

Sistem Benzetimi
Arena
Laboratuar Notları ©



Murat M. Günal

Deniz Harp Okulu
Endüstri Mühendisliği Bölümü

Ağustos - 2012

İçindekiler

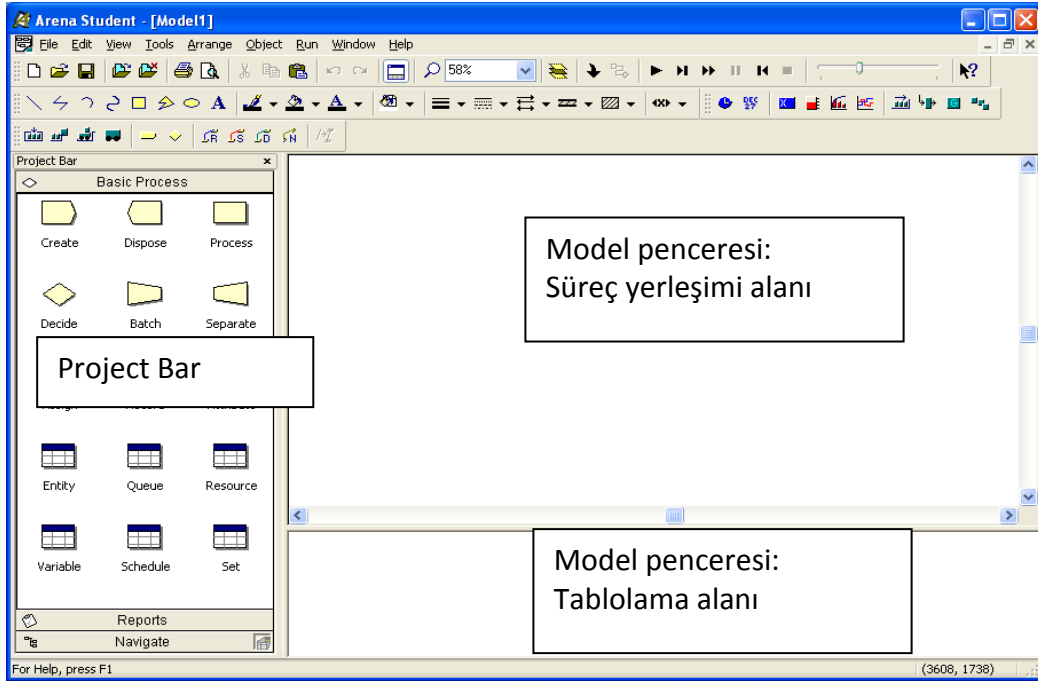
1	Arena Genel Tanıtım	3
2	Basit Kuyruk Modeli	5
2.1	Tanım.....	5
2.2	Genel Görünüm (Süreç Yerleşimi).....	5
2.3	"Create" Modülü.....	5
2.4	"Process" Modülü.....	6
2.5	"Dispose" Modülü.....	7
2.6	Modelin ilk defa çalıştırılması.....	8
2.7	"Resource" Modülü.....	9
2.8	"Entity" Modülü.....	9
2.9	Model çıktısının grafik ile gösterimi.....	10
2.10	Model çalışma süresinin belirtilmesi.....	12
3	Acil Servis Modeli	14
3.1	Tanım.....	14
3.2	Genel Görünüm (Süreç Yerleşimi).....	16
3.3	"Assign" Modülü.....	17
3.4	"Process" Modülü.....	18
3.5	"Expression" Modülü.....	20
3.6	"Decide" Modülü.....	21
3.7	"Record" Modülü.....	23
3.8	Modelin çalıştırılması ve deneyler.....	24
4	Boğaz Geçiş Modeli	26
4.1	Tanım.....	26
4.2	Model.....	27
4.2.1	Parametreler.....	27
4.3	Senaryolar.....	27
4.4	Hazırlık.....	27
4.5	Genel görünüm.....	28
4.6	Modül diyagramı.....	29
4.7	Variables ve Resources.....	29
4.8	Assign Modülleri.....	30
4.9	Station'lar.....	31
5	OTOYOL GIŞELERİ MODELİ*	33
5.1	Tanım.....	33
5.2	Geliş Üretimi.....	35

5.3 Arabaları Gişelere Gönderme.....	37
5.4 Arabalara Gişelerde Hizmet Vermek.....	41
5.5 Ücret Gişesi Modeli Benzetim Sonuçları.....	46
5.5.1 Temel Model.....	46
5.5.2 Geliştirilmiş Model.....	47

1 ARENA GENEL TANITIM

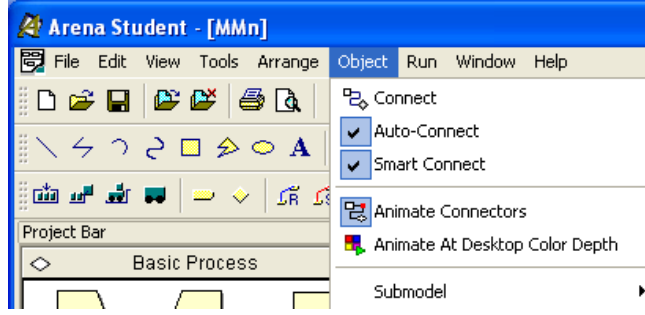
Arena yazılımı genel maksatlı bir benzetim yazılımıdır. Windows işletim sisteminde çalıştığından kullanıcı arayüzü diğer Windows programlarındaki gibi menüler, araç çubukları ve çalışma alanından oluşmaktadır.

Arena'yı ilk açtığınızda Şekil 1-1'e benzer bir ekran görüntüsü belirecektir. Sol taraftaki "Project Bar", modelinizi kurarken kullanacağınız öğeleri içermektedir. Buraya daha önceden tanımlanmış şablonlar (Template) eklenip çıkartılabilir. Örneğin Şekil 1-1'de "Basic Process" şablonunun modülleri görünmektedir. Ayrıca varsayılan olarak çıktığı raporlarını içeren "Reports" ve modelde gezinmeyi sağlayan "Navigate" alanları da bulunmaktadır. Mouse ile "sürükle bırak" yöntemiyle "module" adı verilen bu öğelerle, modelinizin süreç yerleşimini (process layout) tasarlayabilirsiniz. Modelinize dahil edilen bir öğenin üzerine tıkladığınızda o modüle ilişkin "Tablolama alanı"nda detaylar belirecektir.



Şekil 1-1 Arena çalıştırıldığında ekran görüntüsü

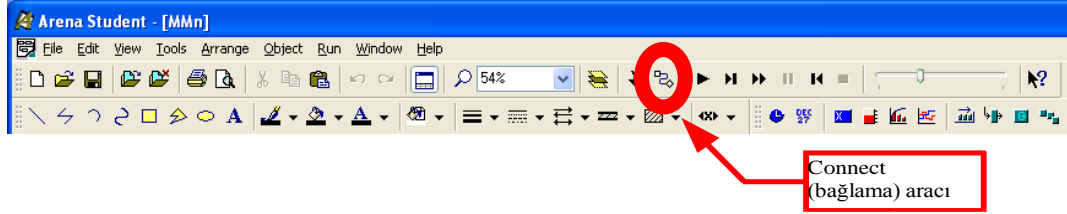
Modüllerin birbirine bağlanması, eğer Şekil 1-2'de görüldüğü üzere "Auto-Connect" ve "Smart-Connect" seçili ise otomatik olarak yapılacaktır. Bu bağlantılar, modülleri yerleştirme sırasına göre olmaktadır. Örneğin bir "Create" modülü seçip sürükleyip bırakmış ve daha sonra bir "Process" modülü sürükleyip bırakmışsanız bunların birbirine bağlanmış olması gerekir.



Şekil 1-2 Modüllerin otomatik bağlanmasının sağlanması

Eğer bu bağlantı sağlanmamışsa, veya yaratma sırasını takip etmemişseniz (mesela önce Process sonra Create sürükleyip bırakmışsanız) daha sonra elle de bağlantıyı sağlayabilirsiniz. Bunun için araç çubuğundan Şekil 1-3’de görülen “Connect” (bağlama) aracı seçilmelidir. Mouse’un şekli değişecek ve biryerden diğerine bağlanabileceği zaman yeşil ve kırmızı olarak renk değiştirecektir.

Şekil 1-3 Bağlama (Connect) aracı



Arena’ya ilişkin diğer konular takip eden bölümlerde yeri geldikçe açıklanacaktır.

2 BASİT KUYRUK MODELİ

2.1 Tanım

Bu bölümde bir kuyruk sisteminin modeli yaratılacaktır. Sistemimiz şu şekilde işlemektedir: Herhangi bir tipteki varlıklar sistemimize gelmekte, işlem görmekte, ve işi biten varlıklar sistemden ayrılmaktadır. Ancak işleme tabi tutulmaları için bir kaynağın (veya işlem yapan bir çalışanın veya makinenin) var olması gerektiğini ve eğer bu kaynaktan elimizde yoksa varlıkların kaynaklardan elimizde olana kadar bekleyeceklerini biliyoruz. Birden fazla varlık kaynak için bekliyorsa, önce gelenin önce işlem göreceğini kabul edebiliriz.

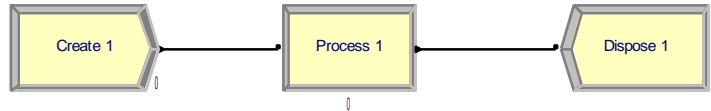
Bu sistemin kuyruk teorisi notasyonları ile M/M/n/FIFO olduğunu kabul edelim. Yani, varlıkların sisteme gelişleri ve sistemde işlem görmeleri "Poisson süreci" olup, sistemde "n" tane kaynak (sunumcu veya işlem yapan) bulunmaktadır. Kuyruk disiplini ise "ilk giren ilk çıkar"dır (First-In-First-Out (FIFO)). Varlıkların sisteme gelişleri arasındaki sürenin olasılık dağılımı ve işlem sürelerinin olasılık dağılımı Üssel (Exponential) dağılımdır. Üssel dağılımın tek parametresi olup bu parametre "ortalama"dır.

Sistemimizde gelişler arası sürenin ortalama 8 dakika olduğunu, her bir varlığın ortalama 10 dakika işlem gördüğünü, ve işlem yapan 2 sunumcunun olduğunu kabul edelim.

2.2 Genel Görünüm (Süreç Yerleşimi)

"Basic Process" panelinden "Create", "Process" ve "Dispose" modüllerini sürükleyip ana ekrana bırakın (Şekil 2-1). Otomatik olarak birbirlerine bağlanmaları gerekir, ancak bağlanmamışsa "Connect" aracını kullanarak da bağlayabilirsiniz.

Şekil 2-1 Kuyruk modeli süreç yerleşimi

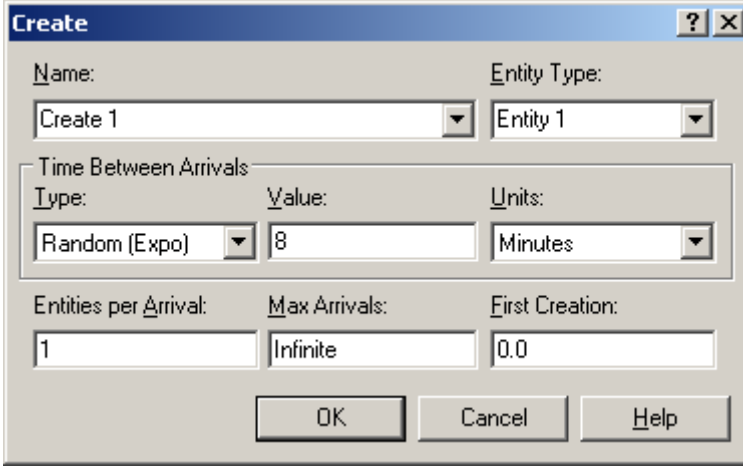


2.3 "Create" Modülü

"Create 1" modülüne çift tıkladığında bu modülle ilgili detayların girilebileceği bir form görünür (Şekil 2-2). "Name" alanına bu modüle sizin vereceğiniz bir isim yazılabilir. "Entity Type" ise sistemdeki varlıklarınızın, yine sizin belirleyeceğiniz, tipinin yazılabildiği bir alandır. Varsayılan olarak "Entity 1" yer almasına rağmen siz bunu, örneğin, *parça*, *hasta*, *araba* olarak değiştirebilirsiniz. Buraya yazılacak olan ifade, sisteminizde hareket

halindeki, ya da süreçler arasında gezinecek olan varlığın ne olduğunu söyleyecektir.

Şekil 2-2 Create modülü



“Time Between Arrivals” bölümünde yaratılan Entity’lerin hangi zaman aralıklarında yaratılacağına ilişkin ifadeler girilir. Burada “Type” olarak “Random (Expo)” girilirse gelişler arası zamanlar üssel dağılıma uygun olacak demektir. “Value” alanına ise bu dağılıma ilişkin parametre değeri girilir. Problemimizde tanımlandığı üzere gelişler arası süreler ortalaması 8 dk. üssel dağılım olduğundan bu alana 8 girilir. Ayrıca zaman birimi olarak “Units”den “minutes” seçilir.

En altta kalan üç alan sırasıyla; “Entities per Arrival” her geliş olayında kaç varlık geleceğini, “Max arrivals” bu Create modülünün en fazla kaç varlık yaratacağını, ve “First Creation” ise ilk varlığın kaçınıcı benzetim zamanında (yani 0.0 ise benzetim başlar başlamaz) yaratılmaya başlanacağı girilir.

2.4 “Process” Modülü

Process modülü entity’ler yaratıldıktan sonra işlemin yapılacağı yeri temsil eder. Şekil 2-3’de görüldüğü üzere ilk yaratıldığında “Name” alanında varsayılan isim olarak “Process 1” yer almaktadır. Bu alana istenilen bir isim girebilir. Girilen isim bu süreci temsil etmelidir. “Type” seçili alanında “Standart” yer alacaktır. Şimdilik bu alanda bir değişiklik yapmaya gerek yoktur.

“Logic” bölümü süreç ile ilgili asıl detayların girildiği alanları içermektedir ve Process ilk yaratıldığında Şekil 2-3’deki görüntüden farklıdır. Öncelikle bir kuyruk modeli inşaa ettiğimizden dolayı bu süreçten önce gelen varlıkların bir kaynak aramalarını söylemeliyiz. “Action” bölümünde “Seize Delay Release” seçildiğinde, gelen her varlığın önce bir “Resource”u, yani kaynağı, elde etmeye

çalışacağını, sonra gerekirse kaynak için bekleyeceğini, ve en sonunda da “Release” yani kaynağı serbest bırakacağımızı söylemiş oluruz. “Priority” kısmında bir değişiklik yapmayacağız ama “Resources” bölümünde öncelikle “Add...” düğmesine basmalıyız. Bu şekilde bir varlığın bu süreçte çalışması için hangi tipten ve kaç tane kaynağa ihtiyacı olacağını belirtmiş oluruz. Bizim örneğimizde “Resource 1” tipinden 1 adet kaynağa ihtiyacımız olacaktır.

Şekil 2-3 Process modülü

Varlıkların “Process” içerisinde ne kadar zaman geçireceğini belirtmek için öncelikle “Delay Type” seçilmelidir. Bu listede “Constant” (Sabit) seçilirse, varlıkların süreç içinde sabit bir süre kalacaklarını belirtmiş oluruz. Bizim problemimizde varlıklar üssel dağılımdan ortalama 10 dk kaldıklarına göre, “Delay Type” olarak “Expression” seçilir ve alttaki listeden “EXPO (mean)” seçilir. “mean” yazan yere 10 yazılır. “Units” bölümü “Minutes” olarak seçilmelidir. Böylece ortalaması 10 dk. olan üssel dağılımdan değerler üretilecektir.

