now leave until the end of the week. Therefore increasing the update frequency from daily to weekly (or monthly) was not considered an option to reducing the model execution time.

What we needed was a modeling approach that does not need to monitor the state of each patient at each update period and indeed does not use the concept of an update period at all. Such an approach exists and has been in use for quite some time – discrete event (DE) modeling. In DE modeling, when the patient enters a state such as a waiting period, since we know the time the patient should spend in the state, we can specify the time the patient will exit the state. This is shown in Figure 4.

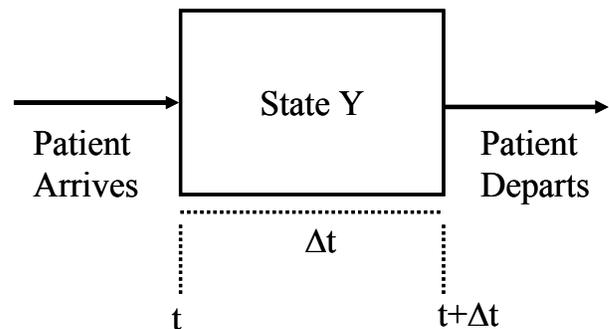


Figure 4: State transitions in discrete event (DE) modeling

When the patient arrives at state Y at time  $t$ , we know that the patient must exit the state at time  $t + \Delta t$ , where  $\Delta t$  is the time the patient expects to stay in state Y. The DE model schedules an event “patient leaves state Y” for time  $t + \Delta t$  and executes this event at the correct time. We do not have to keep monitoring the state of the patient between times  $t$  and  $t + \Delta t$  since we know the patient will not change state between  $t$  and  $t + \Delta t$ . Note also that this does not depend on  $\Delta t$  being a deterministic value. Even if  $\Delta t$  is a random variable chosen from a probability distribution, the exit time

will still be at  $t+\Delta t$ . It was therefore decided to port the model over to a DE modeling environment.

At Eli Lilly, we had experience using the Extend software package, from ImagineThat, Inc, to set up discrete event models. It is a block based approach to simulation where complex systems can be built up from elemental building blocks such as queues and tasks. We therefore decided to model the patient enrollment process in Extend to see if we could meet our simulation speed challenge. It is not the purpose of this paper to discuss the details of DE simulation. There are many available references that give in-depth description of the approach (Fishman 2001, Laguna and Marklund 2004, Leemis and Park 2005, Banks et al. 2000).

In porting over the model to a DE environment, we were also able to better represent a key step in the process that had proved difficult for the Markov based model. When a patient is in the washout period, we must know

- the probability that the patient will move to the next step
- the probability that the patient will exit the process (move to state X in Figure 3).

The probability of the patient staying in the step is obviously 1 minus the sum of these two probabilities. However, there was no historical data available on the pattern of patient drop out from the washout period. So while we knew the fraction of patients that moved from the washout state to state X we did not know when they moved to state X. The only data available was for patients who moved onto the next step in the process, the first randomized patient visit. In the Markov model, assumptions had to be made in order to estimate the probability that a patient would move to state X while in the washout state. In the DE environment this did not prove to be an issue. We simply let all patients stay in the washout period until the end of the washout period. The distribution of times all patients spend in the washout period is the same as the distribution for those patients who exited the washout period and moved to the first randomized patient visit, for which we had historical data. Then when a patient exits the washout period, we randomly send patients to state X using the exit probability estimated from historical data. Thus the DE model will correctly predict the flow of patients through the process but not the exit pattern. This is acceptable since we are only interested in the former.

As the DE model was being built, we paid a lot of attention to building a streamlined model so that the execution speeds could be minimized. One useful tool provided in Extend is the ability to track the CPU time used by each block in a simulation. This allowed us to identify a database access block that was taking a lot of CPU time and to realize a 25% CPU usage reduction by recoding the block. Another aspect of our streamlining efforts was to only include tasks in the model that really need to be there. Therefore the DE model itself does little or no analysis of the re-

sults. Rather the basic results are exported to an Excel file for analysis. Not only does this save CPU time but it provides the results in an environment that most users are comfortable with and so encourages such users to develop their own approaches to analyzing the data.

