cases where other mechanisms are necessary. Specific types of data, such as time-varying arrival patterns, require a unique data representation. In other situations, the volume of data is large enough that it's necessary to represent the data in a more convenient form, and in fact even import the data from an external source. And in situations where the analyst using the model may not be the same as the modeler who builds the model, it may be necessary to consolidate the data rather than having them scattered around the model. In

7.1 Data Tables

To add columns to a table you select a table to make it active and then click on property types under `Standard Property`, `Element Reference`, `Object Reference`, or `Foreign Key`. A table column is typically represented by the Standard Properties illustrated in Table 7.1. Use an Object Reference when you want a table to reference an object instance or list of objects such as an `Entity`, `Node`, `Transporter`, or other model object. Likewise use an Element Reference if you want a table to reference a specific element like a TallyStatistic or Material.

7.1.2 Model 7-1: An ED Using a Data Table

Emergency Department

Table 7.2: Model 7-1 ED Basic patient data.

Patient Type	Priority	Treatment Time (Minutes)
Routine	1	Random.Triangular(3,5,10)
Moderate	2	Random.Triangular(4,8,25)
Severe	3	Random.Triangular(10,15,30)
Urgent	4	Random.Triangular(15,25,40)

There are four types of patients. The type of a patient arriving is random. That means a patient can be routine, moderate, severe, and urgent. Moreover, the specific type of patient arrives with some probability. If a patient is the type of routine, then the treatment time follow s a specific distribution. That means we need change the distribution of treatment time based on the type of patient arriving.
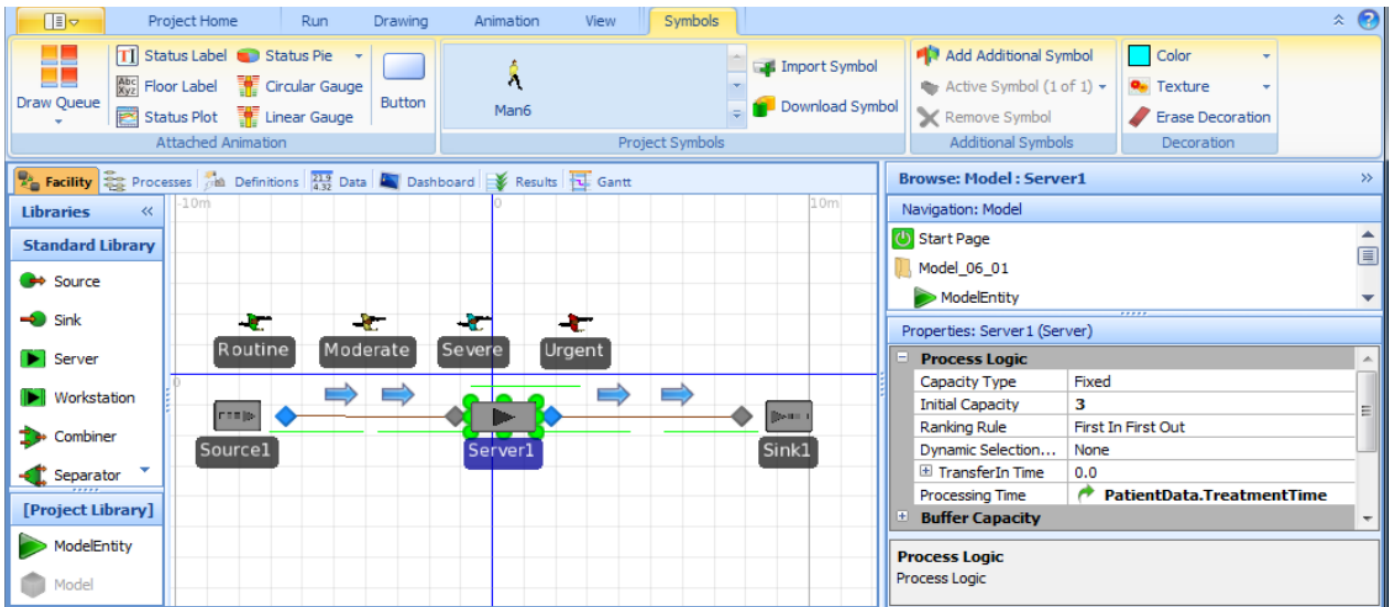
	Patient Type	Priority	Treatment Time (Minutes)	Patient Mix
>	Routine	1	Random.Triangular(3,5,10)	40
	Moderate	2	Random.Triangular(4,8,25)	31
	Severe	3	Random.Triangular(10,15,30)	24
	Urgent	4	Random.Triangular(15,25,40)	5

Panels «   **PatientData**

Tables

$f(x)$

Figure 7.3: Model 7-1 ED Enhanced patient data in Simio table.