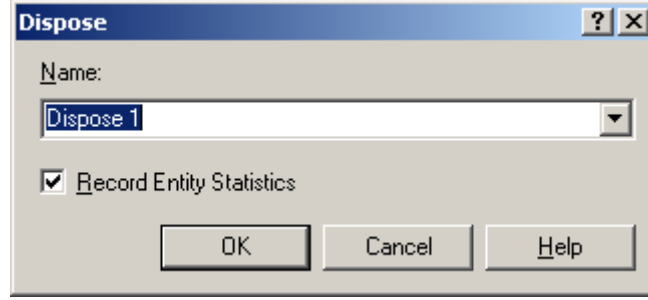
Bu işlemler bittiğinde modelin süreç yerleşiminde “Process 1” üzerinde bir çizgi görülecektir. Bu çizgi “Process 1”den önceki kuyruğu temsil etmektedir.

2.5 “Dispose” Modülü

Create modülü ile yaratılmaya başlayan her varlık süreç yerleşiminin en sonunda “Dispose” modülüne ulaşmalıdır (Şekil

2-4). Bu modül sadece “Name” (isim) detayını içermektedir.

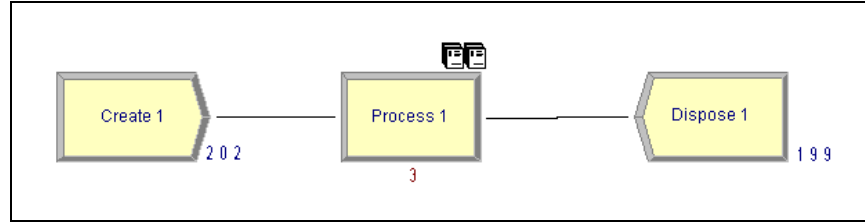
Şekil 2-4 Dispose modülü



2.6 Modelin ilk defa çalıştırılması

Modeli çalıştırmak için araç çubuğundan “Go” butonuna basılır. Eğer her şey yolundaysa ikonların hareket ettiğini ve bazen “Process 1”in üstünde bekleyen varlıkların olduğunu göreceksiniz.

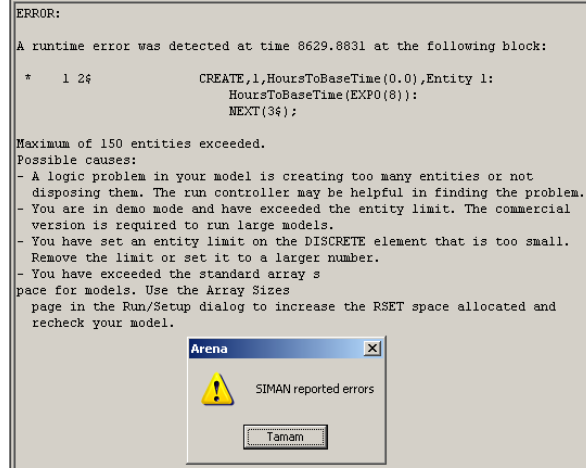
Şekil 2-5 Modelin çalıştırılması ve animasyon



Model bu halde iken çalıştırılmaya devam ettiğinde bir süre sonra aşağıdaki gibi bir hata mesajı görünecektir. Bu hatanın oluşması doğaldır çünkü Arena'nın öğrenci sürümü, model içerisinde aynı anda 150 varlık olmasına izin vermektedir. Şu ana kadar modelimizin kuyruğunda biriken varlık sayısı bu sayıyı aşmıştır. Yani kuyrukta 150 varlık beklemektedir. Peki bu beklenen bir şey midir? Dikkat edilirse gelişler arası süre ortalama 8 dk. ve hizmet süresi de her bir sunumcu için 10 dk.'dır. Bu durumda kuyruk oluşması doğaldır ama unuttuğumuz bir nokta var ki o da 2 sunumcunun hiçbir yere girilmemiş olmasıdır. Yani model, ilk bölümde tanımlanmış olmasına rağmen hala 1 sunumcu ile çalışmaktadır.

Bu sorunu gidermek ve modelimizi tanımlandığı hale getirmek için bir sonraki bölümde anlatılacak kaynak değeri değiştirilmelidir.

Şekil 2-6 Hata mesajı



2.7 “Resource” Modülü

Kaynaklara ilişkin bilgilerin girildiği “Resource” modülü, tablo şeklinde bir modüldür. “Basic Process” proje öğelerinin bulunduğu yerde “Resource”a tıklanınca süreç yerleşimi alanında bir şey görünmeyecek ama tablolaştırma alanında şekildeki detaylar görünecektir.

Şekil 2-7 Resource detayları

Resource - Basic Process									
	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1 ▶	Resource 1	Fixed Capacity	2	0.0	0.0	0.0		0 rows	<input checked="" type="checkbox"/>

Double-click here to add a new row.

Modeldeki sunumcu adı “Resource 1”in sabit kapasiteli ve 1 olduğu görünecektir. Eğer kapasite (Capacity) 2 olarak girilirse, bu, modelde 2 adet “Resource 1” tipinden kaynak olduğunu belirtecektir. Şimdilik diğer detayları olduğu gibi bırakacağız.

2.8 “Entity” Modülü


Resource modülünde olduğu gibi Entity modülü de tablolaştırma alanında görüntülenen bir modüldür. Bu modül ile modelinizdeki Entity’ler için bazı detaylar girilebilir. Örneğin animasyonda varlıkların resmini değiştirmek isterseniz, şekilde görülen “Initial Picture” değiştirilebilir.

Şekil 2-8 Varlıkların ikonlarının değiştirilmesi

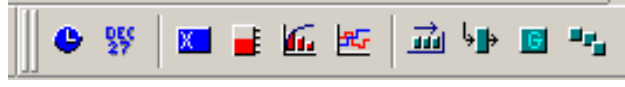
Entity - Basic Process									
	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran Cost	Initial Other Cost	Report Statistics
1 ▶	Entity 1	Picture.Blue Ball	0.0	0.0	0.0	0.0	0.0	0.0	<input checked="" type="checkbox"/>

“Initial Picture” listesine eklemeler yapmak ise “Edit | Entity Pictures” menü seçeneği ile mümkündür.

2.9 Model çıktısının grafik ile gösterimi

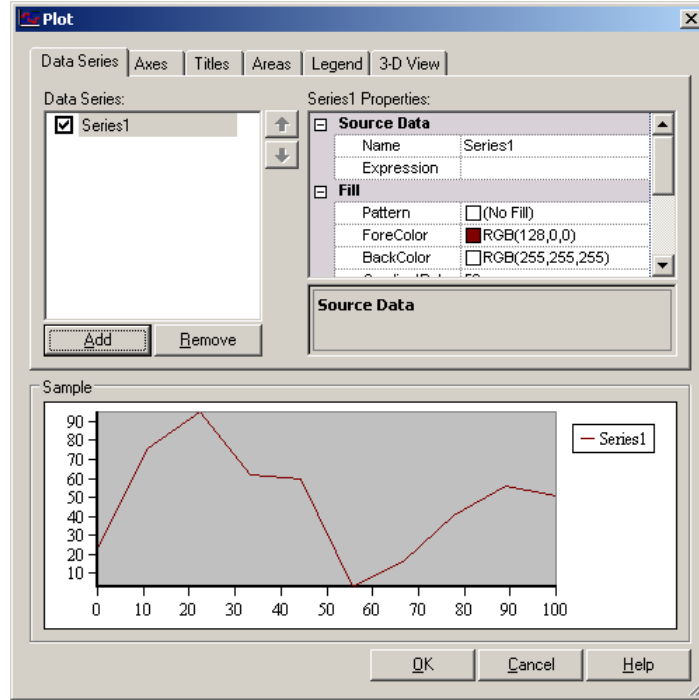
Model çalışırken bir taraftan da modele ilişkin değerleri grafik veya sayaç olarak gösterebilirsiniz. “Animate” araç çubuğunda bulunan “Plot”  seçeneğine tıkladığınızda yeni bir pencere çıkacaktır.

Şekil 2-9 "Animate" araç çubuğu



Bu pencerede çizdirilecek grafiğe ilişkin detaylar yer almaktadır. “Add” (Ekle) düğmesine tıkladığınızda aşağıdaki ekran görüntüsü karşınıza çıkacaktır. Burada “Series 1” grafiği çizilecek veri setidir ve alacağı değerler “Properties” listesindeki “Expression”a girilmelidir.

Şekil 2-10 Plot özellikleri

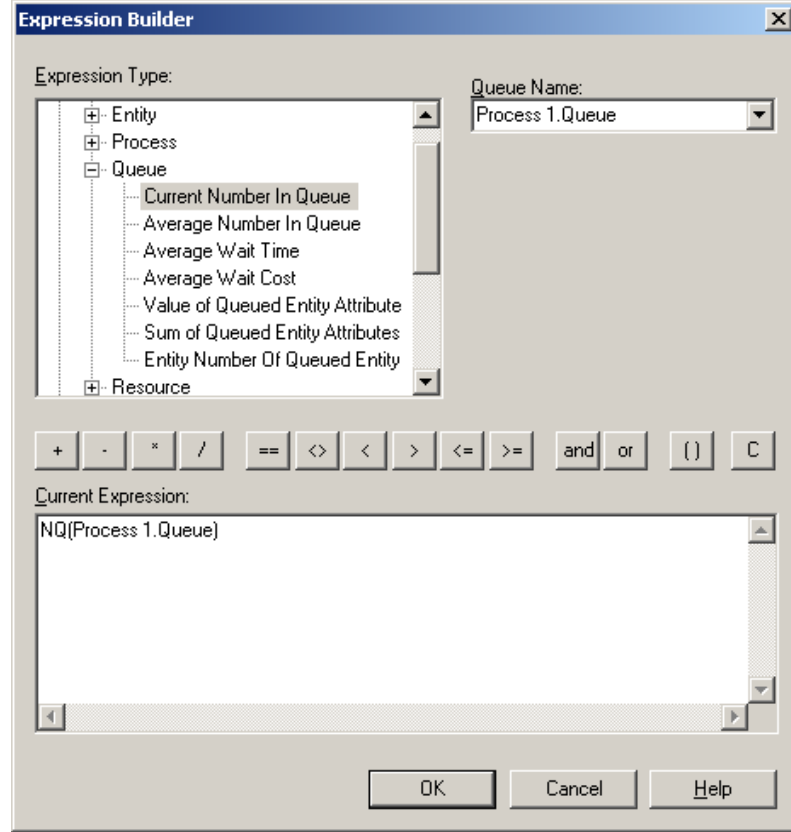


“Expression” listesinde “.NumberOut” ve “NumberIn” ifadeleri ile birlikte en üstte “Build Expression...” yer almaktadır. Bu bölüm aslında Arena’nın en önemli özelliklerinden olan “Expression Builder”ı yani “**deyim kurucu**”yu çalıştırmaktadır. Bilgisayar programı yazmak yerine, deyim kurucu ile Arena’nın dili olan SIMAN’da komutlar yazabilirsiniz. Deyim kurucu, Arena’nın hemen hemen her yerinde kullanılabilir. Örneğin bu örnek için kuyruktaki aktif varlık sayısının grafiğini çizmek istiyorsak; “Expression Type”dan “Queue” ve “Current Number In Queue” seçilebilir. Böylece şekilde görüldüğü gibi

NQ(Process 1.Queue)

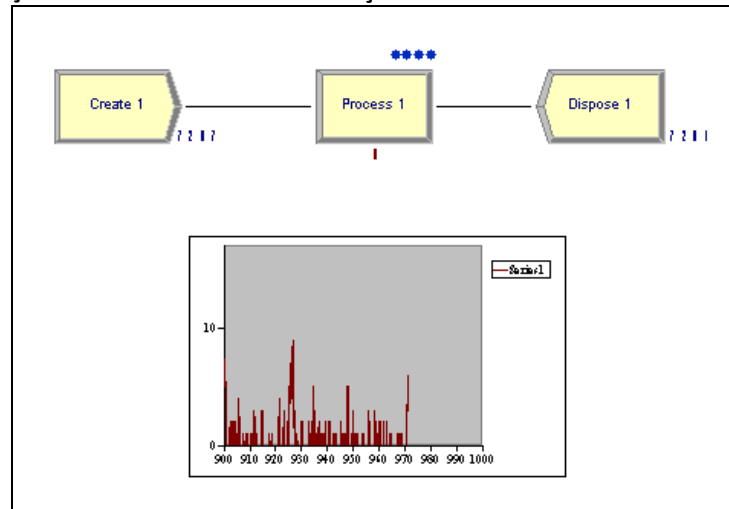
ifadesi oluşturulmuş olur. “Current Expression” bölümüne bilinen komutlar elle de yazılabilir.

Şekil 2-11 Deyim kurucu



Bu işlemler tamamlandıktan sonra farenin işaretçisi değişecektir. Süreç yerleşimi alanına uygun bir yere grafiğinizi ekleyebilirsiniz. Şimdi modeli çalıştırdığınızda grafiğin de güncellendiğini göreceksiniz.

Şekil 2-12 Modelin tamamlanmış hali



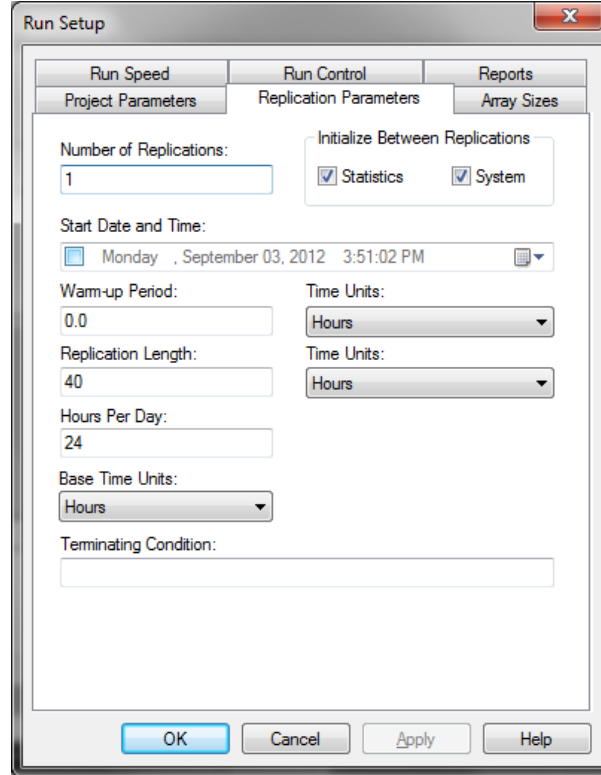
Model tamamlandıktan sonra ve tekrar çalıştırılınca Şekil 2-12'ye benzer bir görüntü oluşacaktır. Entity'leri mavi noktalar olarak göreceğiniz grafiğin ise dalgalandığını göreceksiniz.

2.10 Model çalışma süresinin belirtilmesi

Modeli ilk defa çalıştırdığımızda bir hata oluşmuştu ama daha sonra bunu gidermiştik. Şimdi tekrar başlattığımızda ise modelin hiç durmadan çalışacağını göreceksiniz. Bunun nedeni modelin "Run length"inin yani çalışma süresinin varsayılan değerinin sonsuz olmasıdır. Model çalışmasına ilişkin parametrelere "Run | Setup" ile ulaşabilirsiniz.

