Emergency Department Patient Flow Simulations Using Spreadsheets

Michael G. Klein, MBA;

Gilles Reinhardt, PhD

Introduction: Patient flow computer simulations allow Emergency Department stakeholders to assess operational interventions, develop utilization and performance measures, and produce estimates for budgeting or planning purposes. Key challenges of traditional discrete-event computer simulation software are their inherent complexity for modeling, coding, or analyzing output and their significant costs and training. We propose a simulation platform that runs in spreadsheets. Because of their low cost, popularity and powerful functionality and performance, spreadsheets also allow for the development and management of simulations that efficiently output results that are just as reliable as those from traditional software.
Methods: A spreadsheet simulation is developed by modeling one row as one simulated minute (more than 20,000 rows for a 2-week period). Uncertainty in arrivals, patient type, routing, and treatment times is modeled using the "rand()" function to simulate the state of the Emergency Department at a given point in time. The patient is tracked with embedded "if()" functions and summary statistics are obtained through range functions. We use an equivalence test to determine whether the resulting average length-of-stay figures are the same as those of a traditional simulation platform.
Results: We find little significant difference in average length-of-stay figures between both models.
Conclusions: Spreadsheet simulations are as effective as traditional simulations but easier to use, understand, and implement. Spreadsheet software is widely available, at a fraction of the cost of discrete-event simulation software. Coding spreadsheet simulations may be more challenging as it requires a different and more novel expertise than traditional computer programming. However, spreadsheets can be organized to reference existing datasets, thus minimizing the burden of copying and likelihood of transcription errors and information leakage. Output analysis can also be customized with user-specific performance statistics and charts.
(*Sim Healthcare* 7:40–47, 2012)

Key Words: ED administration, ED operations, Patient flow modeling, Discrete event simulations, Spreadsheets.

Because Emergency Department (ED) services are critical, in high demand and open to the public, their overall process structure (patient routing, staffing, resource schedules, and task allocations) is subject to ongoing analysis and interventions to ensure that appropriate care is provided when and where needed. Therefore, operational and strategic improvements should be carefully tested before implementation so that disruptions to ED services are kept to a minimum. Discrete-event simulations are widely used to model a number of healthcare settings. They have been successful at assessing ED performance and efficiency by simulating patient flows through current or proposed services to assess the value and impact of potential expansion, resource upgrades, schedul-

ing policies, or other improvements.[1–3] They are particularly useful when testing process or resource change[4] because, albeit with limitations, their output can be generated and analyzed at minimum costs to the institution and zero risk to quality of care and patient safety. In addition, the process nature of an ED (abstracted as a connected set of staffed activities with flow units routed through them based on priority) lends itself well to a process-based computer model such as a simulation.[5,6]

An effective simulation platform should be capable of modeling the flexibility inherent to ED operations with minimum effort from the user and be sufficiently powerful to produce relevant and accurate results quickly. Efforts to model a process into a simulation program, validating and testing it, and processing and interpreting its results and statistics often rest as much with simulation experts as with clinical ones. Before using a simulation to draw conclusions or provide recommendations, one must first properly represent and build the blocks for the modeled process, flows, resources, and constraints; acquire accurate figures and other input to the system; and validate the model against empirical evidence.
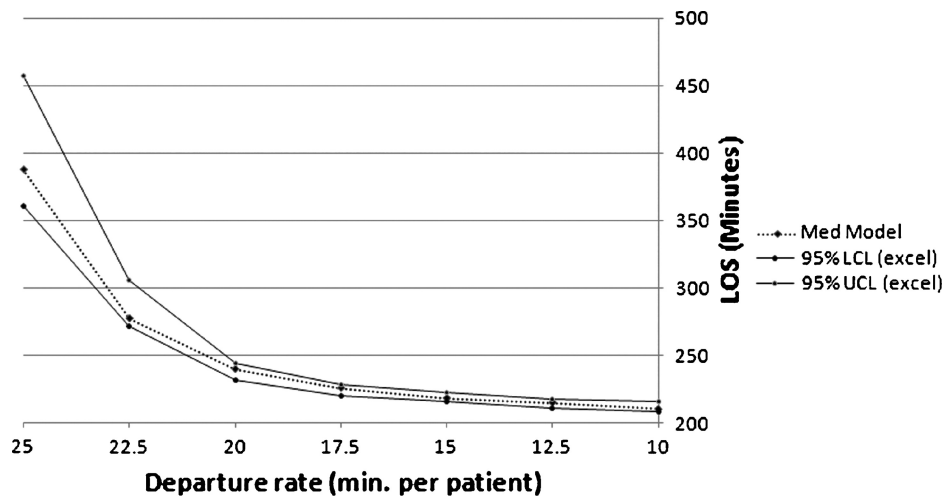
Many traditional simulation platforms are available to carry out such analyses. Professional versions of such