Properties: Source 1 (Source)

Arrival Logic
Entity Type	→ PatientData.PatientType
Arrival Mode	Interarrival Time
⊞ Time Offset	0.0
⊞ Interarrival Time	**Random.Exponential(4)**
Entities Per Arrival	1

⊞ **Stopping Conditions**

Table Reference Assignment
  ⊟ Before Creating Entities
Table Name	**PatientData**
Row Number	**PatientData.PatientMix.RandomRow** ▼

  ⊞ On Created Entity

⊞ **State Assignments**

Row Number
The one-based row index into the table.

Figure 7.4: Selecting an entity type from a table.

2. Now we can continue building our model. The last change we made allows us to have a single Source that will produce the specified distribution of our four patient types. Place a Source in your model and specify an Interarrival Time of `Random.Exponential(4)` and units of `Minutes`. Instead of specifying one patient type in the Entity Type property with a specific row number as we did in Figure 7.2, we'll let Simio pick the row number and then we'll select the Entity Type based on the `PatientType` specified in that row. We must select the table row before we create the entity; otherwise the entity would already be created by the time we decide what type it is. So in the Table Reference Assignment, Before Creating Entities category, we'll specify the Table Name of `PatientData` and the Row Number of `PatientData.PatientMix.RandomRow`. After the row's selected, the Source will go to the Entity Type property to determine which Entity Type to create. There we can select `PatientData.PatientType` from the pull-down list. This is illustrated in Figure 7.4.

Figure 7.5: Model 7-1 Completed ED model.
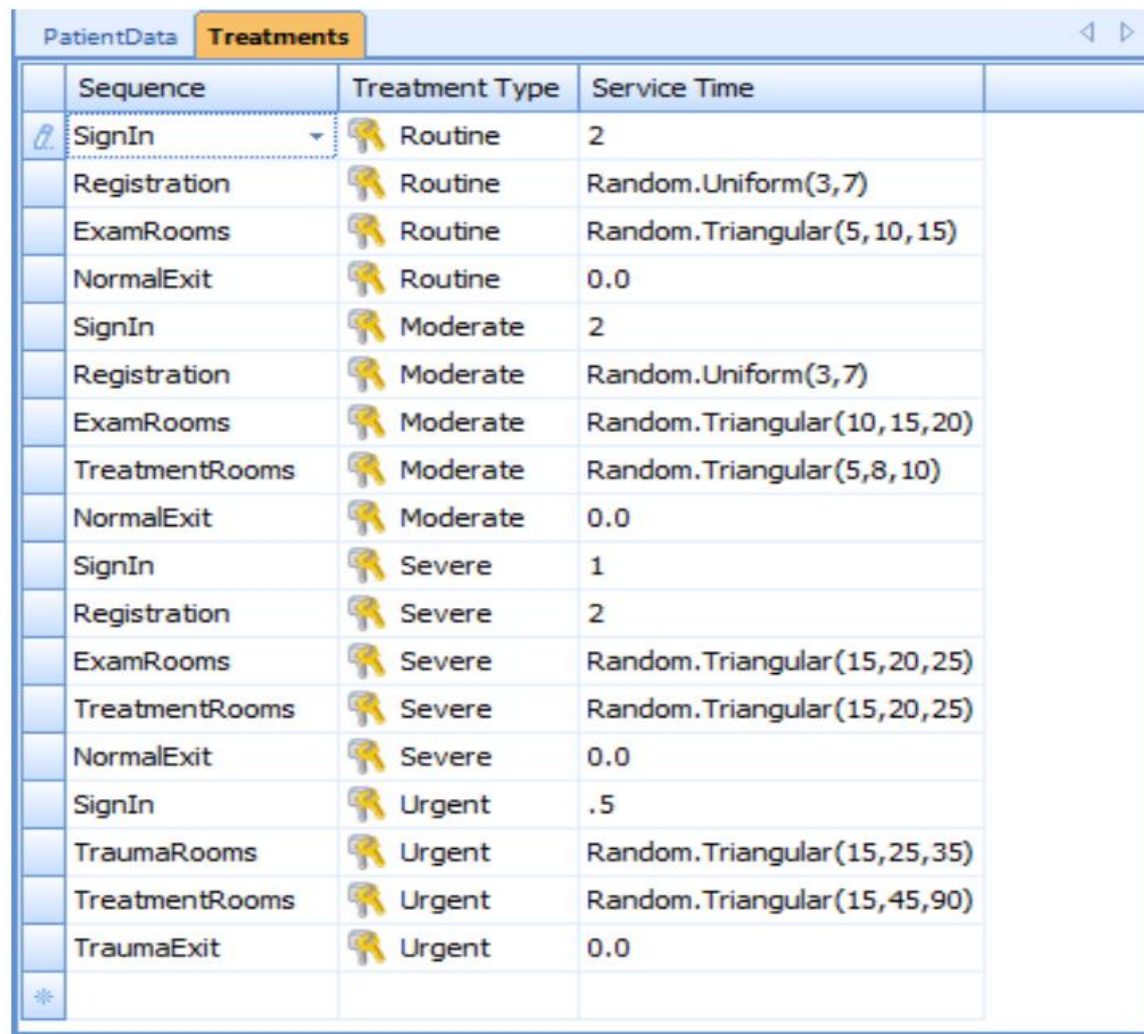
Verify Model