Bu ekranda "Number of Replications" ile modelin kaç kere çalıştırılacağı, "Warm-up Period" ile modelin durağan duruma ne zaman ulaşacağı ve "Replication Length" ile de modelin ne kadar süre ile çalıştırılacağı belirtilebilir. Bu model için çalıştırma süresini 40 saat olarak girebiliriz.

Şekil 2-13 Modelin çalıştırılmasına ilişkin parametreler



Bu süre kadar model çalıştıktan sonra bir mesaj ile model sonuçlarının görüntülenmesinin istendiği sorulur. Eğer evet seçilirse hiyerarşik şekilde modelin sonuçları görüntülenir. "Unnamed Project" in altında Entity, Queue, Resource başlıklarında sonuçlar görüntülenebilir. Projenize Şekil 2-13'deki "Project Parameters" sekmesindeki "Project Title" dan isim verebilirsiniz.

Bu raporun “Queue” seçeneğinin “Time” bölümünde, örneğin, “Process 1.Queue” ortalama olarak (average) 0.069 dk. beklenildiği görülmektedir. Burada ortalama değerin olmasının nedeni modelin çok defa çalıştırdığımızda çıktı değişkeni değerlerinin ortalamasının belirtileceğidir. Bu modeli bir kez çalıştırmış olduğumuzdan burada yazan değer aslında ortalama değil çıktı değişkeni değerinin kendisidir. Dolayısıyla da “Half Width” değeri hesaplanmamıştır.



Bu örnekte
Arena’da

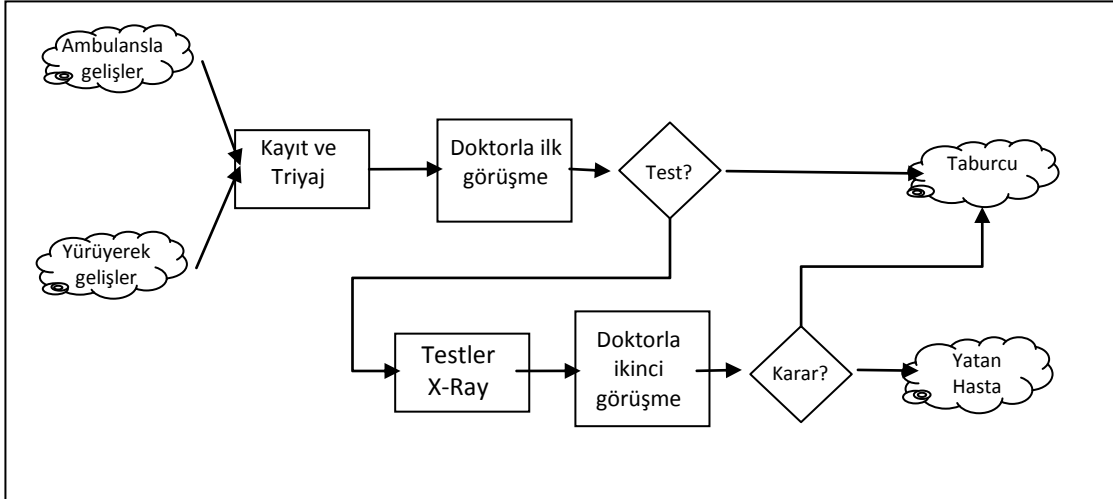
- Create Modülü
- Process Modülü
- Dispose Modülü
- Resource Modülü
- Entity Modülü

3 ACİL SERVİS MODELİ

3.1 Tanım

Bu bölümde bir hastanenin Acil Servis (AS)'inin benzetim modeli yaratılacaktır. Bu AS'e iki şekilde hasta gelişi olmaktadır; yürüyerek ve ambulansla. Her iki hasta tipi için de; gelişlerinden hemen sonra kayıt ve triyaj yapılmaktadır. Trijaj, hastaları rahatsızlıklarına göre önceliklendirme işlemi olup uzman bir hemşire tarafından yapılmaktadır. Kayıt işleminin de bu sırada yine aynı hemşire tarafından yapıldığı varsayılmaktadır. Triyaj sonucunda hastalara kırmızı, sarı, yeşil renklerinden birisi verilerek önceliklendirilmektedir. Gerçek hayatta bu önceliğe göre hastalar doktor muayenesi için kuyruğa girdiklerine rağmen biz şimdilik bu modelde Doktorla ilk görüşme için yine FIFO bir kuyruk olduğunu varsayalım. Ayrıca bu doktorla ilk görüşme sırasında bir hemşirenin de hazır bulunması gerekmektedir. Şekilde tüm süreç gösterilmiştir.

Şekil 3-1 Acil Servis süreçlerinin şematik gösterimi



İlk doktor muayenesi renklere göre farklı sürelerde olmaktadır. Muayene sonucunda hastaların bazıları taburcu edilmekte ve sistemden dışarı çıkmaktadır. Bazı hastalardan ise test ve X-Ray gibi tetkikler istenebilmektedir. Elbette bu da zaman almakta ve yine öncesinde bir FIFO kuyruk oluşmaktadır. Test sonuçlarını alan hastalar daha sonra tekrar doktorun kontrolünden geçmekte (Doktorla ikinci görüşme) ve bir karar verilmektedir. Doktorla ikinci görüşmede sadece doktor bulunmakta, hemşire yer almamaktadır. Bazı hastalar taburcu olurken bazıları ise hastaneye yatırılmaktadır.

Tanımlanan bu sistemde aşağıda belirtilen girdilere ihtiyaç vardır. Modelimiz içinde bu girdilerin alacağı değerler ise yine tabloda belirtilmiştir.

Tablo 3-1 Model girdileri ve deęerleri

Girdi	Deęer
Süreler	
Ambulans gelişleri arası süre	Üssel dağılım (ortalaması 30dk.)
Yürüyerek gelişler arası süre	Üssel dağılım (ortalaması 5dk.)
Kayıt ve Triyaj süresi	Üçgensel dağılım (en az:2dk, genelde 5dk. En çok:10dk.)
Doktorla ilk görüşme	Kırmızı hastalar için: Lognormal dağılım (ort.30 dk.,std.sapma 15 dk.) Sarı hastalar için: Üssel dağılım (ort.20 dk.) Yeşil hastalar için: Üçgensel dağılım (en az:5 dk, genelde 8 dk., en çok:12 dk.)
Testler ve X-Ray	Tüm hastalar için: Üçgensel dağılım (en az:20 dk, genelde 40 dk., en çok:60 dk.)
Doktorla ikinci görüşme	Tüm hastalar için: Üçgensel dağılım (en az:5 dk, genelde 10 dk., en çok:15 dk.)
Oranlar	
Trijaj rengi oranları	Ambulansla gelenler için: %70 Kırmızı %30 Sarı Yürüyerek gelenler için: %1 Kırmızı %19 Sarı %80 Yeşil
Test ihtiyacı oranı	Sarı ve Kırmızı hastalar için: %10 ihtiyaç var %90 ihtiyaç yok Yeşil hastalar için test istenmiyor.
İkinci doktor görüşmesi sonrası doktorun hastayı taburcu etme kararı	Tüm hastalar için: %20 Hasta yatsın %80 Taburcu olsun
Kaynaklar	
Doktor sayısı	3
Hemşire sayısı	6
Test Makinesi	1

Her modelleme projesinde olduğu gibi bu projede de bir amacımızın olması gerekmektedir. Bu örnekte;

- İlk doktor muayenesi için ortalama bekleme zamanınının 10 dk.'yı geçmemesini,
- Doktorun kullanım oranınının (utilization) %70'i geçmemesini

istiyoruz.

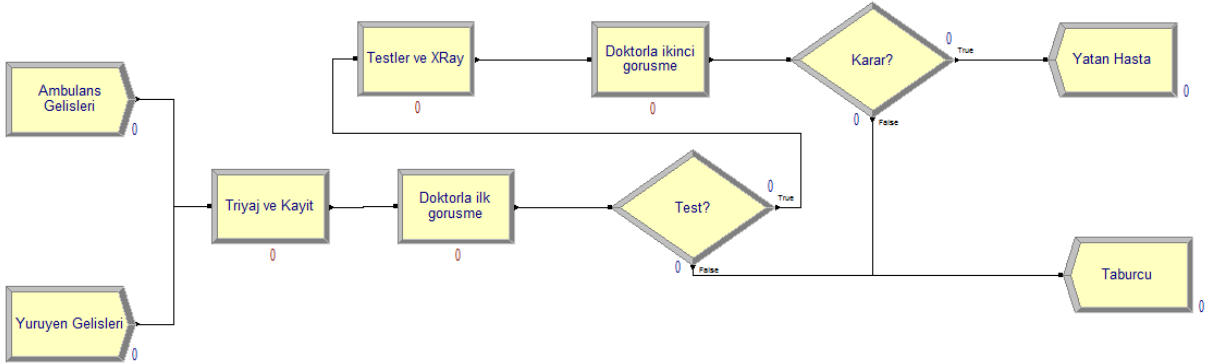
Öncelikle modelin bu parametre değerleri ile bu hedeflerimizi sağlayıp sağlamadığına bakacağız. Eğer sağlamıyorsa sistemde bazı değişiklikler önererek bunların etkisini inceleyeceğiz.

3.2 Genel Görünüm (Süreç Yerleşimi)

Önceki basit kuyruk modelinde olduğu gibi Arena’da modellemeye başlamak için ilk yapmamız gereken Arena modüllerini kullanarak tanımlanan sistemin süreç diyagramını çizmeliyiz. Aslında yapılan işlem **Error! Reference source not found.**’de basitçe çizilen şemayı Arena modülleri ile çizmek olacaktır.

Şekil 3-2’de görünen süreç yerleşiminde, daha önce kullanılan süreçlere ilave olarak eşkenar dörgen şeklindeki Karar (Decision) modülünü kullanılmıştır. Bu haliyle modelimiz henüz ilk aşamasındadır ve sistemi modelleme için yeterli değildir. Bu haliyle Şekil 3-1’deki gösterimin aynısıdır.

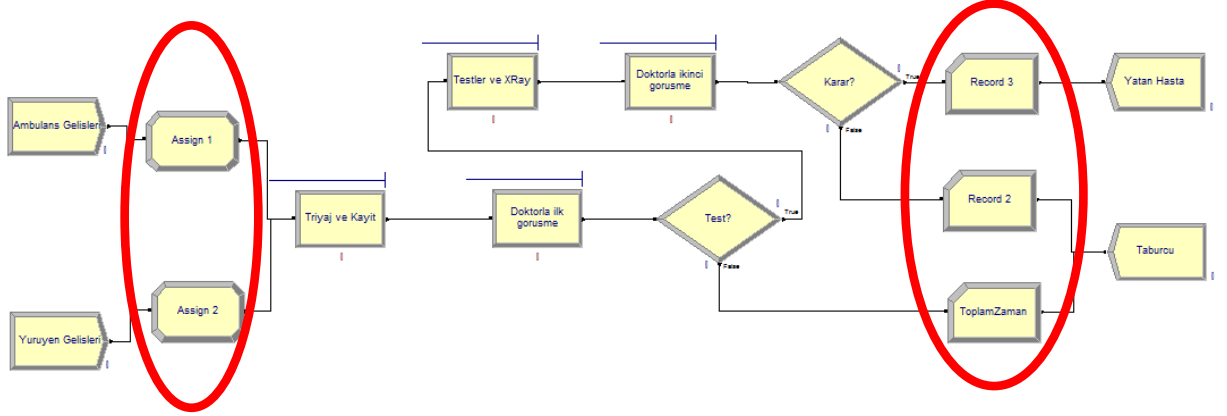
Şekil 3-2 Acil Servis modeli süreç yerleşimi (ilk aşama)



Bu modelde bazı istatistikler kullanabilmek için ve hasta tiplerine bağlı olarak süreç zamanlarını modelleyebilmek için ilave modülleri kullanmamız gereklidir. Şekil 3-3’de görüldüğü gibi Create modüllerinden hemen sonra birer tane “Assign” modülü yerleştirilmelidir. Benzer şekilde Dispose modülünden hemen önce üç adet “Record” modülü yerleştirilmelidir.

Artık modüllerimizi detaylandırmaya geçebiliriz.

Şekil 3-3 Acil Servis modeli süreç yerleşimi (ikinci aşama)



Modellediğimiz sistemde iki tip geliş olduğundan aynı şekilde iki adet "Create" modülü eklenmiştir. Herikisinde de "Entity Type" olarak "Hasta" yazabilirsiniz. Hastaların nasıl geldiklerini bir sonraki bölümde "Entity Attribute" kavramı ile göreceğiz. "Name" ve "Time Between Arrivals"daki "Value" ve "Units" bölümlerini Tablo 3-1'deki ilgili değerler ile doldurunuz.

3.3 "Assign" Modülü

Modelediğimiz sistemde iki tip geliş olduğundan ve bu geliş tipine göre sistem içerisinde bazı parametreler değer aldığından (Trijaj rengi oranları) gelen varlıklara geliş tiplerini kaydetmek için bir mekanizma gereklidir. "Assign" modülü işte bu işe yarayan bir modüldür. Yani Entity'lere bir özellik (attribute) değeri atamada kullanılırlar.

"Assign 1"e çift tıklayıp açılan pencerede "Add" düğmesine basın ve "Type" olarak "Attribute" seçin. "Attribute Name"e geliş tipi ile ilgili bir ifade yazacağımızdan, örneğin, "gelisTipi" yazalım. Burada Türkçe karakter kullanmamaya dikkat ediniz. "New Value" kısmına ise 1 yazalım (Şekil 3-4). Kullandığımız 1 değeri Ambulansla gelişleri temsil etmektedir.

Şekil 3-4 Attribute "GelişTipi"

Gelen hastaların geliş zamanlarını da kaydetmek istiyoruz. Bunun için yine her hastaya bir "Attribute" (Özellik) tanımlayacağız. Yine

Assign1'de "Add" düğmesine basıp "Type" olarak "Attribute" ve isim olarak da "GelisZamani" giriniz. Buradaki değer ise Entity'nin geldiği zamanı ifade edecek bir şey olmalıdır. Bunun için uygun Arena fonksiyonu "TNOW"dır. Bu fonksiyon o anki benzetim zamanını ifade etmektedir. "Value" kısmına "TNOW" yazınız.

Son olarak gelen ambulans hastasının Triyaj rengini burada belirtmeliyiz. Yine bunun için bir hasta özelliği tanımlayıp bu özelliğe bir değer atayacağız. Bunun için yine Assign 1'de "Add" düğmesine basıp Type olarak "Attribute" seçelim. "Attribute Name" olarak "TrijajRenk" yazalım. Değer bölümüne ise Tablo 3-1'de belirtildiği şekilde %70'i kırmızı ve %30'u sarı olarak belirtmeliyiz. Bunu yapabilmemizin bir yolu bu olasılık değerleri ile renk değerleri atamaktır. Arena'nın "DISC" fonksiyonu discrete (kesikli) olasılıklarla bir değer geri döndüren hazır bir fonksiyondur. "Value" bölümüne aşağıdaki ifadeyi yazalım.

DISC(0.7, 1, 1.0, 2)

İlk parametre ilk değer için olasılık değeridir. "0.7", ondalık ayırıcı "." olmak üzere, %70 olasılığı temsil eder. Sonrasında gelen "1" ise %70 ihtimalle elde edilecek değerdir. Bizim durumumuzda bu "Kırmızı" rengi temsil etmektedir. DISC fonksiyonu birikimli olarak olasılık değeri alır. Dolayısıyla sonraki "1.0" ise %70'den kalan bütün olasılıkları, yani %30'u temsil etmektedir. Bu ise "2" değerini verecektir, yani Sarı rengin değeri olacaktır.

