Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

MODELING A HOSPITAL MAIN CAFETERIA

William A. Stout, Jr.

Office of Organizational Effectiveness
The Johns Hopkins Hospital
Baltimore, Maryland 21287, U.S.A.

ABSTRACT

Taylor II version 2.0 is a PC based, menu-driven, discrete event simulation modeling software package featuring 2D/3D animation; a user programmable interface--Taylor Language Interface (TLI); graphics and paintbox (drawing) capabilities; distribution curve-fitting; and context-sensitive help. Add-on modules include advanced statistics, a run-time development kit, and shaded animation.

When learning a new simulation language, it is helpful to have a frame of reference that delineates at least the minimum programming elements required to formulate a successful model. Unfortunately, this is not always the case. Often, examples in the documentation are presented out of the context of a completed model leaving it up to the simulation practitioner to assemble the various pieces. Concepts presented in sample programs included with the package may be difficult to follow and to place in the context of your specific program development effort. The purpose of this paper is to provide the new user with the basics of Taylor II by building a "hard-copy" application step by step.

We illustrate the importance of the relationships between various individually presented menus to emphasize the linkage between menu choices and the construction of user-written functions. By providing an overview of the total application development cycle, the user gains perspective on how individual parts are integrated into a working model.

1 THE APPLICATION TO BE MODELED

The process to be modeled is a hospital main cafeteria serving food to both staff and visitors. The cafeteria opens at 6 a.m. and closes at 6 p.m. Monday through Friday. Weekends and holidays are not included in the model. There are seven cashier stations which are staffed based on perceived customer flow requirements and availability of cashiers. The time to "ring-up" a sale (service time) varies according to time of day--breakfast transactions have a shorter service time than lunch transactions because the menu selection is more limited. Customer inter-arrival rates (1 customer every x number of seconds) also vary by time of day; lunch is the busiest time of the day and therefore is the time when most cashiers are needed. Historically, lunch times have been characterized as chaotic.

The time of arrival at the cafeteria food service area and the time involved in the choice of individual food items is not modeled, although both could become parts of a more inclusive model of the entire process. Availability of food at point of service could also be modeled since delays here influence the arrival time at the checkout lines. This model will probably be expanded to address the above two issues.

Inter-arrival times in the model are based on the time the customer arrives at the checkout line. If the cashier is idle, the customer receives service immediately; if the cashier is busy, the customer joins the queue for the cashier selected. No provision is made for reneging or jockeying between lines, although this could be included. It is assumed that customers naturally gravitate to the shortest line. As stated above, the service time distributions depend on the time of day.

Early model development effort should be focused on the layout and the process routing requirements. While it is crucial to the accuracy of the final model, the absence of actual arrival rate/service time distributions data should not hinder initial model development. Both the arrival and service time distributions are estimates sufficient to allow the model development phase to proceed. Actual arrival/service times can be observed, summarized, and the appropriate distributions identified and inserted independent of model development.

2 THE MODEL OVERVIEW

Figure 1 depicts the layout of the seven cashier lines in the main cafeteria labeled "Aux 1","2","3","4","5","6",