Finally, as we were porting over the model to a DE environment, we took the time to make some other improvements to the model that the original focus groups had requested. These improvements would not have stopped the model from being used, but they would accelerate its usage. Some of these improvements include:

- The interface is now more “user-friendly” and intuitive. In fact the user interface now consists of Excel files for data input and output and an easy to use dialog box style console to run the model.
- The ability to easily add or delete sites.
- The model is now calendar date driven. This includes both specifying input data (such as seasonal effects) and presenting results (such as enrollment profiles) by calendar dates as distinct from simulation time.
- Input data can now be specified on an individual site basis. In the original model, sites were grouped by clusters and input data could only be specified on a per cluster basis.
- It is now easy to roll up site data by group or country. A group is defined by some common site attribute other than country. Typically this group attribute has been used to indicate the sites enrollment speed (high enrolling sites to low enrolling sites). This feature allows us to create a master model that includes all groups and geographies and then only run the model for different groups or geographies. This helps users look at their own piece of the trial while still allowing them to see the big picture.
- It is now possible to easily show enrollment curves by day, week, or month.

#### 4 MINIMIZING THE NUMBER OF RUNS REQUIRED

As noted earlier, because the model is stochastic, multiple runs must be performed to get a full representation of the model prediction. Therefore we must minimize the number of runs performed while maintaining the quality of the model prediction. In order to do this, we must first define a metric for this quality of prediction. Then we can stop the model when the quality of prediction metric is better than some defined target. In order to define such a metric, consider a set of  $n$  runs where each run generates a patient enrollment profile  $E$ . Further suppose that we measure the enrollment profile at  $m$  discrete time points. Then  $E_{\theta,i}$  represents the patient enrollment at time point  $\theta$  for run  $i$ , where  $1 \leq \theta \leq m$  and  $1 \leq i \leq n$ . At each time point  $\theta$  a confi-

dence interval,  $CI_\theta$ , can be defined for the value of E across all n runs.

$$CI_\theta = \overline{E}_\theta \pm D_\theta, D_\theta = t_{\alpha, n-1} \frac{s_\theta}{\sqrt{n}}$$

$\overline{E}_\theta$  is the average of the profile at time point  $\theta$ ,  $D_\theta$  is the half-width of the confidence interval at time point  $\theta$ ,  $t_{\alpha, n-1}$  is the Student t distribution at a confidence level  $\alpha$  with n-1 degrees of freedom, and  $s_\theta$  is the sample standard deviation of the enrollment profile at time  $\theta$ , given by

$$s_\theta^2 = \frac{1}{n-1} \sum_{i=1}^n (E_{\theta,i} - \overline{E}_\theta)^2 \quad \overline{E}_\theta = \frac{1}{n} \sum_{i=1}^n E_{\theta,i}$$

The half-width  $D_\theta$  reflects the uncertainty with which we know the profile average at time point  $\theta$ . Therefore we can define an average (normalized) uncertainty across the entire profile as our metric of quality, Q, as follows

$$Q = \frac{1}{m} \sum_{\theta=1}^m \frac{D_\theta}{\overline{E}_\theta} \tag{1}$$

The value of Q is normalized using the value of  $\overline{E}_\theta$  so that Q represents a relative uncertainty (uncertainty relative to the average) and is expressed as a %. The smaller the value of Q the higher the quality of the prediction. A target can then be established for Q, say  $Q^*$ , so that the model stops when  $Q \leq Q^*$ . Typically a value of 1% has been used for  $Q^*$ . This means that users can set up the model to stop when the average uncertainty of the profile is 1% or less of the average profile (the alternative is to let the model run for a specified value of n, irrespective if the value of Q). In implementing this quality metric, the following enhancements were made to make it more useful:

- Since it is always possible to get a value of Q less than  $Q^*$  by chance when Q is close to  $Q^*$ , users have the option of specifying that the model will not stop unless Q is  $\leq Q^*$  for a sequential numbers of runs (referred to as the “Number of Repeats” in the model user interface).
- Since the relative uncertainty can be large in the early parts of the profile, the user has the option of omitting some of the initial time points in (1). This is equivalent to starting the summation in (1) at some point  $\theta > 1$  rather than at  $\theta = 1$ . This is referred to as the “Starting Point” in the user interface. The model output contains a profile of the uncertainty ( $D_\theta$  over time) to make it easy for the users to decide an appropriate value for the “Starting Point”.