7.1.3 Sequence Table

1. A Sequence Table is a special type of data table used to specify a sequence of destinations for an entity.
2. Add Sequence Table button will create a table and automatically add a column named Sequence for specifying a routing sequence for an entity.
3. Each simple sequence table defines one routing plan (e.g., a list of node visitations).
4. Each row in the sequence table corresponds to a specific location.
5. Other data in row is usually used for location-specific properties, for example the processing time, priority, or other properties required at a particular location.

Simple Sequence Table

Relational Sequence Tables

1. Combine several different sequences into a single sequence table

7.1.4 Model 7-2: Enhanced ED Using Sequence Tables

Sequence	Treatment Type	Service Time
SignIn	Routine	2
Registration	Routine	Random.Uniform(3,7)
ExamRooms	Routine	Random.Triangular(5,10,15)
NormalExit	Routine	0.0
SignIn	Moderate	2
Registration	Moderate	Random.Uniform(3,7)
ExamRooms	Moderate	Random.Triangular(10,15,20)
TreatmentRooms	Moderate	Random.Triangular(5,8,10)
NormalExit	Moderate	0.0
SignIn	Severe	1
Registration	Severe	2
ExamRooms	Severe	Random.Triangular(15,20,25)
TreatmentRooms	Severe	Random.Triangular(15,20,25)
NormalExit	Severe	0.0
SignIn	Urgent	.5
TraumaRooms	Urgent	Random.Triangular(15,25,35)
TreatmentRooms	Urgent	Random.Triangular(15,45,90)
TraumaExit	Urgent	0.0

Figure 7.6: Model 7-2 Relational sequence table to define Treatments.

3. Move to the Data Window and the Tables panel. We'll start by designating our existing `PatientType` column as a unique Primary Key so that our sequences table can reference this. Select the `PatientType` column and click on `Set Column as Key` in the Ribbon. At this point we can also

5. We need to add another column to identify the treatment type. Click on `Foreign Key`. This will create a column that uniquely identifies the specific treatment type. In our case the treatment type corresponds exactly to the `PatientType` in the `PatientData` table. Go to the properties of this new column and name it `TreatmentType`. Because it's a foreign reference, that means that it actually gets its values from somewhere else. The Table Key property specifies from where the Treatment Type value comes. Select `PatientData.PatientType` from the pull down list.