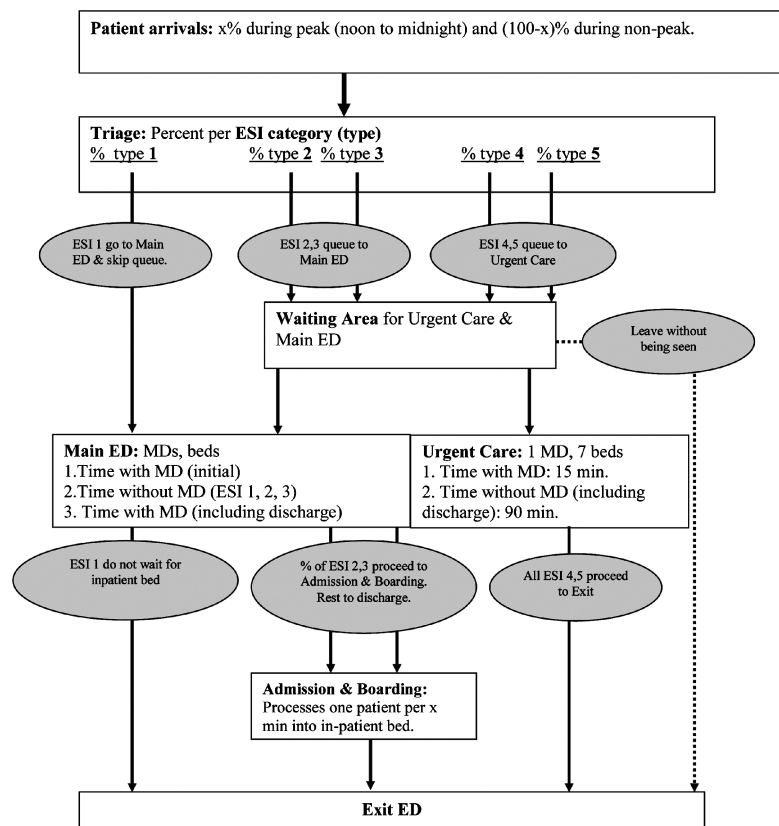**Figure 1.** Simulated LOS: MedModel Average; Excel 95% CI, 23 beds.

software carry steep purchasing or licensing costs: $18,500 for the professional version of MedModel (ProModel, Orem, UT)[7] and require significant internal or consulting resources to program, test, and produce accurate estimates.

Take for example one point on the dashed line in Figure 1. The value on the vertical axis represents the simulated estimate over a 2-week horizon of the average length of stay (LOS) under the ED structure in Figure 2 and the in-patient exit rate on the horizontal axis. To minimize the risk of bias and also produce an estimate of the standard deviation of the LOS, the value is obtained by averaging more than 40 replications of the 2-week horizon instance. MedModel

produces text (".txt") files grouped into sets of 40 rows, one for each performance measure. Further manipulation is required to copy and organize the output into a spreadsheet format that allows for summary statistics and hypothesis testing.