The model runs for a given time horizon which is specified by the user. We decided not to stop an individual run of the model when the total enrollment reached the enrollment target. The reason for this is that this approach would have led to different length enrollment profiles for each run (in (1), m would be different for each run). This would have made the calculation of Q more complicated and so we did not pursue this approach. When the user sets up the model for a new trial, they set the model time horizon to be much longer than they know they will need and then they set the number of runs for the model to 2 or 3. This small number of runs is sufficient to allow users to define a good model time horizon.

## 5 RESULTS

The results from the model will be presented in two parts. Firstly we will look at how well the model predicted previous trials for which we have data. Secondly we will look at the CPU times required for running the model to see if our original goals were met. Six trials were chosen to demonstrate the results from the model. The first four (trials 1 to 4) were completed trials where actual enrollment data was available. Trial 5 was a small, ongoing trial used as a “reality check”. Trial 6 was a large, ongoing trial (13,000 patients) used specifically to explore the feasibility of our speed goal for large trials. Table (1) shows pertinent data that illustrate the size of the trials modeled.

Table 1: Patient enrollment trials modeled.

Trial #	1	2	3	4	5	6
# of Sites	22	33	28	25	69	631
# of Patients Required	400	480	320	300	400	13,000
Time Horizon of Model (Days)	343	448	399	560	183	574

### 5.1 Predicting Historical Trials

Since the main purpose of this paper is to focus on the model execution time, only one prediction curve will be presented here to illustrate the typical prediction capability of the model. The enrollment prediction for trial #1 is shown in Figure 5.

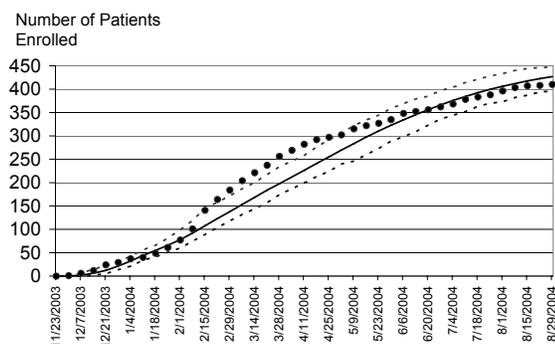


Figure 5: Comparison of actual and predicted enrollment for trial 1. Full line represents the average predicted profile, the dashed lines represent the minimum and maximum values for the predicted profile and the solid circles represent the actual trial data.

Note that the model works well at the beginning and the end of the trial (from Table 1 this trial required 400 patients to complete enrollment). In the middle, however, we observe some disagreement. We interpreted this as an indication that some sites may have enrolled faster than predicted at the beginning of the trial, but later slowed as their enrollment reached its goal. This deviation should not be interpreted as a major deficiency in the model but rather as an opportunity to learn about the patient enrollment process. Since we know what data we have used to generate the model results, we can now look to identify gaps in our knowledge and hence improve future predictions.

### 5.2 Model Execution Time

The model execution time data for all six trials are shown in Table 2.

Table 2: Execution time data for all six trials. All runs were done on a laptop computer with a 1.73 MHz processor and 1 GB of RAM.

Trial #	1	2	3	4	5	6
Average # of Patients Enrolled	450	584	395	549	650	15,500
# of Runs	100	100	100	100	100	16
Final Value of Q	2.2	1.9	2.3	2.4	0.74	0.94
CPU Time (Minutes)	1.0	1.5	1.2	1.7	2.3	3.0

Looking at the first 5 trials, it is clear that the CPU time results for small trials is acceptable since all CPU times were on the order of 5 minutes or less. Further optimization could have been done to get the times even lower. For

example, each trial enrolled more patients than required so the time horizon of the model could have been reduced thereby reducing the CPU time required per run.