10. Recall from the general discussion above that we must specify on each outbound node if the departure is to follow a sequence. Click on the blue TransferNode on the output side of the Source. In the routing Logic category change the Entity Destination Type to `By Sequence`. You could repeat this process for each outbound node, but Simio provides a shortcut. In our case all movements will be by sequence, so we can change them all at once. Click on any blue node, then Ctrl-click one at a time on every other blue node (you should have six total). Now that you've selected all six transfer nodes you can change the Entity Destination Type of all six nodes at once.

13. Some patients are more important than others. No, we're not talking about politicians and sports stars, but rather patient severity. For example we don't want to spend time treating a chest cold when another patient may be suffering from a severe heart attack. We already have patient priority information in our patient table, but we need to change the server-selection priority from its default First In First Out (FIFO). For each server you need to change the Ranking Rule property to `Largest Value First`. That will expose a Ranking Expression property that should point to our table: `PatientData.Priority`. This will guarantee that an `Urgent` patient (priority 4) will be treated ahead of all other patients (priorities 1-3).
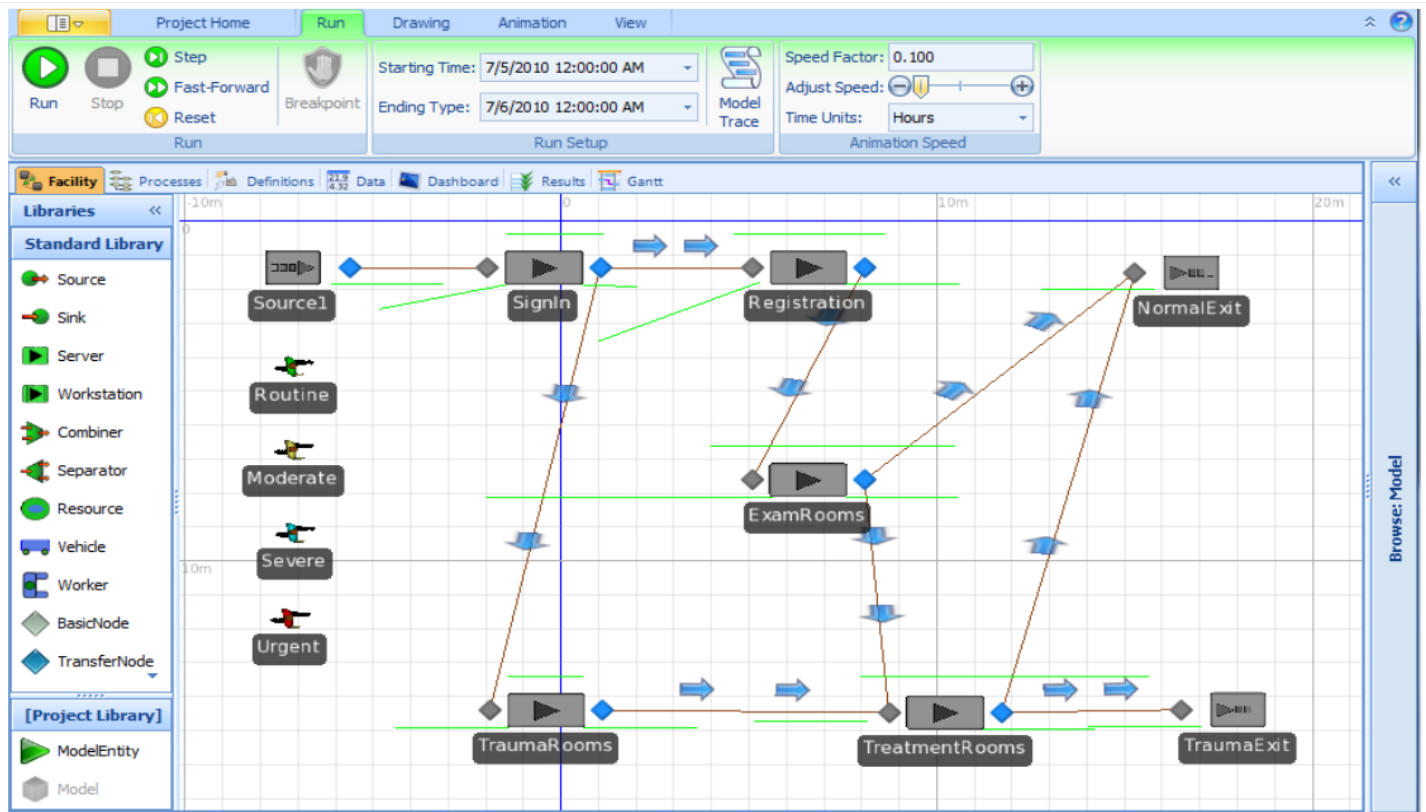
Figure 7.7: Model 7-2 completed ED model with sequences.