There are a number of factors that impact the quality of software (http://www.sqa.net/iso9126.html). We pointed out the cost of MedModel which is over one hundred times the cost of Microsoft Excel (Microsoft, Redmond, WA; http://office.microsoft.com/en-us/buy). Other factors include usability, portability, security, functionality, and performance. In the following paragraph, we argue that usability,

Figure 2. Patient flow diagram.
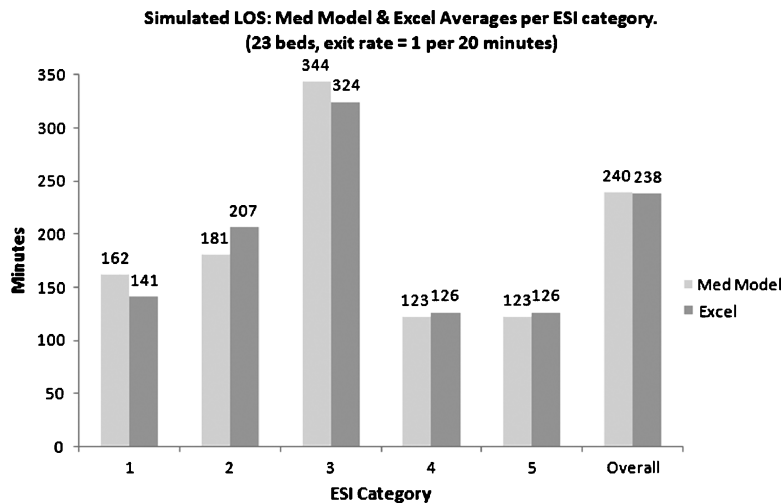
Figure 3. Simulated LOS: MedModel and Excel Averages per ESI category.

portability, and security are as good as or better in a spreadsheet model than in the more traditional platforms such as MedModel, the platform that was used to produce the estimates in Figures 1 and 3 and Table 1.

We propose a user-friendly discrete-event simulation platform based in Microsoft Excel 2007. Spreadsheets have become the norm for quantitative reporting and analysis of data. Administrators and providers alike rely on it to monitor and control the state of the systems for which they are responsible. They can also rely on spreadsheets to assess the value of potential interventions, to develop utilization and performance expectations, to test research hypotheses, or as a medium to long-range budgeting and planning tool. They provide an intuitive landscape to input, chart, and manage data along with efficient and powerful statistical functions to conduct analysis and derive conclusive evidence. As spreadsheet programs are widespread (nearly all office suites include one), a spreadsheet simulation will have enhanced accessibility and visibility, and more stakeholders will be able to take part in its development and management. In addition, patient tracking systems or other electronic systems use table-based storage which can easily be read and manipulated directly into spreadsheets, thus minimizing the likelihood of data entry, transcription errors, and information leakage.

For building and coding simulations, however, spreadsheets require an approach that is distinct from MedModel and possibly more challenging, depending on the user's expertise with either platform. In MedModel, the process, flow, and resource blocks are loosely based on process flow diagrams,[6] so the coding structure is somewhat intuitive and straight-forward but needs to be repeated for every block. The resulting program is a small file that, when run, creates a number of large text files summarizing the performance measures for each replication.

Table 1. MedModel and Excel LOS Comparison

In-Patient Exit Rate	Excel				Med Model		
	Average	SD	95% LCL	95% UCL	Average	SD	% Difference
1 per 25 min	409	131	360	457	388	96	0.054
1 per 22.5 min	288	46	271	305	277	39	0.040
1 per 20 min	238	17	232	244	240	14	−0.007
1 per 17.5 min	224	11	220	228	226	12	−0.007
1 per 15 min	219	10	215	223	218	8	0.004
1 per 12.5 min	214	9	211	218	215	8	−0.003
1 per 10 min	212	11	208	216	210	8	0.010
				In-Patient Exit Rate			
x (%)	1 per 25	1 per 22.5	1 per 20	1 per 17.5	1 per 15	1 per 12.5	1 per 10
2	0.697	0.714	0.189	0.127	**0.043**	**0.032**	0.158
4	0.585	0.496	**0.012**	**0.002**	**0.000**	**0.000**	**0.002**
6	0.466	0.280	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
8	0.350	0.125	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
10	0.247	**0.043**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
12	0.163	**0.012**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
14	0.101	**0.003**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
16	0.058	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
18	**0.031**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
20	**0.016**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**

P values for equivalence test between Excel and MedModel LOS averages (boldface if <5%).