Şekil 3-5'de Assign 1 ve 2'nin belirtildiği şekilde doldurulduktan sonraki halleri görünmektedir. Assign 2'de yine 1'deki gibi yine aynı "Attribute"lar tanımlanacak sadece değerleri farklı olacaktır. "gelisTipi" 2 ve "TrijajRenk" de aşağıdaki gibi olacaktır.

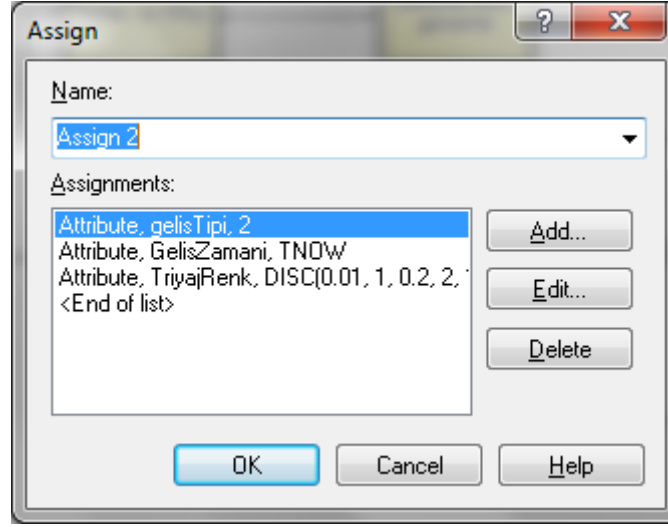
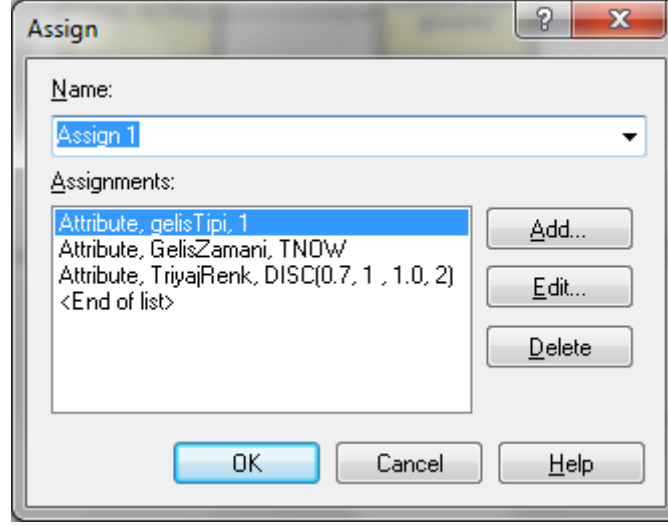
DISC(0.1, 1, 0.2, 2, 1.0, 3)

Bu DISC fonksiyonu %10 olasılıkla 1 (yani kırmızı), %19 olasılıkla 2 (yani sarı) ve %80 olasılıkla 3 (yani yeşil) değeri atayacaktır.

3.4 "Process" Modülü

Modelde kullanılan ilk process modülü "Trijaj ve Kayıt" modülüdür. Bu modülün doldurulmuş hali Şekil 3-6'da görünmektedir. Burada 1 hemşire görev alacağından "Resources" bölümünde "Hemsire" tipinden 1 kaynağa ihtiyaç olduğu belirtilmiştir. Ayrıca "Action" bölümünde "Seize Delay Release" seçilmiş olduğu bu süreç için bir FIFO kuyruk olacağını belirtmektedir. Süreç zamanı olarak "Triangular" dağılımı ve değerleri olarak da 2, 5 ve 10 değerleri girilmiştir.

Şekil 3-5 Assign modüllerinin doldurulmuş halleri



Şekil 3-7’de ise “Doktorla ilk gorusme” modülünün doldurulmuş hali görülmektedir. Yine FIFO kuyruk olacağı belirtilmiş ama bu sefer, problem tanımında belirtildiği üzere, işlemin hem 1 doktor hem de aynı anda 1 hemşire ile yapılacağı belirtilmiştir. Bu modüle süreç zamanının belirtildiği kısımfa farklı bir yapı kullanılmıştır. “Delay Type” olarak “Expression” seçildiğinde süre değerinin aşağıdaki belirtilen “Expression” ifadesinin vereceği belirtilir. Bu ifade Arena’nın “Advanced Process” modüllerinden birisi olan “Expression” modülü ile oluşturulmuş bir yapıdır ve bir sonraki bölümde açıklanmıştır.

Modelin diğer iki süreci olan “Testler ve X-Ray” ile “Doktorla ikinci görüşme” uygun şekilde doldurulur.

Şekil 3-6 Triyaj ve Kayıt modülü

The screenshot shows a 'Process' dialog box with the following fields and options:

- Name:** Triyaj ve Kayıt
- Type:** Standard
- Logic:**
 - Action:** Seize Delay Release
 - Priority:** Medium(2)
- Resources:**
 - Resource, Hemsire, 1
 - <End of list>
 - Buttons: Add..., Edit..., Delete
- Delay Type:** Triangular
- Units:** Minutes
- Allocation:** Value Added
- Minimum:** 2
- Value (Most Likely):** 5
- Maximum:** 10
- Report Statistics
- Buttons: OK, Cancel, Help

Şekil 3-7 Doktorla ilg görüşme modülü

The screenshot shows a 'Process' dialog box with the following fields and options:

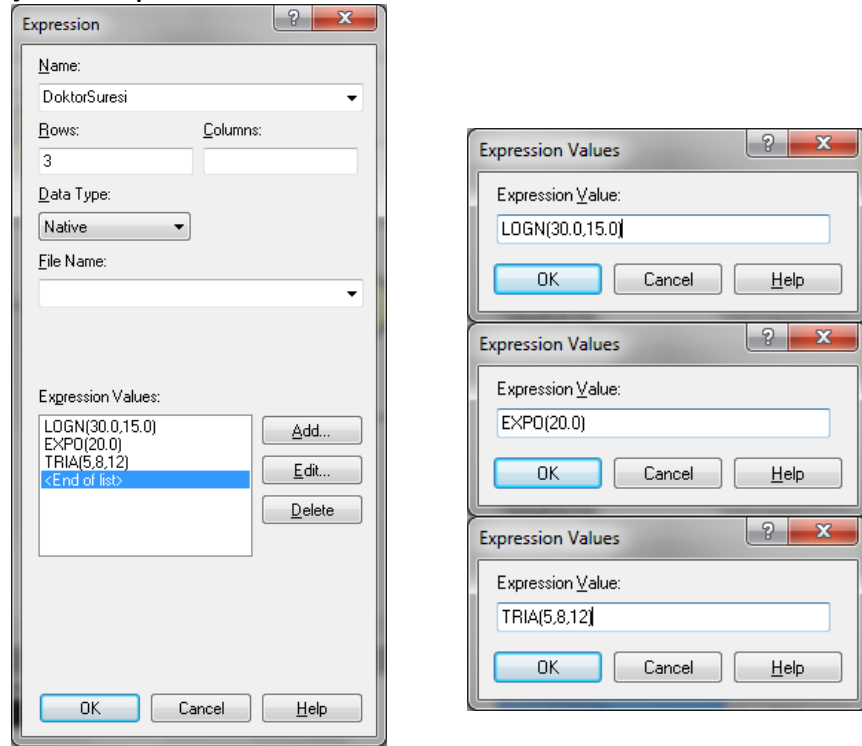
- Name:** Doktorla ilk görüşme
- Type:** Standard
- Logic:**
 - Action:** Seize Delay Release
 - Priority:** Medium(2)
- Resources:**
 - Resource, Doktor, 1
 - Resource, Hemsire, 1
 - <End of list>
 - Buttons: Add..., Edit..., Delete
- Delay Type:** Expression
- Units:** Minutes
- Allocation:** Value Added
- Expression:** DoktorSuresi(TrijajRenk)
- Report Statistics
- Buttons: OK, Cancel, Help

3.5 “Expression” Modülü

“Expression” modülü şuna kadar kullandığımız modüllerin bulunduğu “Basic Process” template’inde yer almamaktadır. Bu modül Project Bar’daki “Advanced Process” template’indedir. Eğer

ekranın sol tarafında bu şablon görünmüyorsa bölüm 4.4’de anlatıldığı gibi eklenebilir. “Expression” template’i çizelge şeklindedir (sürükle bırak ile modele eklenmez). Çift tıklayarak yeni bir eleman ekleyiniz ve Şekil 3-8’deki gibi doldurunuz. Burada “Name” belirttiniz, “Doktorla ilk görüşme” modülündeki expression’daki ifadenin aynısıdır. “Expression” ile aslında bir dizi (array) tanımlanmaktadır. Bu dizinin kaç elemanı olduğu “Rows” da belirtilmiştir. Eğer iki boyutlu bir dizi isteniyorsa “Columns”a da bir değer girilebilir. Bizim dizimiz 3 satırdan oluşmakta ve her satırdaki değerler ise “Expression Values”a tek tek eklenmektedir. Eklenecek değerleri elle yazabildiğiniz gibi mouse’un sağ tuşuna bastığınızda “Build Expression ...” ve “Random Distributions” seçerek de yazabilirsiniz.

Şekil 3-8 Expression modülü



Burada belirtilen değerler aslında birer olasılık dağılımıdır. Bu şekilde yapmamızın nedeni Tablo 3-1’de belirtildiği üzere hastaların triyaj renklerine bağlı olarak “Doktorla ilk görüşme” sürelerinin, yani dağılımlarının, farklı olmasıdır. “DoktorSuresi” adında bir dizi oluşturarak, “TrijajRenk” indeksi ile doğru dağılım çağrılacaktır. Yani TriyajRenk 1 ise Lognormal, 2 ise Üssel, 3 ise Üçgensel dağılımdan bir değer döndürülecektir. Buradaki ekleme sırası önemlidir çünkü DoktorSuresi(1) listedeki ilk elemanı verecektir.

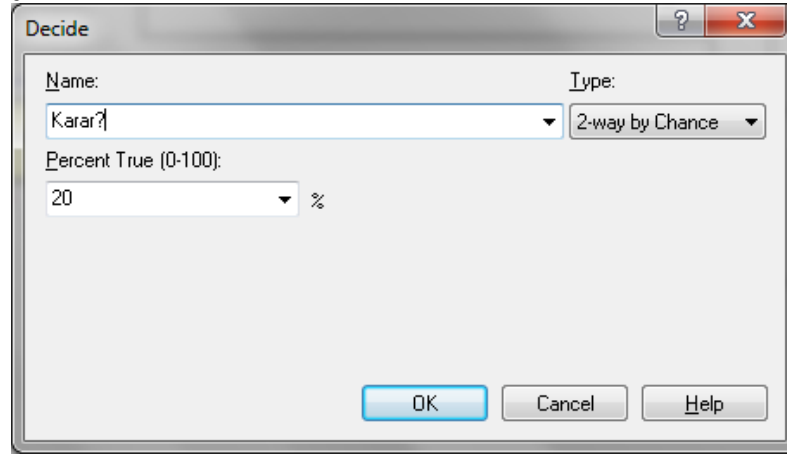
3.6 “Decide” Modülü

Modelde iki adet Decide modülü kullanılmıştır. Bu modüller

Entity'nin yol ayrımlarını temsil etmektedir. Bir koşula veya bir olasılığa bağlı olarak bu modüle gelen varlıklar yön değiştirirler.

Öncelikle daha basit olduğundan "Doktorla ikinci görüşme"den sonraki "Decide" modülü açıklanacaktır. Şekil 3-9'da bu modülün doldurulmuş hali görülmektedir. "Type" olarak iki yönlü şansa dayalı olarak karar tipi seçilir. Bu durumda "Percent True" alanına 0 ve 100 arası bir değer girilir. Bu olasılıkla gelen Entity (bu modelde Hasta) Decide'in "True" yönüne gidecektir. Tabi her gelen Entity için farklı bir yöne olacaktır. Eğer o Entity için üretilen olasılık değeri bu alana girilen değerden (%20'den) büyükse "False" yönüne gidecektir.

Şekil 3-9 "Karar" Decide modülü



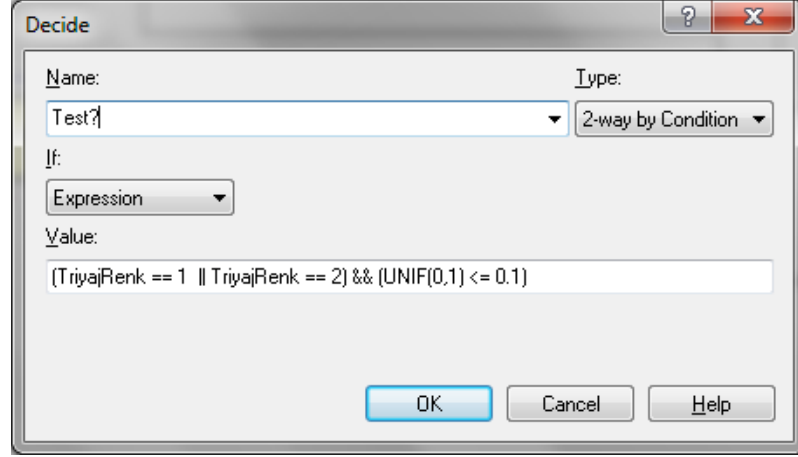
"Doktorla ilk görüşme" sonrasındaki Decide modülü Şekil 3-10'da görülmektedir. Bu modül ise iki yönlü ve koşula göre Entity'ye yön verir. Koşul çeşitli şekillerde belirtilebilir. Bir "Attribute" değerine bağlı olarak (örneğin TriyajRenk 1 ise) veya bir değişkene bağlı olarak belirtilebilir. Bizim örneğimizde ise, Tablo 3-1'de belirtildiği üzere, Hastanın test ihtiyacı oranı hem triyaj rengine göre hem de olasılık değerine göre belirlenmektedir. Bu durumda önce hastanın triyaj rengi kontrol edilmelidir. İlk parantezli ifade triyaj renginin 1 mi veya 2 mi olduğunu kontrol eder. "==" ifadesi eşit olup olmadığını, "|" ise veya ifadesini temsil etmektedir. Bu ifade "Trijaj rengi kırmızı veya sarı mı?" şeklinde bir ifadedir.

(TrijajRenk == 1 || TriyajRenk == 2)

Takip eden "&&" ise "ve" bağlacıdır. İkinci parantez ifadesi 0 ile 1 arasında bir düzgün dağılmış değeri (yani bir rassal sayıyı) %10'dan küçük olup olmadığını kontrol eder. Eğer gelen varlık için üretilen değer %10'dan küçükse yine "True" değeri elde edilmiş olacak ve dolayısıyla varlık "Testler ve X-Ray" modülüne gönderilecektir.