Speed optimization was critical to getting reasonable execution speeds for trial 6. Given the user’s original dataset, each run took about 30 seconds, and so 100 runs would have taken about 50 minutes. Looking at the results with just 2 runs, we saw that, on average, the model was enrolling just over 26,000 patients, twice as many as the 13,000 patients required to complete the enrollment. We therefore reduced the simulation end time so that about 15500 patients were being enrolled on average, still 20% greater than what the trial required. This reduced the CPU time per iteration to about 12 seconds. Next, we considered the stop criterion. Initially the stop criterion for the model was based on the number of runs,  $n$ , and after 100 runs, the value of  $Q$  was of the order of 0.3 indicating a level of accuracy not required. Therefore we set the value of  $Q^*$  to 1 and set the stop criterion to monitor  $Q$ . Also, we looked at the profile of the relative uncertainty,  $D\theta$ , and saw that it was very large in the initial section of the profile, as shown in Figure 6.

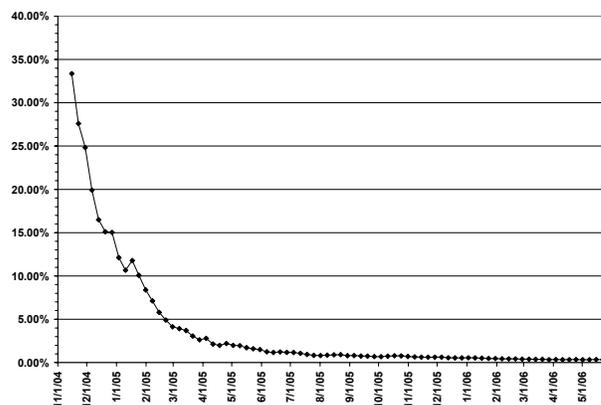


Figure 6: Profile of  $D_0$  for trial #6

Therefore we set the “Starting Point” to 20. With these changes, we required only 16 iterations to get a  $Q$  value less than 1 resulting in a 3 minute execution speed, well within the original goal of 30 minutes. And of course we could also have looked for a faster computer than the laptop computer we used to generate the data in Table 2.

## 6 DISCUSSION AND FUTURE WORK

### 6.1 Results to Date

The main result of the new version of the model was its development as a practical tool that is now being used regularly by those individuals responsible for predicting clinical trial enrollment. The model development cost was

small, taking only about 150 man-hours to date. The speed of the simulation together with the productivity tools we have put in place to simplify the Excel input database setup has reduced the time required to make a typical prediction from about 80 hours to 30 minutes. This has made scenario analysis for clinical trial predictions a reality. Further, this has given analysts more time to spend on the analysis of results rather than model setup, leading to a significant increase in the accuracy of the predictions made by the model. As users continue to gain confidence, this improved accuracy will enable better resource allocations, thereby reducing waste associated with the unpredictability of the patient enrollment process.

The current version of the model contains some features that we have yet to exploit fully. The most important of these is patient tracking. As patients move through the DE model, various attributes can be attached to the patient. Examples of these attributes include:

- The time the patient entered the washout period.
- The time the patient exited the washout period.
- If the patient dropped out of the washout period.

As patients exit the model, these attributes can be logged to a data file, one row per patient. Given that a trial might involve 10,000 patients and we might do 100 runs, this means that a model could generate 1,000,000 rows of data. This data can be easily imported into a statistical analysis package for further analysis. Datasets with such large amounts of data are an analyst's utopia! This feature became affectionately known as "virtual patients". It also proved very valuable in the initial debugging of the model when being able to track individual patients allowed us to see where all the patients were going and hence to see if patients were ending up where they should not be.

## 6.2 Future Plans

One of the major features that will be added to the model in the next version will be the ability to initialize the model using data from a trial already in progress. Traditionally initialization of a simulation has been difficult because as patients are added at various points in the model structure the "history" of the patient must also be added. For example, a patient might be treated differently at a later stage of a model due to conditions the patient saw earlier in the model. The conditions move with the patient as attributes of the item representing the patient. If at initialization the patient is entered into the model after the point where the conditions were set, the item does not have the required history. The history must be created as the patient enters at the later stage. Fortunately recent versions of Extend have provided tools that facilitate the addition of this history and we have set up the current version of the model to enable the addition of these tools in the future.