In spreadsheets, the information contained in the blocks is distributed across columns of data through cell formulas. This causes the initial spreadsheet file to be very large compared with MedModel because it reserves cells for all output summary. In addition, development of the cell formulas has to be done in lockstep with the other cells they reference. For example, one piece of information that must be tracked at each minute of the simulation is the "Time Left" in bed for a given patient (see the Methods section). In our model, that formula requires six embedded "if()" functions to properly update the "Time Left" value. Experienced spreadsheet users with such expertise will possibly find the spreadsheet easier to code, whereas those with more procedural and object-oriented skills may yet prefer MedModel.

In addition, simulation estimates from a spreadsheet platform require no additional manipulation upon completion. The text file output from traditional simulation programs store all intermediate values and require significant manipulations after the runs are completed to produce the required output.

The intent of the remaining analysis is to validate the functionality and performance of the spreadsheet model by statistically testing the equivalence of its output compared with MedModel.

METHODS

Spreadsheet Description

Our spreadsheet model for discrete-event simulations rests on two key points:

1. We define a spreadsheet row as one unit of time (generally 1 minute). A 2-week simulation of patient flow therefore contains more than 20,000 rows.
2. The "rand()" function in Excel generates a pseudo-random number uniformly distributed in the open interval (0,1). That number can be transformed into any type of distribution including one constructed

with empirical data. The function also determines all factors that are stochastically resolved during the simulated horizon. In an ED setting, those include triage level, routing, duration of treatment, admission or discharge decision, and availability of in-patient bed.

The set of columns associated with a given row provides the entire *state of the system* for the minute defined by that row. The columns track the prioritized queue of patients in the waiting area, the ED bed occupancy, the medical doctor's (MD's) bedside visits (in length and sequence), and the queue of admitted patients waiting for discharge or transfer into in-patient wards.

Figure 2 illustrates the flow of patients in a given ED as it is coded [except for the patients who leave without being seen (LWBS), those are not modeled in the spreadsheet simulation]. The first process, arrivals, consists in defining a peak period and providing the distribution of total arrivals. Next, patients are categorized by triage priority (*type*) in one of (up to) five levels using a historical or census-based distribution, for example, the Emergency Severity Index (ESI)[8] or the Canadian Triage and Acuity Scale.[9] The type determines the queue priority in the system. In our setting, patients of type 1, 2, or 3 are seen in that priority in the main ED, whereas types 4 and 5, which are lower priority, are routed to the Urgent Care center. The user also inputs the number of beds in each ward and the number of MDs working, along with the expected time spent per type for the initial and final assessments and the time waiting in-between (for labs, consultations, or other results). On the basis of the historical data, the user also enters the proportion of types 1, 2, and 3 that gets admitted to in-patient wards. Figure 4 shows the input area for the Excel simulation.

The formulas are calculated and immediate results for the first replication ensue. Formulas generate the patient flow through the simulated horizon (21,600 rows for 2

Figure 4. Excel simulation input.

43

weeks) and the operational performance for each patient and resource, and aggregate statistical measures over the simulated horizon.

Whether generated through spreadsheets or through a traditional simulation program platform, results from a single replication cannot be immediately projected as reliable estimates for the ED setting being simulated. More precision is needed because the output is based on the generation of streams of random numbers. The results are also biased from the simulation always beginning in an empty (simulated) ED. To mitigate those limitations, simulation programs revert to adding replications to one simulated horizon, extending its horizon and truncating the first day (the *warm-up* period), and averaging over the subsequent ones. For a 2-week horizon, the spreadsheet reserves a set of rows where each row represents a minute: 1,

2, 3, ... , 21,600 where that last number represents 15 days' worth of minutes [(1 + 14) × 24 × 60] which includes the initial warm-up period of 1440 minutes (=24 × 60 = 1 day). The "rand()" function is updated, and based on routing and other model assumptions, columns and neighboring rows are then matched to the random events to effectively simulate the process of patients going through the ED. The results are saved in an aggregate sheet, and a simple macro is run to replicate the process and update the aggregate results sheet. Figure 5 shows the output from the aggregate sheet. Each Excel data point reported on Figure 1 represents an average over 40 replications of a 15 − 1 = 14 day horizon. One data point takes an average of 8 minutes to generate on an Intel Core 2 Duo CPU T7250 @ 2.00 GHz (Intel, Santa Clara, CA) with 2.0 GB of RAM operating on Windows Vista Service Pack 2 (Microsoft, Redmond, WA).