&& (UNIF(0,1) <= 0.1)

Şekil 3-10 Test Decide modülü



Decide

Name: Test? Type: 2-way by Condition

If: Expression

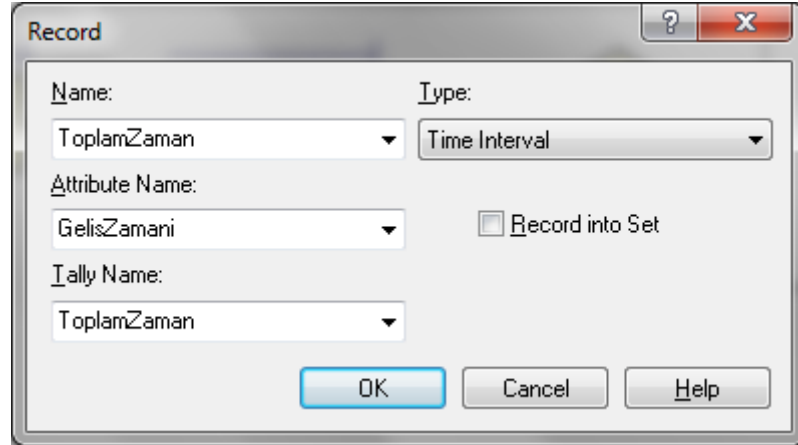
Value: (TriyajRenk == 1 || TriyajRenk == 2) && (UNIF(0,1) <= 0.1)

OK Cancel Help

3.7 "Record" Modülü

Record modülleri istatistik toplamak için kullanılan modüllerdendir. Modelimizin en sonunda 3 adet kullanılmıştır. En altta yer alan "ToplamZaman" modülü Şekil 3-11'de gösterilmektedir. Bu modülde "Type" olarak "Time Interval" seçilir. Böylelikle iki zaman arasındaki farkı data olarak kaydedecektir. Entity bu modüle geldiği zaman o anki benzetim saati ile "GelisZamani" varlık özelliğindeki değerinde yazan saati birbirinden çıkartarak "ToplamZaman" çetele tablosuna yazar. Bu tablodaki değerlerin özetleri (ortalama, standart sapma) model çalışması bittikten sonra görülebilir.

Şekil 3-11 Record modülü



Record

Name: ToplamZaman Type: Time Interval

Attribute Name: GelisZamani Record into Set

Tally Name: ToplamZaman

OK Cancel Help

Modelimizdeki diğer record modülleri de bu modül gibidir. Sadece "Name" bölümleri farklıdır. Diğer bölümler ise, "Type", "Attribute Name" ve "Tally Name" ise aynıdır.

3.8 Modelin çalıştırılması ve deneyler

Modelin çalışma süresini (Replication Length) 2880 dakika olarak, tekrar sayısını ise 10 olarak girip çalıştırıldım.

Raporun "Queue" bölümüne gittiğimizde aşağıdaki sonuçları görürüz. Sizin değerleriniz bunları aynısı olmasada yakın değerler olmalıdır. Burada ilk gözümüze çarpan "Doktorla ilk gorusme.Queue" değerinin 117.41 dakika gibi çok yüksek bir değeri olduğudur. Hastalarımız ortalama 117 dakika, neredeyse 2 saat, doktoru beklemektedir. Bu istediğimiz hedefin çok çok uzağındadır. Hedefimiz hastaların ortalama 10 dk. beklemesi, diğer taraftan doktorların %70 oranında çalışmasıdır.

5:20:21PM **Category Overview** September 3, 2012
Values Across All Replications

Acil Servis Modeli

Replications: 10 Time Units: Minutes

Queue

Time

Waiting Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Doktorla ikinci gorusme.Queue	117.08	48.68	33.9898	233.19	0.00	343.74
Doktorla ilk gorusme.Queue	117.41	39.20	40.6602	221.34	0.00	378.74
Testler ve XRay.Queue	8.5987	4.02	3.7715	22.6200	0.00	126.07
Triyaj ve Kayit.Queue	0.3020	0.04	0.2353	0.3887	0.00	8.9821

Other

Number Waiting	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Doktorla ikinci gorusme.Queue	0.7440	0.36	0.2124	1.7406	0.00	6.0000
Doktorla ilk gorusme.Queue	27.6176	9.30	9.0436	53.1942	0.00	88.0000
Testler ve XRay.Queue	0.05819669	0.03	0.02488154	0.1806	0.00	3.0000
Triyaj ve Kayit.Queue	0.07141842	0.01	0.05522315	0.0953	0.00	5.0000

İlk hedefimizi tutturamadığımızdan ikincisine bakmadan çare düşünelim. Kurukta bekleme zamanını düşürmenin bir yoluda kaynakların artırılmasıdır. Bu çalışmamızda biz de doktor kaynağının bekleme zamanlarını indirmemiz için gerekli olduğunu düşündüğümüzden doktor sayısını 1 arttırarak etkisinin nasıl olacağını inceleyeceğiz.

Resource'lardan Doktor sayısını (Capacity) 4 yapıp modeli tekrar çalıştırıldım. Yine 10 tekrar yaptığımızda bu sefer doktor ilk görüşme kuyruğunun 9.2 dakikaya düştüğünü göreceksiniz. Doktor sayısını 1 arttırmakla bekleme zamanları ciddi oranda değişmektedir. Bu aynı zamanda ilk hedefimizin de tuttuğunu göstermektedir. Diğer taraftan ikinci hedefimiz olan doktor kullanım oranına bakabiliriz.

Sonuç raporunun "Resource" bölümünün "Instantaneous

Utilization” bölümünde doktorun zamanının %80’inde meşgul olduğu görülmektedir. Hemşireler ise %74 yoğunurlar. Doktor hedefimiz %70 olduğundan belki 1 tane daha doktor ekleyip deniyebiliriz.



Bu örnekte
Arena’da

- Assign Modülü
- Decision Modülü
- Record Modülü
- Advanced: Schedule Modülü
- Advanced: Failure Modülü

4 BOĞAZ GEÇİŞ MODELİ

4.1 Tanım

İstanbul Boğazı'nın Marmara - Karadeniz yönünde geçişinin benzetimini yapmak istiyorsunuz. Boğazdan geçmek isteyen gemiler Yenikapı açıklarına demirleyip bir kılavuz kaptan gelmesini beklemektedir. Kılavuz kaptan geldikten sonra boğaz geçişi yapılmakta ve Karadeniz'e çıkan gemi yoluna devam etmektedir. İstanbul Boğazı'nın bir kuyruk sistemi olarak benzetimini yapınız.



4.2 Model

4.2.1 Parametreler

Gemilerin Yenikapı'ya gelişlerinin rassal olduğunu ve gelişler arası sürenin üssel dağılımdan geldiğini ortalaması 1 saat olduğunu bilmektesiniz.

Bir kılavuz kaptanın ana merkezden Yenikapı'da bekleyen kılavuzluk hizmeti vereceği gemiye ulaşması süresinin üssel dağılıma uyduğunu ve ortalamasınının 30 dk. olduğunu bilmektesiniz.

Gemilerin sabit hızla boğazdan geçmesi gerektiğinden geçiş süreleri arasında önemli bir fark olmamakta ve sabit 5 saat sürmektedir.

Bir kılavuz kaptan geminin boğaz geçişi süresince başlangıçtan çıkışa kadar gemi ile birlikte kalmakta, boğaz çıkışında, yani gemi Karadeniz'e ulaştığında gemiden bir bot ile ayrılmakta ve Yenikapı'daki merkeze geri dönmektedir. Bu geri dönüş süreleri Normal dağılıma uymakta ve ortalama 1 saat, standart sapması ise 15 dakikadır.

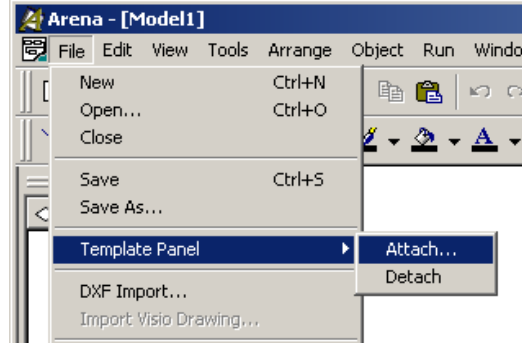
4.3 Senaryolar

Modeli kurmaktaki amacınız, Yenikapı'daki gemi bekleme sürelerini en aza indirecek uygun kılavuz kaptanı sayısını bulmaktır. Bu nedenle modelinizi kılavuz kaptanı sayısı 6, 8, ve 10 için ayrı ayrı çalıştıracak ve bekleme sürelerini gözlemleyeceksiniz.

4.4 Hazırlık

Bu modelin kurulması için öncelikle Arena'nın Advanced Process ve Advanced Transfer template'lerinin eklenmesi gerekmektedir. Aşağıdaki şekilde görüldüğü gibi menüden seçim yapılacak ve bu iki template eklenecektir.

(Template dosyalarının bulunduğu yer
\\Arena_Templates)

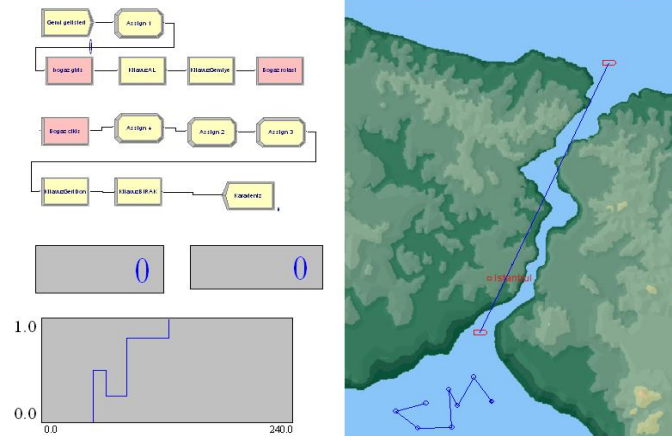


Advanced Transfer template'inden Station ve Route modülleri, Advanced Process'den ise Seize ve Delay modülleri kullanılacaktır.

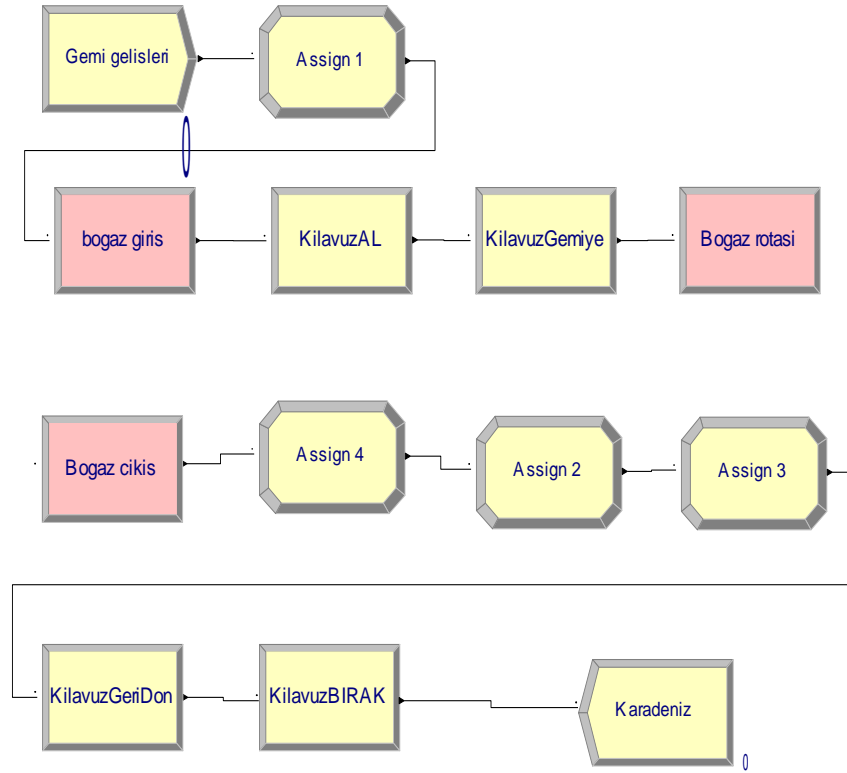
Giriş bölümünde gördüğünüz harita resmini bu dokümandan kopyalayarak (copy) Arena'ya yapıştırdınız (paste).

4.5 Genel görünüm

Modelin bitmiş halinin genel görünümü aşağıdaki şekle benzer olacaktır. Bir sonraki bölümde modul diyagramı açıklanmıştır.



4.6 Modül diyagramı



Modül Adı	Modül Tipi	Template Adı
Gemi gelişleri	Create	Basic Process
Bogaz giris	Station	Advanced Transfer
Kilavuz AL	Seize	Advanced Process
Kilavuz Gemiye	Delay	Advanced Process
Bogaz rotasi	Route	Advanced Transfer
Bogaz cikis	Station	Advanced Transfer
KilavuzGeriDon	Delay	Advanced Process
Kilavuz BIRAK	Release	Advanced Process

4.7 Variables ve Resources

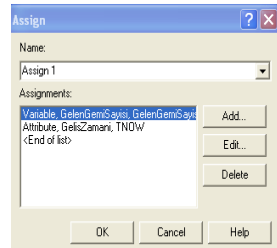
Aşağıdaki değişkenleri Basic Process'deki Variables listesinde tanımlayınız.

KilavuzKaptanSayisi
GelenGemiSayisi
OrtalamaGecisSuresi
ToplamGecisSuresi

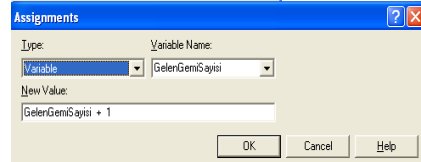
Ayrıca yine Basic Process'deki Resources'da bir kaynak tanımlayınız (KilavuzKaptan). Bunun kapasitesi de her bir senaryonuzda 6, 8 ve

10 olacaktır.

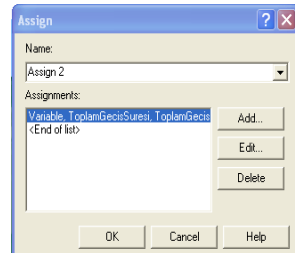
4.8 Assign Modülleri



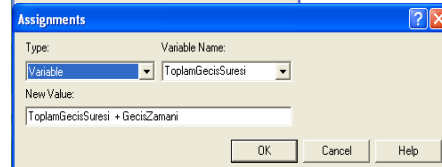
The 'Assign' dialog box for 'Assign 1' shows a list of assignments. The first assignment is 'Variable, GelenGemiSayisi, GelenGemiSayi'. The 'Add...' button is highlighted.



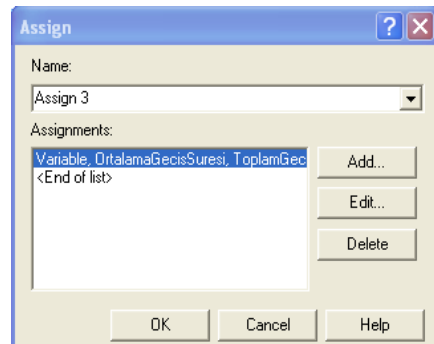
The 'Assignments' dialog box for 'Assign 1' shows the 'Type' set to 'Variable' and the 'Variable Name' set to 'GelenGemiSayisi'. The 'New Value' is 'GelenGemiSayisi + 1'.



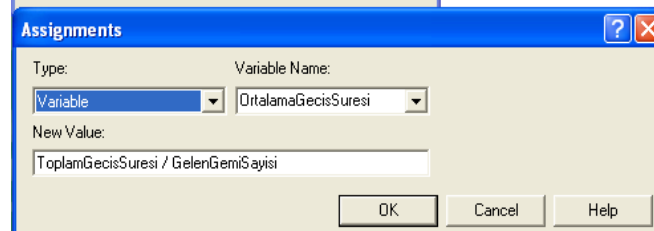
The 'Assign' dialog box for 'Assign 2' shows a list of assignments. The first assignment is 'Variable, ToplamGecisSuresi, ToplamGecis'. The 'Add...' button is highlighted.