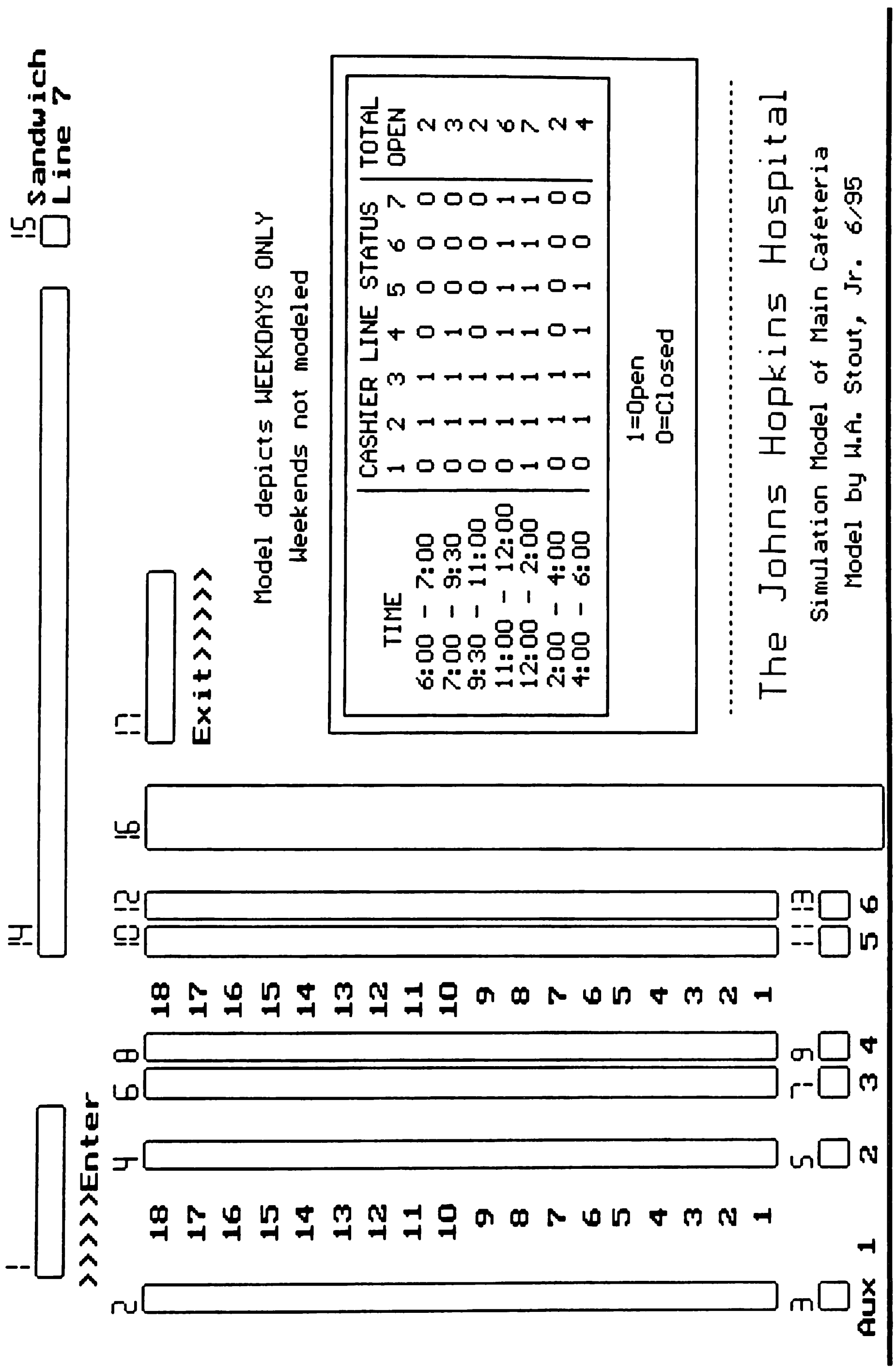

Figure 1: Taylor II Screen Layout of Main Cafeteria

and "Sandwich Line 7". The box labeled "Cashier Line Status" indicates which cashiers are open/closed as well as the total number of cashiers open by time of day. Rectangles depicting the queues and the cashier locations are generated during the layout phase of model construction by choosing "Layout" from the menu options. The line status box and other text are generated in the paintbox feature included in the package. The columns of vertical numbers from 1 to 18 represent positions in the queues and were added as text to assist the observer in visually identifying the number of customers in the waiting line as the model runs. The text added in the paintbox is not required to run the model but is included to provide information and clarification to the viewer.

It is assumed that your system meets the minimum requirements to run Taylor II and that the software has been properly loaded onto your hard drive in a directory named "taylor" or a name of your choice. Minimum system requirements are a 80386 (80486 is recommended) processor, 2mb RAM internal memory, mouse, and a VGA graphics adapter. An inkjet or laser printer is recommended. Also remember that Taylor II is DOS-based and must be executed from the "C" prompt with Microsoft Windows inactive. If one wants to run Taylor II in the windows environment, one needs to use the taylor.pif file included with the software.

Once a simulation opportunity has been identified and an assumptions document written listing the problem; initial assumptions; process layout; data on process times and distributions (if available); and jointly defined client/provider expectation and outcome requirements, the steps in setting-up the simulation model can proceed as follows:

1. Create the layout and routing.
2. Customize the layout and routing to fit the model.
3. Determine and enter time representation.
4. Enter the arrival rate and service time distributions.
5. Enter any special handling or routing requirements.
6. Enter any user-written functions required.
7. Add aesthetic detail using the paintbox feature.

The remainder of this paper presents a discussion of these seven steps in modeling the cafeteria example. Examples of Taylor II screens designed for program selection options and user data input are shown. Some screens are not shown due to space considerations but may be discussed in the text.

3 CREATING THE TAYLOR II MODEL DESIGN

Step 1. Create the layout and routing. Taylor II element types include machine, transport, buffer, aid, conveyor, path, reservoir, warehouse, and inout. The screen layout is approximated using the Taylor II elements. In the cafeteria model we will use inout, buffer, and machine elements. From the main menu click on "Visuals", "Settings" to activate the visualization settings menu. This menu contains thirteen settings so cursor down to the "Display numbers" box and set it to "On". This will cause the individual elements in the layout to be numbered. Press the escape key to clear the pop-up table. Click on "Layout" to begin and notice that the displayed default element is numbered. The first element is shown in the upper, left corner of the screen. Notice that the element on the screen has a cross-hair in the upper, left corner which identifies it as the active element. Use the space bar to toggle through the types of elements or type the first letter of the element you want, in this case, the inout element. A new element is created and becomes active each time the enter key is pressed. Repeat the toggle or first letter method of identifying the element and when all elements have been defined press the escape key to return to the sub-menu. To switch active elements simply click on the element you desire.