OUTPUT - CURRENT REPLICATION

Triage	Initial # in System (After Day 1)	Number of Arrivals	Number of Exits	# Remaining in System	Avg Boarding Time (minutes)	Avg Length of Stay (minutes)
1	0	6	5	1	N/A	141.17
2	0	63	42	21	126.80	197.68
3	0	100	53	47	79.23	313.63
4	0	41	34	7	N/A	140.38
5	0	33	27	6	N/A	140.38
Overall	*0*	*243*	*161*	*82*	*108.06*	*234.48*

Main ED Resource Utilization

Resource	Utilization % (Peak)	Utilization % (Non-Peak)	Utilization % (Peak & Non-Peak)
Bed 01	91.9%	70.1%	81.0%
...	...	...	...
Bed 07	91.9%	71.2%	81.6%
...	...	...	...
Bed 23	92.8%	69.3%	81.0%
MD 1	96.0%	77.8%	86.9%
MD 2	95.8%	77.4%	86.6%
MD 3	95.7%	76.6%	86.1%
Avg. Bed Utilization	**91.7%**	**71.4%**	**81.6%**
Avg. MD Utilization	**95.8%**	**77.3%**	**86.5%**

Urgent Care Resource Utilization

Resource	Utilization % (Peak)	Utilization % (Non-Peak)	Utilization % (Peak & Non-Peak)
Bed 01	77.6%	38.9%	58.2%
...	...	...	...
Bed 07	78.2%	37.4%	57.8%
MD 1	75.9%	34.5%	55.2%
Avg. Bed Utilization	**77.8%**	**37.7%**	**57.7%**
Avg. MD Utilization	**75.9%**	**34.5%**	**55.2%**

Main ED Patient Wait Times

Waiting for Beds	Total (minutes)	Avg (minutes)
ESI 1	34.00	5.67
ESI 2	714.00	11.33
ESI 3	10870.00	108.70
Total	**11618.00**	**68.75**

Waiting for MD	Total (minutes) - ESI 1	Total (minutes) - ESI 2	Total (minutes) - ESI 3	Total (minutes) - ESI 1-3
Bed 01	0.00	87.00	45.00	132.00
...	...	...	...	...
Bed 07	0.00	155.00	7.00	162.00
...	...	...	...	...
Bed 23	0.00	66.00	45.00	111.00
Total	**3.00**	**1909.00**	**1173.00**	**3085.00**

Waiting for MD	Avg (minutes) - ESI 1	Avg (minutes) - ESI 2	Avg (minutes) - ESI 3	Avg (minutes) - ESI 1-3
Bed 01	0.00	1.38	0.45	0.78
...	...	...	...	...
Bed 23	0.00	1.05	0.45	0.66
Total	**0.50**	**30.30**	**11.73**	**18.25**

Admission & Boarding: In-Patient Bed Waiting Time	Total (minutes)	Avg (minutes)	Avg boarding time (minutes)
ESI 2	1956.00	31.05	126.80
ESI 3	820.00	8.20	79.23
Total	**2776.00**	**17.03**	**108.06**

Total Wait Time	Total (minutes)	Avg (minutes)
ESI 1	37.00	6.17
ESI 2	4579.00	72.68
ESI 3	12863.00	128.63
Total	**17479.00**	**104.03**

Urgent Care Patient Wait Times

	Total (minutes)	Avg (minutes)
Waiting Area	1926.00	26.03

Waiting for MD	Total (minutes)	Avg (minutes)
Bed 1	57.00	0.77
...	...	...
Bed 7	44.00	0.59
Total	**322.00**	**4.35**
Total Wait Time	**2248.00**	**30.38**

Figure 5. Excel simulation output.

Time	Arrival	Triage	Waiting Area
19:10	1	3	:2:3:3:3:3:3:3:3:3:3:3:3:3
19:11	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3
19:12	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3
19:13	1	3	:2:3:3:3:3:3:3:3:3:3:3:3:3:3
19:14	1	2	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3
19:15	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3
19:16	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3
19:17	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3
19:18	1	3	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:19	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:20	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:21	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:22	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:23	0	0	:2:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:24	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:25	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:26	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:27	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:28	0	0	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3
19:29	1	3	:2:3:3:3:3:3:3:3:3:3:3:3:3:3:3

Figure 6. Excel simulation waiting area representation.