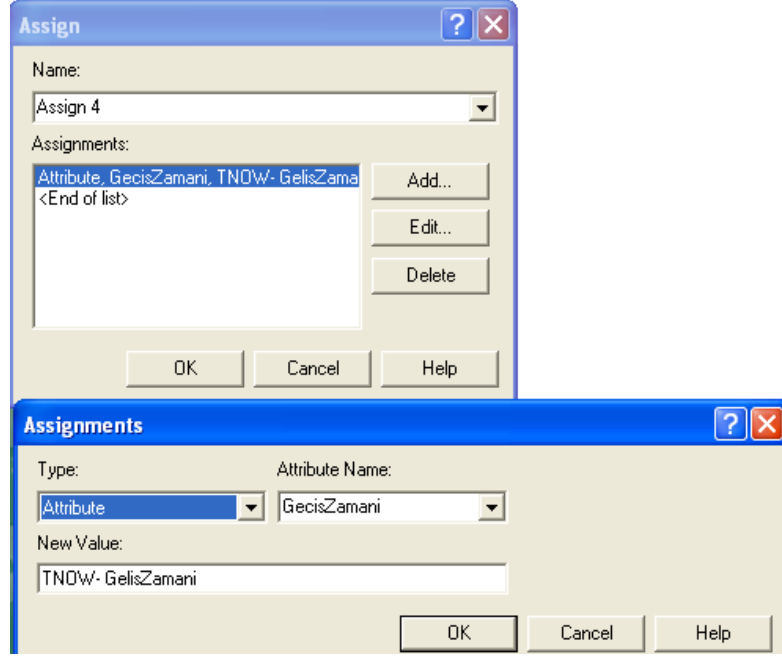
The 'Assignments' dialog box for 'Assign 2' shows the 'Type' set to 'Variable' and the 'Variable Name' set to 'ToplamGecisSuresi'. The 'New Value' is 'ToplamGecisSuresi + GecicZamani'.



The 'Assign' dialog box for 'Assign 3' shows a list of assignments. The first assignment is 'Variable, OrtalamaGecisSuresi, ToplamGecis'. The 'Add...' button is highlighted.



The 'Assignments' dialog box for 'Assign 3' shows the 'Type' set to 'Variable' and the 'Variable Name' set to 'OrtalamaGecisSuresi'. The 'New Value' is 'ToplamGecisSuresi / GelenGemiSayisi'.



4.9 Station'lar

Create modülünde yaratılan Entitilerin ekran üzerinde bir noktadan bir noktaya transferi Station noktaları ile olur. Ekrana bir Station noktası koymak için araç çubuklarından "Animate Transfer"deki Station seçilir. Seçimden sonra Yenikapı ve Karadeniz dolaylarına Station ekleyiniz.

Animate Transfer araç çubuğunu çıkartmak için View ve Toolbars... seçilmelidir.



Daha sonra iki station noktası arasına, yine Animate Transfer araç çubuğundan Route seçilmeli ve bir rota çizilmelidir.

Boğaz giris modülünde station olarak Yenikapı dolaylarına eklediğiniz station'ın ismi seçilmeli, aynı şekilde Boğaz çıkış modülünün station'ı olarak da Karadeniz dolaylarına eklediğiniz station noktası seçilmelidir.

Ayrıca Bogaz Rotası modülünde de Destination Type olarak station seçilmeli ve Karadeniz için oluşturduğunuz Station noktası seçilmelidir.



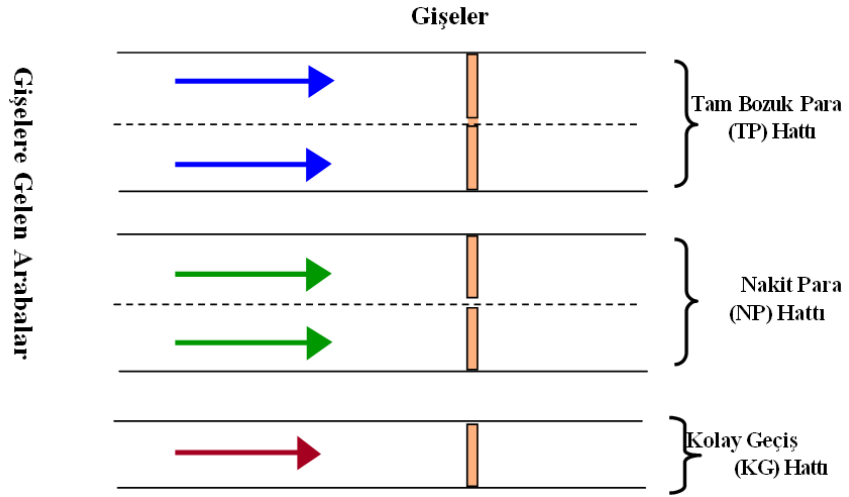
Bu örnekte
Arena'da

- Advanced Transfer: Station Modülü
- Advanced Transfer: Route Modülü
- Advanced: Seize Modülü
- Advanced: Delay Modülü
- Advanced: Release Modülü

5 OTOYOL GIŞELERİ MODELİ*

5.1 Tanım

Bu örnekte¹ New Jersey Turnpike'deki otoyol gişelerden oluşan bir ulaşım sistemini gösterilmekte ve gişe ücretlerinin toplanmasından kaynaklanan kuyruk gecikmeleri üzerine bir çalışma hedeflenmektedir. Şekil 5-1'de çalışma yapacağımız bu sistemi gösterilmektedir.



Şekil 5-1 New Jersey Turnpike'daki Ücret Gişeleri Sistemi

Gişeler; 2 Tam Bozuk Para (TP) Hattı, 2 Nakit Para (NP) Hattı ve 1 Kolay Geçiş (KG) hattından oluşuyor. Gelen araçlar aşağıda 3 grupta sınıflandırılmıştır:

- 1) Tüm gelen arabaların %50'si TP hattına gitmektedir, ve aldıkları servis hizmeti zamanı dağılımı Norm(4.81,1.01)'a göre dir. Unutulmamalıdır ki sadece pozitif değerler geçerlidir.
- 2) Tüm gelen arabaların %30'u NP hattına gitmektedir ve aldıkları servis hizmeti zaman dağılımı $5 + \text{Logn}(4.67,2.26)$ 'a göre dir.
- 3) Tüm gelen arabaların %20'si KG hattına gitmektedir ve aldıkları servis hizmeti zaman dağılımı $1.18 + 4.29 \times \text{Beta}(2.27,3.02)$ 'a göre dir.

İşleri zorlaştırmamak için kuyruklar arasında kaynamalar olmadığını kabul ediyoruz. Yani, bir araba bir kere gişesi karşısındaki sıraya son dan girdiği zaman, asla diğerlerine geçiş yapamaz.

*¹ Bu örnek "Tayfur Altıok ve Benjamin Melamed (2007)'in Simulation Modeling and Analysis with ARENA (Academic Press) kitabından Türkçe'ye çevirilerek alınmıştır

Traffik sıklığı durağan değildir ve günün saatleriyle birlikte geniş bir yelpazede değişmektedir. Beklenildiği gibi trafik, sabah ve akşam telâşeli saatlerde yoğun, talebin az olduğu saatlerde ise seyrek. Aşağıdaki tabloda araç gelişleri arası zaman dağılımını 24 saatlik periyotlar halinde özetlemektedir.

Zaman Periyodu (saat)	Gelişler Arası Zaman Dağılımı (saniye)
0 AM– 6 AM	8 + Gamm (4.4,4.12)
6 AM – 9 AM	Tria (1.32,1.57,1.76)
9 AM – 16 PM	2.64 + Weib (0.82,4.5)
16 PM – 19 PM	Tria (1.32,1.57,1.76)
19 PM – 24 PM	4.2 + Gamm (0.87,8.24)

Faal olan nakit para gişelerinin sayısı zamana göre değişmektedir. Bu tarz gişelerde memur görevlendirilmesi gerektiği, üstelik işletilmesi de pahalı olduğu için talebin az olduğu zamanlarda bir tanesi kapalı tutulmaktadır. Sadece sabah ve akşam telâşeli saatlerde tüm nakit para gişeleri açık tutulmaktadır.

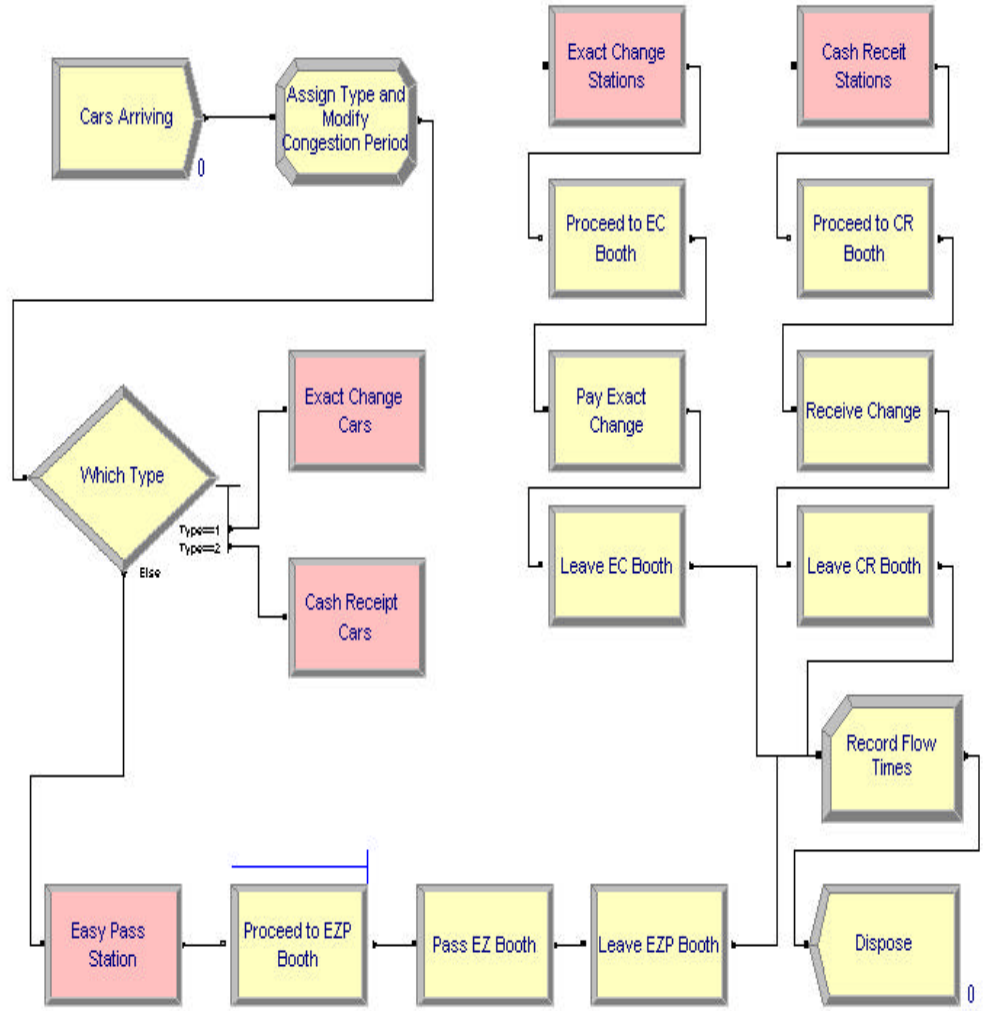
Gişe sistemi için tipik genel performans analiz hedefleri aşağıdaki soruları akla getirmektedir:

1. Ek trafiğin araba gecikmelerindeki etkisi nedir?
2. Başka gişeler açmak bekleme zamanlarını belirgin bir şekilde azaltacak mı?
3. Bazı gişeler hafif trafik saatleri boyunca, bekleme zamanlarını hissedilir derecede artırmadan kapalı tutulabilir mi?
4. Bazı nakit para gişelerini tam bozuk para veya kolay geçiş gişelerine çevirmenin etkisi ne olurdu?
5. Her iki nakit para kabini de her zaman açık tutulacak olursa bekleme süreleri nasıl azalacaktır?

Tabii ki bu gibi ilave fikirler çalışma altındaki belirli gişelere özel olabilir, ama biz bizim durumumuzda, sistem ve kabin kullanımı sırasında ortalama sürelerin performans ölçütlerini kullanarak en son soruya daha fazla önem vermek istiyoruz.

Yukarıdaki ücretli geçiş sistemi için bir Arena modeli süreç yerleşimi Şekil 5-2’de gösterilmiştir.

Şekil 5-2 Ücretli geçiş sistemi için bir Arena modeli



Model şu bileşenlere ayrılabilir: Çeşitli zaman periyotlarında doğru dağılımlarla yayılan araba mevcutları yaratma, en kısa kuyrukla doğru gişeye araba yollanması ve gelen arabalara hizmet.

Bu amaçla paralel mantıkla (çoklu gişeler gibi) modül kümelerinin modellenmesini kolaylaştırmak için **Set** yapısını kullanıyoruz. Sonuç olarak, böyle yapıların sayısı kolaylıkla değiştirilebilir. Sonraki başlıklarda bazı detaylarıyla ve simülasyon sonuçlarıyla birlikte Ücretli geçiş modeli tanımlanacaktır.

5.2 Geliş Üretimi

Araba gelişleri Gelen Arabalar diye adlandırılan Create modülünde üretilmektedir (Şekil 5-3).

Şekil 5-3 Create modülünün iletişim kutusu Gelen Arabalar

Create

Name: Cars Arriving Entity Type: Entity 1

Time Between Arrivals

Type: Expression Expression: Int_Times(k) Units: Seconds

Entities per Arrival: 1 Max Arrivals: Infinite First Creation: 0.0

OK Cancel Help

Burada “Gelişler Arası Zaman” kısmı, araba gelişleri arası zamanlar için zamana bağlı örneklem dağılımlarını “Int_Times(k)” ifadesi aracılığıyla belirtir. “Int_Times” değişkeni örneklem ifadelerinin bir vektörüdür, Şekil 5-4’de tanımı görüntülenmiştir. k bu vektör için 1 ve 5 arasında bir indeksi temsil eden bir değişkendir.

Şekil 5-4 Int_Times Vektöründe zamana bağlı geliş-araları için bir örneklem dağılımını belirten iletişim kutusu

Expression - Advanced Process

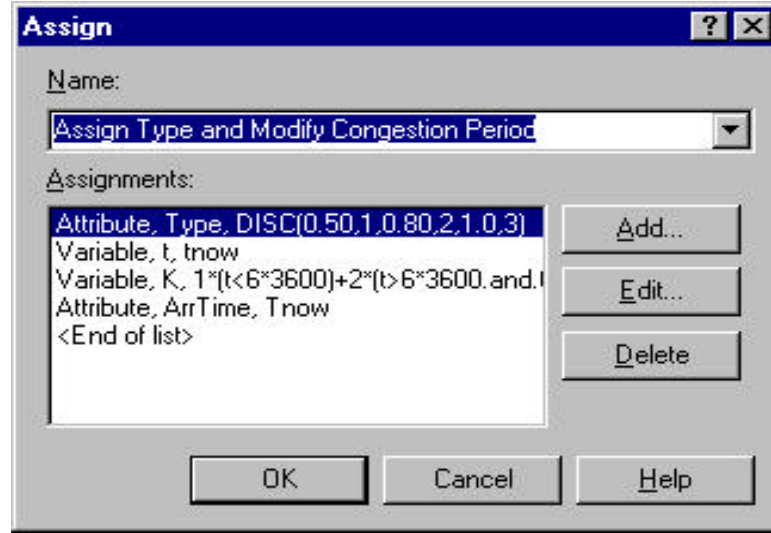
	Name	Row	Columns	Expression Values
1	Int_Times	5		5 rows

Expression Values

	Expression Value
1	8+GAMM(4.4,4.12)
2	TRIA(1.32,1.57,1.76)
3	2.64+WEIB(0.82,4.5)
4	TRIA(1.32,1.57,1.76)
5	4.2+GAMM(0.87,8.24)

Unutmayın ki k ve Int_Times önceki tabloda verilen girdilere karşılık gelir. Gerekli dağılıma ulaşmak için, k değişkeni, geliş gününün zamanıyla (saatiyle) değişen araba miktarlarına göre, uygun olan satır dizinine yazılmıştır. Bu atama; araba oluşturulmasından hemen sonra, araba miktarı Tipi Atama ve Sıkışıklığı Değiştirme Periyodu olarak adlandırılan Ata modülüne girdiğinde meydana gelir. Şekil 5-5’de bunun iletişim kutusu görüntülenmiştir.