7.1.5 Arrival Tables and Modes 7.3

Arrival tables are a type of data tables that generate specific arrivals

When you use the Arrival Table features of the Timer it will process each row in the table at the correctly specified time and, when triggering a process, the token in that process will have the specified row associated with it.
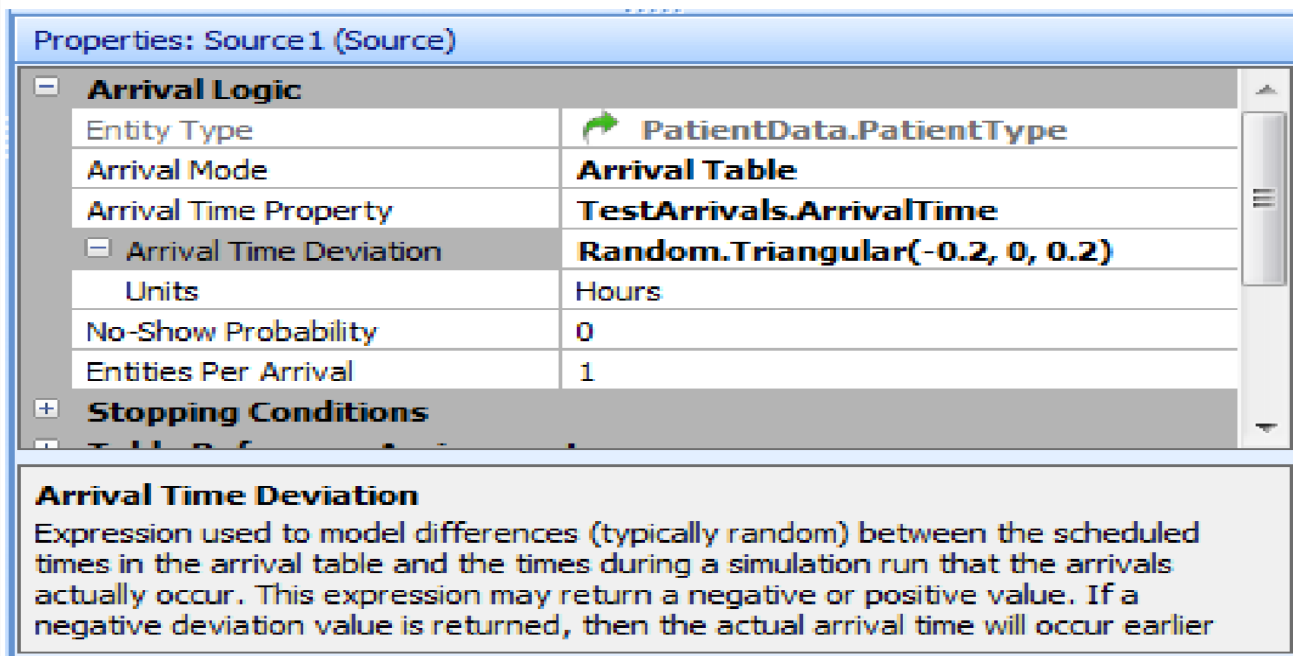

Figure 7.9: Sample arrival table.

Figure 7.10: Using Arrival Tables with the Source object.

5. To trigger use of this table we must specify it on our Source. For the Arrival Mode property select `Arrival Table`. The Arrival Time Property is where you specify the table and property name that will determine the arrivals — select `TestArrivals.ArrivalTime` as shown in Figure 7.10.

6. Also on the Source we will specify an Arrival Time Deviation of `Random.Triangular(-0.2,0,0.2)` to support random arrival times. This indicates that each individual patient may arrive up to 0.2 hour early or up to 0.2 hour late (i.e., within 12 minutes of the scheduled time).

7.2.1 Calendar work Schedules