Another potential improvement to the current model is to enable the use of the predicted enrollment profiles as a

tracking tool for a trial. A simple approach to this would be to place control limits on the enrollment profile and then track the actual trial results to see if they stay within the control limits. Control limits could be placed, for instance, at the mean  $\pm 3$  standard deviations of the multiple run results at each profile time point. The best way to implement such a tracking system remains to be investigated. Whatever approach we finally identify, it must address the following issues:

- successive points on the enrollment profile are correlated.
- the "s" shape of the typical enrollment profile means that the likely deviation of an actual trial profile from the predicted profile will not be constant across the profile but will be largest in the middle of the "s".
- how to "re-start" the control chart if an "out of control" point occurs and we have taken some intervention as a result.

The scope of the model may be expanded to cover more of the enrollment process, both at the beginning and the end. The model currently begins simulating patient flow when they apply for initial enrollment in the trial; there is a pre-screening step that could be included once data become available. On the other side of the process, the model ends at the first randomized patient visit. We have designed the model to easily accommodate more patient visits. Also the level of detail in the model can be increased. For example, more details could be added to the protocol approval and site preparation steps. Of course, as the scope and detail of the model increases, the CPU time will increase and we may have to look at further enhancements to manage the simulation speed. In addition as the scope and detail increases so too will the size of the input database. We intend to provide further productivity tools for the creation of the database to accommodate this expansion. Finally, as model use increases, we believe other opportunities for improvement will be identified.

## ACKNOWLEDGMENTS

The authors would like to thank Mr. Roger Scott of Eli Lilly and Company for his technical review of this paper.

## REFERENCES

- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-event system simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- Felli, J. C., W. H. Anderson, J. P. Kremidas, and S. J. Ruberg. 2007. A semi-Markov model for patient progression through clinical trials. *European Journal of Operational Research* 176:542 – 9.
- Fishman, G. 2001. *Discrete-event simulation*. USA: Springer.

Leemis, L., and S. Park. 2005. *Discrete-event simulation : A first course*, USA: Prentice Hall.

Laguna, M., and M. Marklund. 2004. *Business process modeling, simulation, and design*, USA: Prentice Hall.

## **AUTHOR BIOGRAPHIES**

**BERNARD MCGARVEY, Ph.D.** is an Engineering Advisor at Eli Lilly and Company. He leads a group of engineers who focus on the use of modeling technologies within process development and manufacturing. He has previously held positions in process engineering, process control engineering and technical services. One of his current interests is in developing efficient methods for creating process simulations.

**NANCY J. DYNES** is a Clinical Trial Enrollment Operations Consultant at Eli Lilly and Company. She manages productivity metrics for the global clinical trial operations organization, analyzes internal and external databases to develop accurate enrollment predictions, and serves as the internal expert on enrollment data.

**BURCH C. LIN** is a Clinical Trial Enrollment Operations Analyst at Eli Lilly and Company. He provides data and metrics that support the decision making process involved in clinical trial patient enrollment.

**WESLEY H. ANDERSON** is a Business Process Consultant in the Medical Business Systems Group at Eli Lilly and Company. In this role he provides statistical analysis for clinical operations. In addition, he provides training and consulting for the Six Sigma program at the company. Prior to this he has provided statistical support to manufacturing operations.

**JAMES P. KREMIDAS** heads the Global Enrollment Optimization Group at Eli Lilly and Company. His organization is within Global Clinical Operations and is focused on predicting and accelerating the enrollment rates for all corporate studies. He has previously held positions in the chemistry lab, marketing research, sales and procurement.

**JAMES C. FELLI, Ph.D.** is a Senior Research Advisor at Eli Lilly and Company in the Decision Sciences Department. In addition to advising senior management, his responsibilities include designing and delivering decision analysis courses, developing new decision tools, and engaging in applied research and internal consulting. His current research interests focus on the modeling and analysis of stochastic multi-criteria decision problems, enriching communication between analysts and non-analysts, and developing new methods of data visualization.