Once the elements have been identified, the programmer must specify the routing. Click on the "Routing" button. The element where routing begins is identified by a larger cross-hair than the one in the "Layout" section. Press the enter key to establish the routing from element to element. When routing is completed, press the escape key to return to the sub-menu. This routing establishes the initial linkage between elements only, and should not be tailored to fit actual model requirements at this time. Customizing the routing to fit the actual model requirements is discussed in step 2.

Step 2. Customize the layout and routing to fit the model. Elements may be repositioned and/or enlarged or contracted to fit layout requirements. The large block at the bottom, middle of the screen provides key-use instructions and a list of the available element types. Escape to the main menu and click on the first element you have defined. A menu titled "What to edit?" appears. Choose the "Stages" button to specialize the routing. A pop-up menu showing routing parameters appears and allows you to specify where the flow is received from or sent to. Another way to do this is to select "Detail", "Routing" from the main menu. The pop-up editing window can be moved from element to element with the left and right arrow keys.

The number of the element in the crosshairs is shown at the top of the box. For instance, the top line of the example shows element 1, job 1 as the selection with present routing, "Send to", to element 2. It is not necessary to always send the entity to the next sequential element and here is where the desired routing is identified. Once the job to be used is designated, cursor down to the "Send to" or "Receive from" lines (the active line changes color from grey to blue). If you want to modify only a portion of the present entry, press the enter key, and use the left cursor key to the position on the line where the new data is to be placed, make the updated entry, or use the delete key to remove unwanted characters. This is useful if the selection criteria is a statement ("select 1 from 2,3,4) rather than a single number. The enter key will cause the blue highlighted active line to change to yellow, ready for editing. The backspace key will clear the entry completely allowing new routing to be entered. The escape key will clear the edit mode (yellow) and return you to the active line (blue).

Step 3. Determine and enter time representation. It is important that time representation be coordinated in four separate but related areas--the time bar (F8); the clock (F10); the time settings menu; and in any user-written functions in TLI. The time units chosen on these menus must be compatible and whether they are or not will show up on the system clock when the model is run. The following three tables; Tables 1, 2, and 3; depict a compatible time representation.

The clock (F10) definitions only affect the clock display on the screen, and do not have any effect on the model. The time units defined in each of these tables must be compatible within the tables and within any user-written functions. For example, the time to start the clock (6 a.m. equals 360 minutes from midnight) and the total run time (720 minutes divided by 60 minutes equals 12 hours) are controlled in the "Clockstart" and "Clockcycle" sections of Table 2. The number of hours in the shift (day) is controlled by line 3 of Table 1.

Time entries that are a part of user-written TLI functions must be in synchronization with the "Units per hour" entry in Table 2. Table 4 shows an example of the user-written function for service times contained in the cafeteria model. Notice that the time entries are in seconds, the "Units per hour" measure established in Table 2. As calculated, 43200 seconds equals 12 hours. Each "if" statement establishes a time period during which a particular lognormal distribution is active. These distributions are more or less intense depending on the time of the day or the complexity of the menu--breakfast, lunch, and dinner are the busiest times of the day; breakfast has the least complex menu.

In summary, every reference to time in the model (job times, TLI functions, user-defined events, etc.) must be defined in terms of "units" as predefined in the time bar (F8) representation. Taylor II works in "units" and the user gets to define what a "unit" of time equals.

Step 4. Enter the distributions for arrival rates and service times. Arbitrary distributions were chosen for initial model development and entered in

Table 1: The Time Bar (F8)

Time representation	
1	60 units make 1 minute
2	60 minutes make 1 hour
3	12 hours make 1 day
4	5 days make 1 week
Mon Tue Wed Thu Fri	
Time representation	

Table 2: The Clock (F10)

Units per hour	3600
Hours	On
Minutes	On
Seconds	Off
Clockstart(minutes)	360
Clockcycle(minutes)	720
Delay	0
Continuous	Off
Clock settings	

Table 3: Go, Time Menu Selection

	Timenumber	Timeunit
Length of simulation	1	<week>
Stop every	1	<week>
Screen refresh every	1	<units>
Time settings		

TLI using the text editor shown in Table 4. Distributions were entered for arrival rates and service times. Arrivals occur according to a non-homogeneous Poisson process while the service time distributions are lognormal. For instance, there are three separate arrival patterns which describe customer load for breakfast, lunch, and dinner. A less intense arrival pattern is used for the time periods between breakfast, lunch, and dinner. These differences are apparent when the model is run. Also, once the basic model is written, the actual distribution data can be collected by observation and included in the model. The modulo function (|) was used to define a repetitive time period each day.