Although each row depicts a time unit (minute), columns are used for information about patients, beds, and doctors which together constitute the state of the patients and resources in the simulation model. Each cell (intersection between row and column) represents a piece of information about patients, beds, and/or doctors at the minute in time defined by the row. The columns are ordered in the same sequence as Figure 2. Thus, the first set of columns deal with patient arrivals and the waiting area. For each row, the "rand()" function is invoked to determine whether a patient arrives into the system and another "rand()" function is used to determine the patient type. Once a new arrival is processed, the patient then waits for a bed and is given priority based on the type. As shown in Figure 6, a string of characters represents the queue using the colon character ":" as a separator in between each element in the list. Priority is given to types 2 over 3, so in Figure 6, patients of type 2 are always the first elements in the queue.

Because the next steps in Figure 2 consist of placing patients into ED beds, the Main ED and Urgent Care columns are used next for tracking bed occupancy and doctor utilization. Columns are reserved to store the list of available beds and the list of any beds that are to be filled in the current minute by patients previously in the waiting area. Separate columns were created for each bed to track if a bed is in use, the ESI level of the patient using the bed, whether the patient is waiting for a doctor or not, and the number of remaining minutes that the patient will need the bed (if currently in use).

Figure 7 illustrates the columns used to represent two of the ED beds. The highlighted cell Z1222 uses the following formula to calculate "Time Left" (number of minutes remaining):

$$= IF(Y1222 = 1, Z1221, IF(Z1221>1, Z1221 - 1,$$
$$IF(W1222 = 1, IF(X1222 = "1", \$B\$26, IF(X1222$$
$$= "2", \$C\$26, IF(X1222 = "3", \$D\$26, 0))), 0)))$$

There are four cases that the formula handles:

1. When a patient first enters a bed, the value is set to the total time required (not including wait time).
2. While a patient waits for a doctor, the value is set to be the same as the previous minute.
3. After the first minute when a patient enters a bed, the value is decremented by one from the previous minute (unless the patient is waiting for a doctor).
4. When the bed is empty, the value is set to 0.

In case 1, the total time required depends on the ESI level of the patient. In our model, these values are defined in cells $\$B\$26$, $\$C\$26$, and $\$D\$26$ respectively. We use the

Z1222		f_x	=IF(Y1222=1,Z1221,IF(Z1221>1,Z1221-1,IF(W1222=1,IF(X1222="1",B26,IF(X1222="2",C26,IF(X1222="3",D26,0))),0)))				
	W	X	Y	Z	AA	AB	AC	AD
31								
32				**Beds**				
33			03				04	
34	In Use	Triage	Waiting for MD	Time Left	In Use	Triage	Waiting for MD	Time Left
1217	1	3	1	175	1	3	0	69
1218	1	3	1	175	1	3	0	68
1219	1	3	1	175	1	3	0	67
1220	1	3	1	175	1	3	0	66
1221	1	3	1	175	1	3	0	65
1222	1	3	0	174	1	3	0	64
1223	1	3	0	173	1	3	0	63

Figure 7. Excel simulation ED beds representation.

45

value of "Time Left" from the previous row to help determine whether the bed is being occupied by a new patient. If the value of "In Use" for the current row (cell W1222) equals 1 and "Time Left" from the previous row (cell Z1221) equals either 1 or 0, then we set the value for "Time Left" in the current row (cell Z1222) to the total time required (from B26, C26, or D26). Case 2 occurs when the value of "Waiting for MD" for the current row (cell Y1222) equals 1. Case 3 occurs whenever Case 2 does not hold and the value of "Time Left" from the previous row is greater than 1. When neither case 1, 2, or 3 holds, the bed is empty, and case 4 holds.

The model includes a set of columns for doctors which were created using the same methodology used for beds. Although the waiting area contains a queue of patients waiting for a bed, a column with a list of the beds that have patients who are waiting for a doctor is also modeled. Additional columns are included to list doctor availability and to list any doctors ready to be allocated to beds in the given minute of a particular row. For each doctor, we also keep a set of columns to track if the doctor is busy with a patient, the bed number currently being served (if applicable), and the number of remaining minutes that the doctor will be occupied with a patient.