Şekil 5-5 Tipi Atama ve Sıkışıklığı Değiştirme periyodu olarak adlandırılan Ata modülünün iletişim kutusu



k'nın ataması ifadeyi kısmen görüntüleyen Atamalar bölümünün üçüncü satırında yapılmıştır (parantez içinde koşullar saniye cinsindedir):

$$\begin{aligned} k = & 1 * (t < 6*3600) + \\ & 2 * (t \geq 6 * 3600 .and. t < 9 * 3600) + \\ & 3 * (t \geq 9 * 3600 .and. t < 16 * 3600) + \\ & 4 * (t \geq 16 * 3600 .and. t < 19*3600) + \\ & 5 * (t \geq 19 * 3600) \end{aligned}$$

Unutmayalım ki yukarıdaki t değişkeni Atamalar bölümünün ikinci satırındaki simülasyonun saat değişkeni Tnow'a ayarlanmıştır. Sonuç olarak, verilen herhangi bir zamanda bir yüklem (yani koşul) kesinlikle doğrudur (1 ile değerlendirilir) ve tüm diğerleri yanlıştır (0 ile değerlendirilir). Böylelikle gerekli satır dizini sağlar. Ek olarak, Atamalar bölümünün ilk satırı, her arabanın tip numarasını gerekli olasılığı kullanarak (Arena DISC dağılımı aracılığıyla) örnekler (1=Tam bozuk para, 2=Nakit para, 3=Kolay geçiş) ve bunu arabanın Tip özelliğine atar. Dördüncü satır arabanın geliş zamanını, Tnow, arabanın ArrTime özelliğinde saklar (ileride sistem zamanı hesaplamada kullanmak için).

5.3 Arabaları Gişelere Gönderme

Atamalar yapıldığında Hangi Tip diye adlandırılan bir araba miktarı Karar modülüne iletirler, bunun iletişim kutusu Şekil 5-6'de gösterilmiştir.

Şekil 5-6 Hangi Tip'in Karar modülünün İletişim Kutusu

The 'Decide' dialog box has a title bar with a question mark and a close button. It contains the following fields and controls:

- Name:** A text box containing 'Which Type'.
- Type:** A dropdown menu currently showing 'N-way by Condition'.
- Conditions:** A list box containing three items: 'Attribute, Type, ==, 1', 'Attribute, Type, ==, 2', and '<End of list>'. To the right of this list are three buttons: 'Add...', 'Edit...', and 'Delete'.
- Buttons:** At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Bu modül arabanın Tip özellikleri ile uyumlu olarak, bir araba miktarını doğru gişeye yollar. İstasyon modülü üzerinden bir konum kavramı (Gelişmiş Transfer şablon panelinden) uygulanmaktadır. Bu tek bir istasyonu (bölgeyi) veya bir dizi istasyonları (çoklu bölgeleri) ifade edebilir. İstasyonların (bölgelerin) her biri modeldeki bir fiziksel veya mantıksal bölgeyi modelleyecektir. Bizim örneğimizde, tek olan Kolay Geçiş gişesi tek bir istasyon modülü tarafından modellenmiştir ve çoklu olan Tam Bozuk Para ve Nakit Para gişeleri de bir dizi istasyon modülleri tarafından modellenmiştir çünkü her bir gişe tipi için mantık aynıdır (benzerdir). Açıklamak için Şekil 5-7'de Tam Bozuk Para gişelerini temsilen hazırlanmış istasyon modülünün iletişim kutusu gösterilmektedir.

Şekil 5-7 Tam Bozuk Para gişeleri için hazırlanan istasyon modülü iletişim kutusu

The 'Station' dialog box has a title bar with a question mark and a close button. It contains the following fields and controls:

- Name:** A text box containing 'Exact Change Stations'.
- Station Type:** A dropdown menu currently showing 'Set'.
- Set Name:** A dropdown menu currently showing 'EC Stations'.
- Save Attribute:** A dropdown menu currently showing 'Current_Booth'.
- Station Set Members:** A list box containing three items: 'EC_1', 'EC_2', and '<End of list>'. To the right of this list are three buttons: 'Add...', 'Edit...', and 'Delete'.
- Buttons:** At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

Bu amaçla, "İstasyon Tipi" alanından set opsiyonu seçilmiştir (tek bir İstasyon modülü tanımlayan İstasyon opsiyonuna karşıt olarak).

Modelleyici sonra da “İstasyon Dizi Üyeleri” alanından sağındaki 3 düğme yardımıyla dizi üye isimlerini sınıflandırır. Burada, iki Tam Bozuk Para gişesi TBP_1 ve TBP_2 diye isimlendirilmiştir (aynı mantıkla, NP gişeleri de kendi istasyon modül ayarlamalarında NP_1 ve NP_2 olarak isimlendirilmiştir). Bu istasyon modül ayarları isim belirle alanında belirlenir. İstasyon modül ayarlarının belli bir üyesine bir miktar girdiğinde “Özelliği Sakla” alanı özellik isimlerini hangi istasyon ID’sini depolayacaklarına göre sınıflandırmak için kullanılır. Bu ID sadece “İstasyon Belirleme Üyeleri” alan listesinin içinde olan gerekli istasyon isminin sırasındır (pozisyonudur). Bizim durumumuzda, İstasyon modülü gelen araba miktarının Geçerli_Gişe’sine bağlanmak için başka bir yere atanır. Bu ID’yi saklamak, ortak kaynak, kuyruk vb. indekslemekte kullanılabileceği için faydalı olabilir.

Ücret gişeleri İstasyon modülleri tarafından modellenirken, arabaları gişelere tiplerine göre gönderme işi bir “İstasyonSeç” modülünde yapılır (İleri Transfer şablon panelinden). Buna göre modül Hangi Tip, tip1 (Tam Bozuk Para), tip2 (Nakit Para) ve tip3 (Kolay Geçiş) arabalarını sırasıyla, İstasyonSeç modülleri Tam Bozuk Para Arabaları, Nakit Para Arabaları , Kolay Geçiş İstasyonu’na, tip özelliklerine göre, gönderir. İstasyonSeç modülüne giren her araba miktar’ı bir grup istasyon (gişe) arasından bir istasyon (gişe) seçer. TBP arabaları (tip 1) için hazırlanan Şekil 5-8’deki iletişim kutusunda temsilen gösterildiği gibi.

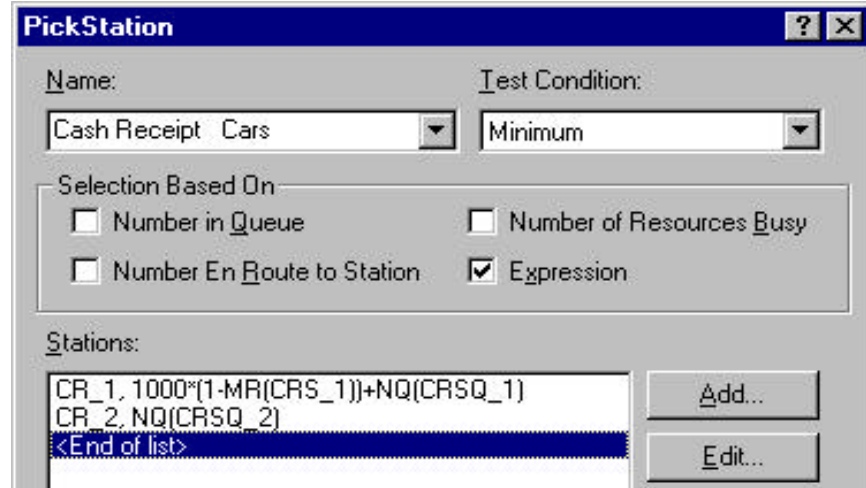
Şekil 5-8 İstasyonSeç modülü TBP arabaları’nın iletişim kutusu

The screenshot shows the 'PickStation' dialog box. The 'Name' field is set to 'Exact Change Cars' and the 'Test Condition' is set to 'Minimum'. Under 'Selection Based On', the 'Number in Queue' checkbox is checked. The 'Stations' list contains 'EC_1, ECSQ_1', 'EC_2, ECSQ_2', and '<End of list>'. The 'Transfer Type' is set to 'Route', 'Route Time' is '0.0', and 'Units' is 'Seconds'. The dialog has 'OK', 'Cancel', and 'Help' buttons at the bottom.

Burada modelci tarafından, İstasyon modüllerinin listesi ve sağındaki 3 düğme yardımıyla içeriden seçilmek üzere İstasyon alanı içerisinde kuyruklar kurulmuştur. Her bir giriş bir İstasyon modülü (NP_1 ve NP_2'nin bu iki NP gişelerini temsil ettiğini hatırlayın) ve ilgili kuyruğu (NPSQ_1 ve NPSQ_2'nin NP gişeleri içindeki kuyrukları temsil ettiğini hatırlayın) belirler. Deneme Durumu alanı içerisinde seçili Minimum şıkkı ve Dayanılan Seçim bölümü içinde tik atılmış Kuyruktaki Numara şıkkı, araba Mevcutlarını en kısa kuyruklu bir gişe seçmek üzere ayarlanmıştır. Modelci alternatif olarak Maximum şıkkını özelleştirebilir ve gişe seçimini diğer miktarlara dayandırabilir (keyfi bir ifade de dahil olmak üzere). Birbirlerine aynı anda geçiş ihtimaline karşı, duruma yeterli gelecek ilk gişe seçilir. Modelciye "Rota Zamanı" ve ünitler alanı içinde yolculuk zamanını ayarlayabilme imkanı sağlaması için Transfer Tipi alanı Rota seçeneğine göre ayarlanmıştır. Diğer Transfer Tipi seçenekleri Bağlantı (sıradan bir mantıksal bağlantı), Taşıma (taşıyıcıları modellemek için, kamyon veya forkliftler gibi) ve Convey (conveyorları modellemek için). Taşıyıcılar ayrı bir örnekte incelenecektir.

NP arabaları (tip 2) ilgili İstasyonSeç modülü Nakit Para Arabaları'nda farklı bir işlem gerektirir. Şekil 5-9'da bunun iletişim kutusu gösterilmektedir.

Şekil 5-9 İstasyonSeç modülü Nakit Para Arabaları'nın iletişim kutusu



Biz tekrar Deneme Durumu alanından Minimum seçeneğini seçeriz. Fakat hatırlanmalıdır ki, TP gişeleri'nden farklı olarak, NP gişelerinden biri (diyelim ki, NP_2) her zaman açıkken, diğeri (diyelim ki, NP_1) off-peak saatlerinde kapalıdır. Bu olayı açıklamak için "Dayanılan Seçim" bölümünden ifade seçeneğini seçiyoruz ve İstasyonlar alanındaki her bir NP gişesi için ayrı bir ifade belirliyoruz. Kaynak NPS_1 ve kuyruk NPSQ_1'in İstasyon modülü olan NP_1 ile ilişkili olmasını sağlayın ve benzer olarak Kaynak NPS_2 ve kuyruk NPSQ_2'nin de İstasyon modülü NP_2 ile ilişkili olmasını sağlayın.

Son olarak da, unutmayalım ki MR(R) ve NQ(Q), kaynak R'de kaynak ünitesi sayısını ve kaynak Q'da Miktarlar'ın sayısını sırasıyla döndüren Arena fonksiyonlarıdır ve farzedelim ki NPSQ_2 1000'den az arabayı kuyruğa alabiliyor (aksi takdirde var olan yeterince büyük bir sayı ile 1000 de değiştirilebilir...). Şimdi İstasyonlar alanı içindeki ifadelerin doğru bir şekilde belirlendiği konusunda emin olmak için, sıradaki, NP_1 için olan ifadelerle ilgili iki durumu inceleyin:

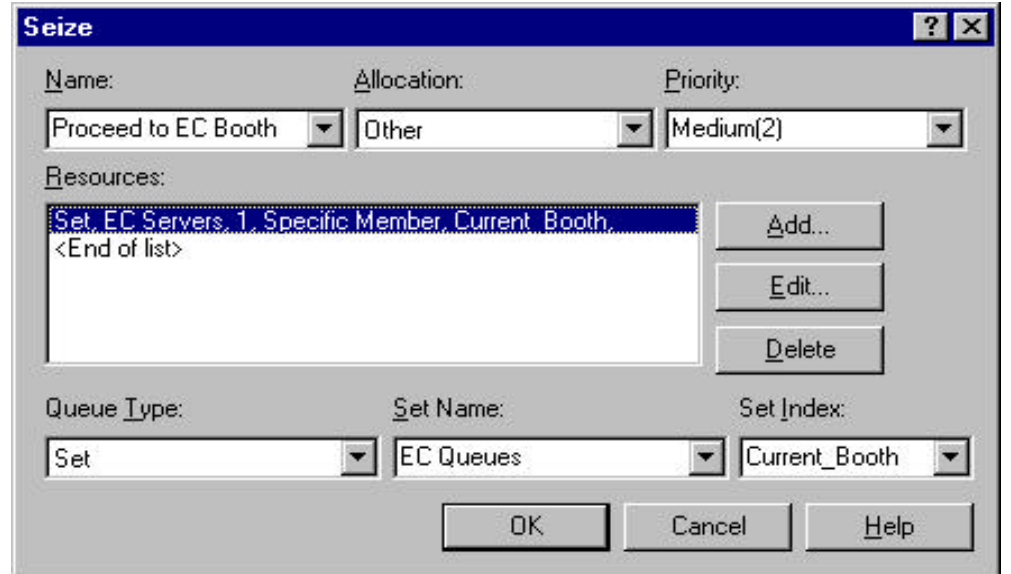
1. NP_1 gişesi açık ise, $1-MR(NPSQ_1) = 0$ 'dır, ve NP_1 için olan ifade $NQ(NPSQ_1)$ olarak değerlendirilir. Gişe bölümü, gerekirse, $NQ(NPSQ_1)$ ve $NQ(NPSQ_2)$ 'nin minimumuna dayandırılır.

2. Diğer taraftan, NP_1 gişesi kapalı ise, $1-MR(NPSQ_1) = 1$ ve $NQ(NPSQ_1) = 0$ 'dır ve NP_1 için olan ifade 1000 olarak değerlendirilir. Bu numara $NQ(NPSQ_2)$ 'ye ulaştığı varsayımından dolayı gerektiğinde kesinlikle NP_2 gişesi seçilecektir.