Step 5. Enter any special handling or routing requirements. Special handling requirements include such items as controlling the display of icons to represent specific phases of the model. In this model I changed the appearance of the icons representing breakfast, lunch, and dinner while using the default icon for all times in between. This approach enhances the visual message of the model and promotes client buy-in when they recognize their process on the screen. Changing icons involves the "Product parameters" menu and a written TLI user function (see Table 4) which relates the time periods during which specific icons are active.

Another special handling feature is the ability to specify when a particular server is open for service. By opening and closing servers, the demand for service can be matched more closely with the supply of customers entering the system. A typical statement to open a server provides the time to open and which server to open. The statement for the event is "elaccept[3]=1" which opens server number 3 at the time specified. The statement "elaccept[3]=0" turns the server off. The "Event list" menu provides this feature. The "Event list" is also useful for providing a beeping sound to signal changes in the model you want to emphasize to the client.

Here is an example of special routing used to be more specific in a "Send to" instruction from the "Stages" menu. Instead of entering a single number to route to the next element you could enter the statement "select 1 from 2,4,6,8,10,12,14 order -elqueue[L]". This instruction says to select 1 path from a group of eligible paths by choosing the path with the shortest queue.

Step 6. Enter any user-written functions required. The format of user-written functions has already been shown in Table 4. The first function, "service", shows service time distributions. The second function statement equates the user-supplied name "ikon" to p1 then uses p1: as a handle to equate the icon symbol number in the "if" statement to the icon number in the product parameters menu. In this model product 1 is icon number 28, therefore, the symbol displayed is the symbol for icon 28, the number "1" enclosed in a yellow square.

Step 7. Add aesthetic detail using the paintbox feature. The screen visuals need not be limited to the boxes generated during the create and layout phase. By clicking on the paintbox icon located in the lower, left corner of the main screen, drawing and text may be added to the basic layout. Text can be added, sized, color selected, repositioned, cut and pasted (copied), and erased by selecting appropriate buttons in the paintbox. Boxes and shapes may also be added. Figure 1 shows an example of some of these features. The numbers from 1 to 18 were included to assist the viewer in quickly assessing the number of customers in line at any point in time. Each cafeteria line is labeled. The content of the table showing cashier line status was first defined then enclosed in the boxes. Various text styles and sizes are also shown. The paintbox drawing screen is saved as background in a file titled "filename".drw which is associated with the main program file titled "filename".sim.

Table 4: TLI Editor

```
function service=p1

        if time|43200>=0 and time|43200<3600 then p1:=lognormal[30,10]
        if time|43200>=3600 and time|43200<12600 then p1:=lognormal[40,5]
        if...

function ikon=p1

        if time|43200>=0 and time|43200<3600 then p1:=4
        if time|43200>=33600 and time|43200<12600 then p1:=1
        if...
```

The Taylor Texteditor functions

4 CONCLUSION

The introduction of animation to simulation has provided a tool that is beneficial to both programmer and client. Animation assists the programmer in understanding and debugging simulation models. Clients gain the advantage of earlier buy-in to the accuracy of the simulated representation of their process. Taylor II provides the platform to address these issues.

Hopefully, this paper removes some of the mystery of the "nuts and bolts" aspects of Taylor II. It speaks to some of the detail in the context of a complete model rather than from fragmented pieces of information in the documentation. There are many features of Taylor II not mentioned herein, but perhaps, with this as a guideline, it will be easier to define the basic model and thereby have more time to investigate its many other features.

ACKNOWLEDGEMENTS

The author thanks Cliff King of F&H Simulations, Inc. for his assistance in reviewing and suggesting improvements to this document.

REFERENCES

F&H Logistics and Automation B.V. 1993. *Taylor II User's Guide*. Spoorlan 424,5038 C G Tilburg, The Netherlands.

F&H Logistics and Automation B.V. 1993. *Taylor II TLI*. Spoorlan 424,5038 C G Tilburg, The Netherlands.

F&H Logistics and Sutomation B.V. 1993. *Taylor II Training Manual Simulation Software*. Spoorlan 424,5038 C G Tilburg, The Netherlands.

AUTHOR BIOGRAPHY

WILLIAM A. STOUT, JR. is a Management Analyst at the Johns Hopkins Hospital. He received a B.S. in business management and an M.B.A. from the University of Baltimore in 1970 and 1977 respectively. His interests lie in application of simulation techniques in management decision-making.