The final columns are "Admission & Boarding" and "Discharge." Patients in the former are admitted to transfer to an in-patient ward which we model in an additional queue. We keep a set of columns to track in-patient bed occupancy as we did for ED beds. Boarding time is estimated with the following formula[10] ROUND(-LN(RAND()) × Mean In-Patient Bed Processing Time, 0). A separate column is used for patient discharges.

A final set of columns is used to track patient exits from the ED in the aggregate and for each type. This is used to produce the output performance statistics such as average LOS, calculated by adding Service Time and Average Wait Time which includes the wait time for the ED bed, doctors, and in-patient bed. We keep separate columns to track the number of patients waiting for each resource for each ESI type.

Model Validation

We compare functionality and performance of the spreadsheet model by testing if its resulting LOS equals that from MedModel as reported in Figure 1. To do so, we use an equivalence test[11] to see if the resulting LOS from the spreadsheet simulation is within x percent of the LOS from MedModel. Our null hypothesis is H_0: $|\mu_s - \mu_m| > x\mu_M$ where μ_S is the average LOS obtained from the spreadsheet model, μ_M is the average LOS from MedModel, x is the percentage, and the vertical bars represent the absolute value function. Rejection of the null implies that the spreadsheet model LOS is within x% of the MedModel results. Table 1 provides the P values for x ranging from 2% to 20%.

RESULTS

In this section, we focus on a specific ED setting[4] and validate the spreadsheet simulation model against the MedModel results for that setting. There are two potential sources of discrepancies between the Excel and MedModel

output results: the "per minute" structure of the former versus the "per patient" structure of the latter, and the distinct pseudorandom number generation processes. Figure 1 shows the 95% confidence limits of the simulated average LOS values obtained using the Excel simulations. It also shows the MedModel averages given a main ED with 23 beds and in-patient exit rates that vary from one patient per 10 minutes to one per 25 minutes. For the seven values of exit rates reported, the confidence interval covers the MedModel output.

The top panel in Table 1 provides the numerical values used in Figure 1. The bottom panel contains the P values for the equivalence tests given a percent difference (row) and the in-patient exit rate (column). The null hypothesis, for a given value of x (the percent difference) is H_0: $|\mu_s - \mu_m| > x\mu_M$ where μ_S is the average LOS from Excel and μ_M from MedModel. For example, when $x = 10\%$, we reject the null and conclude with 95% confidence that the Excel and MedModel LOS are within 10% of one another for each in-patient exit rate, except the "one per 25" as that P value is 0.247.

A closer look at Table 1 also shows that for a given exit rate (column), all P values decrease as the percent difference increases. For the base case of the original study where the exit rate was one per 20 minutes, we can conclude with 95% confidence that the Excel and MedModel LOS are within 4% of each other (P value = 0.012) but cannot conclude that they are within 2% (P value = 0.189). Figure 3 breaks down the LOS averages per ESI category for the base case. We observe differences across ESI categories with the largest being 15% for ESI = 1 and the smallest being 3% for ESI = 4, 5.

We also note that when the exit rate is at its slowest value (one per 25 minutes), we cannot conclude equivalence within anything less than 18%. This is a consequence of the high variability induced by congestion: the standard deviation of the Excel LOS is 32% of its average (131/409) whereas it is 25% (96/388) for the MedModel LOS despite both averages only differing by 5% of one another (see first row of top panel). This results in high P values that prevent us from concluding equivalence (or lack thereof) with any reasonable significance.

DISCUSSION

A discrete-event computer simulation model is a theoretical tool. How well it performs depends on how well the actual ED is modeled, the quality and precision of input, and the efficiency of the program itself at generating clear output and performance statistics. A number of platforms exist in the commercial market, but to implement and use those effectively, large investments of internal and external resources into the modeling, coding, and analytical activities are required. In this article, we show that one can use widely available spreadsheet programs such as Excel to conduct effective simulations (see Spreadsheet, Supplemental Digital Content 1, http://links.lww.com/SIH/A24, the simulation model that was used to generate the results for this analysis. Instructions are provided inside the file).

The key insight comes from understanding the difference in how Excel processes a simulation rather than a more traditional program such as MedModel. In Excel, the entire

system state is summarized in a *minute* sequence, where each row is based on its own random number results along with its preceding minute's values. Performance statistics are derived based on the time series of the states of the system at each minute. In MedModel, the focus is on the stay of a *patient* through the ED, from arrival to admission or discharge, and statistics are tallied and updated based on the performance of the ED resources for that patient.

As there is generally a spreadsheet program included in any office suite, one major advantage of using a spreadsheet for simulations is that there is no need to purchase, install, and learn new software, nor train users in manipulating and interpreting its output. A second advantage lies with spreadsheets being the norm for data storage and analysis. Building and managing spreadsheet simulations also allow for direct referencing of input and output data. This minimizes the burden of copying, the likelihood of transcription errors, and the risk of information leakage while allowing the user to leverage the functionality of spreadsheets by customizing analysis, statistical and performance measures and charts.

MedModel and the other traditional programs will exhibit better functionality than Excel, however, in modeling "exceptional" patients. For example, patients who LWBS are those who checked in to the ED but left before being placed into a bed. This is coded in MedModel by associating a random variable to each patient which will flag it for LWBS based on the time the patient has waited for an ED bed to become available. In Excel, modeling such patients is significantly more complex and time consuming because it consists of embedding additional "if()" statements to all cell formulas in the spreadsheet, switching their value to zero if the condition is satisfied, and ensuring that the time spent by those patients is properly tallied. We did not deem it worthwhile to model LWBS in the spreadsheet as the low rate in the original MedModel study (3%, or 6 per 24-hour cycles) had no impact on the overall performance.

Running time was also slower in Excel than MedModel. It took 3 minutes in the latter to generate one data point (which is the average over 40 replications) while it took 8 minutes to generate a similar value in Excel. Although processor capacity is the key determinant of MedModel performance (a large number of computations are done concurrently), it is the RAM memory in Excel that dictates the running time (one replication produces 21,600 rows and 520 columns). Thus, the Excel simulation must keep in active memory over 11M values and formulas. Adding RAM would be the most effective solution to cutting the running time down to MedModel levels.

REFERENCES

1. Connelly L, Bair A. Discrete event simulation of emergency department activity: a platform for system-level operations research. *Acad Emerg Med* 2004;11:1177–1185.

2. Hung G, Whitehouse S, O'Neill C, et al. Computer modeling of patient flow in a pediatric emergency department using discrete event simulation. *Pediatr Emerg Care* 2007;23:5–10.

3. Saunders C, Makens P, Leblanc L. Modeling emergency department operations using advanced computer simulation systems. *Ann Emerg Med* 1989;18:134–140.

4. Khare RK, Powell ES, Reinhardt G, Lucenti M. Adding more beds to the emergency department or reducing admitted patient boarding times: which has a more significant influence on emergency department congestion? *Ann Emerg Med* 2009;53:575–585.

5. Pennathur PR, Cao D, Sui Z, et al. Development of a simulation environment to study emergency department information technology. *Simul Healthc* 2010;5:103–111.

6. Anupindi R, Chopra S, Deshmukh SD, Van Mieghem JA, Zemel E. *Managing Business Process Flows*. 2nd ed. New Jersey: Prentice-Hall; 2006:41–73.

7. Swain JJ. The 2009 simulation software survey. *OR/MS Today* 2009;36:5.

8. Gilboy N, Tanabe P, Travers DA, Rosenau AM, Eitel DR. *Emergency Severity Index, Version 4: Implementation Handbook*. AHRQ Publication No. 05-0046-2. Maryland: AHRQ; 2005.

9. Bullard MJ, Unger B, Spence J, Grafstein E. Revisions to the Canadian Emergency Department Triage and Acuity Scale (CTAS) adult guidelines. *CJEM* 2008;10:136–142.

10. Myerson RB. *Probability Models for Economic Decisions*. California: Duxbury Press; 2004:107–150.

11. Barker LE, Luman ET, McCauley MM, Chu SY. Assessing equivalence: an alternative to the use of difference tests for measuring disparities in vaccination coverage. *Am J Epidemiol* 2002;156:1056–1061.