5.4 Arabalara Gişelerde Hizmet Vermek

Bir kere bir araba miktar'ı kendi gişesine gönderildiğinde, (muhtemelen) ilgili kaynak kuyruğunda sıraya girer ve neticede ücreti ödeyip gişelerden ayrılır. Özellikle İstasyonSeç modülü olan Tam Bozuk Para İstasyonları'ndan çıkan arabalar TP Gişesine İllerle diye adlandırılmış "Yakala" modülüne girerler. Bu modülün iletişim kutusu Şekil 5-10'da gösterilmiştir.

Şekil 5-10 TP Gişesine İllerle diye adlandırılmış Yakala modülünün iletişim kutusu

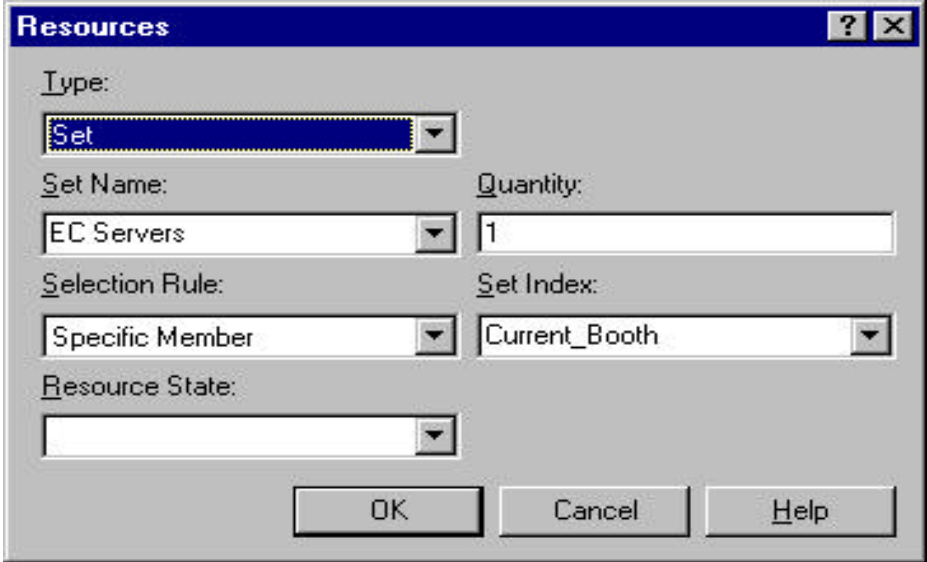


Burada araba miktar'ı ilgili İstasyon modülünün kaynağı (gişenin sunucusu)'nu yakalamaya çalışır. Ama bu kaynağı göstermek için Kaynaklar alanından istasyon (gişe) grubuna karşılık gelen bir grup kaynak(sunucu) atamamız gereklidir. Örneğin, her biri ayrı ayrı TP

gişeleri için, TP_1 ve TP_2'deki kaynakların isimleri TPS_1 ve TPS_2 olur, ve ilgili kuyruklar da TPSQ_1 ve TPSQ_2 diye adlandırılır.

Bu yapıları belirlemek için önce gerekli gruplar kurulur. Daha spesifik olarak, bir grup kuyrukları oluşturmak için Kuyruk Tipi alanı içerisinde "Set" seçeneği seçilir ve Set "Name" alanına ismi girilir. Sonra Şekil 5-11'de Kaynaklar iletişim kutusunda gösterilen ve Ekle butonu kullanılarak ekrana gelen gerekli kaynaklar set oluşturulur.

Şekil 5-11 Kaynak set TP Sunucuları'nın İletişim Kutusu



The screenshot shows a dialog box titled "Resources". It has a title bar with a question mark and a close button. The dialog contains the following fields and controls:

- Type:** A dropdown menu with "Set" selected.
- Set Name:** A text box containing "EC Servers".
- Quantity:** A text box containing "1".
- Selection Rule:** A dropdown menu with "Specific Member" selected.
- Set Index:** A dropdown menu with "Current_Booth" selected.
- Resource State:** An empty text box.
- At the bottom, there are three buttons: "OK", "Cancel", and "Help".

Burada, Tip alanından Set seçeneği ilk olarak seçilir, ve Set Name alanına ismi girilir. Sonra, Seçim Kuralı alanından Spesifik Üye seçeneği seçilir, ve Set İndeks alanından özellik ismi Geçerli_Gişe girilir (Unutmayın ki ikincisi doğru kaynağın ID'sini elinde bulundurmaya üzere tutacaktır). Öteki seçim kuralları rasgele seçeneğini ve konjonktürel seçeneği içinde tutar.

TP Sunucuları set'ni kaynak üyelerini oluşturmak için, modelci Temel İşlem (Basic Process) şablon panelindeki Set modülünü Şekil 5-12'nin altındaki tabloyu ekrana çıkarmak için kullanır.

Şekil 5-12 Set modülündeki (bir altta) setlerin tablo görünümü ve kaynak set TP Sunucuların üyeleri (iki altta)



The screenshot shows a dialog box titled "Members". It contains a table with the following data:

	Resource Name
1	ECS_1
2	ECS_2

Below the table, there is a text box with the instruction: "Double-click here to add a new row."

Set - Basic Process			
	Name	Type	Members
1	EC Servers	Resource	2 rows
2	Flow Times	Tally	3 rows
3	CR Servers	Resource	2 rows

Daha spesifik olarak, ayrı bir set ile ilişkilenen her bir dizi ile birlikte öncelikle tabandaki tablo görünür. Genel olarak, eğer gerekli set henüz oluşturulmamışsa, o zaman yeni bir dizi eklemek için *"Double-click here"* yazan metin mesajına çift tıklanarak tablo içinde yeni bir dizi olarak oluşturulabilir. Yeni set İsim sütununda isimlendirilir ve tipi de Tip kolonundan seçilir. Set bir kez oluşturulup tablo dizisi olarak görüldüğünde modelci Üyeler kolonunun altındaki butona tıklar ve Şekil 5-12'in üst tarafındaki iletişim kutusu çıkar. Sonra yeni set Üyeleri de set eklemeye benzer olarak üst tablodaki diziler olarak eklenir.

Gelişmiş Set modülünün (Gelişmiş işlem şablon panelinden) Set modülü yerine kullanılması haricinde, Kuyruk setlerinin oluşturulması ve üyeleri benzerdir (analojiktir). Önceden Geçerli Gişe özelliğinin ilgili gişe kuyruğunun ID'sini tuttuğu yer olarak Şekil 5-12'nin alt bölgesinde açıklandığı üzere, Şekil 5-13 Gelişmiş Set modülünün (şekilde altta) ve kuyruk seti TP Kuyrukları (şekilde üstte) üyelerinin tablo görüntülerini gösterir.

Şekil 5-13 Gelişmiş Set modulündeki setlerin (bir altta) ve kuyruk seti TP Kuyrukları'nın (iki altta) üyelerinin Tablo görüntüsü

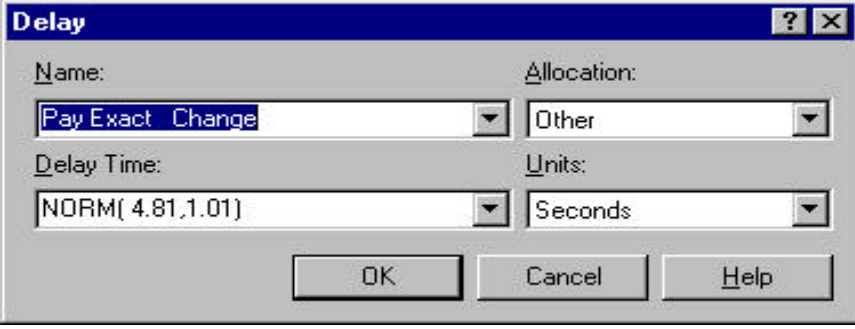
Members	
	Queue Name
1	ECSQ_1
2	ECSQ_2

Double-click here to add a new row.

Advanced Set - Advanced Process			
	Name	Set Type	Members
1	EC Queues	Queue	2 rows
2	CR Queues	Queue	2 rows

Bir kere bir araba kendi sunucu kaynağını (gişe) yakalamayı başardığı zaman, ücret ödeme işlemi süreci için yolunun üzerindeki bir Erteleme modülüne girer. Örnek olarak, Şekil 5-14 TP arabalarının ücret ödeme işlemi amaçlı olan Erteleme modülünün iletişim kutusunu göstermektedir.

Şekil 5-14 Erteleme modülü Tam Para Öde'nin iletişim kutusu

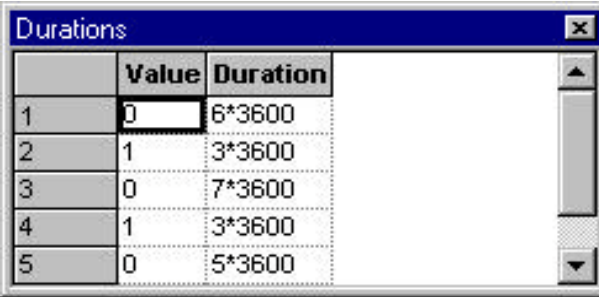


The image shows a dialog box titled "Delay". It has a blue title bar with a question mark and a close button. The dialog contains the following fields and buttons:

- Name:** A dropdown menu with "Pay Exact Change" selected.
- Allocation:** A dropdown menu with "Other" selected.
- Delay Time:** A dropdown menu with "NORM(4.81.1.01)" selected.
- Units:** A dropdown menu with "Seconds" selected.
- Buttons:** "OK", "Cancel", and "Help" buttons are located at the bottom.

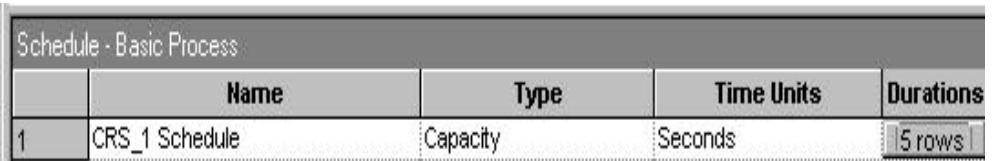
Hatırlanmalıdır ki bir NP gişesi (özellikle, CR_1) talebin daha az olduğu saatlerde kapalıdır. Bu taktik Çizelgeleme modülü kullanılarak ilgili tablo görüntüsünü ekrana çıkarmak için Temel İşlem şablon panelinden uygulanır. Şekil 5-15'de tablo gösterilmiştir.

Şekil 5-15 Çizelgeleme modülündeki (iki altta) kapasite çizelgelerinin tablo görünümü ve gişe NP_1'deki NPS_1 sunucusu ilgili süreçler (bir altta)



The image shows a dialog box titled "Durations". It contains a table with the following data:

	Value	Duration
1	0	6*3600
2	1	3*3600
3	0	7*3600
4	1	3*3600
5	0	5*3600

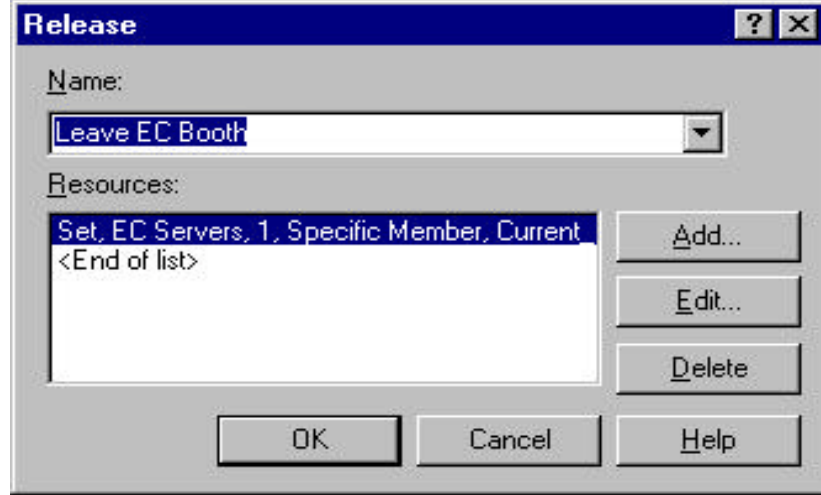


The image shows a table titled "Schedule - Basic Process". It has the following columns and data:

	Name	Type	Time Units	Durations
1	CRS_1 Schedule	Capacity	Seconds	5 rows

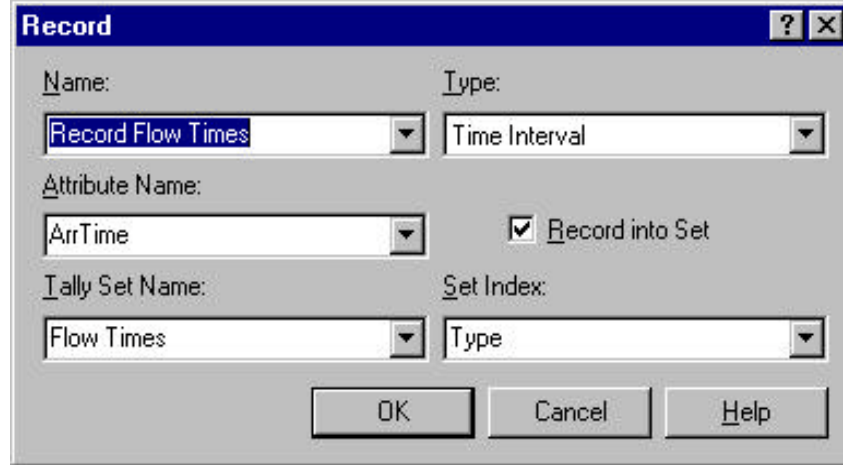
Plan NPS_1 Plan ile kaynak NP_1 arasındaki ilişki, Tip sütunundan Plana Dayalı seçeneğini seçerek ve NPS_1 Plan ismini Plan İsmi kolonuna girerek, Kaynak modülünde (gösterilmedi) belirlenir. Ödemenin tamamlanmasıyla, (kaynak setine indekslemek için tekrar Geçerli_Gişe özelliğini kullanarak) her araba kendi yolunda gişe kaynağından serbest bırakılmak için ilgili Salım (Release) modülüne ilerler. Örnek olarak, Şekil 5-16 TP arabaları için Salım modülünün iletişim kutusunu göstermektedir.

Şekil 5-16 Salım modülü TP Gişesinden Ayrıl'ın iletişim kutusu



Bu noktada araba para ödeme işlemini henüz tamamlamış ve (imha diye çağırılan) imha modülüne çıkarılmadan önce modelde tek önemli görevi kendi akış sürelerinin kaydını tutmaktır. Bu amaçla, her araba miktarı, Kayıt Akış Zamanları diye adlandırılan Kayıt modülüne girer. Bunun da iletişim kutusu Şekil 5-17'de gösterilmiştir.

Şekil 5-17 Dialog box of Record module Record Flow Times



Burada Tip alanı; geçmiş zamanı, Özellik Adı alanında belirlenen arabanın özellikleri'nde depolanan değerden kaydetme işini gören Zaman Aralıkları seçeneğine göre ayarlanmıştır. Bu özellik (ArrTime) önceden gişelerde arabanın varış zamanını kaydetmek üzere ayarlanmıştı, bu yüzden gerekli akış süreleri iyi bir şekilde toplandı.

Unutmayalım ki, tik kutusu Set'e Kaydet Şekil 5-17'de işaretlendi. Ayrıca şunu da kaydedelim ki Set isimlerini Tut alanında bir isim belirlenmiştir (Akış Zamanları), ve seçilen Set İndeksi alanı seçeneği arabanın Tip özelliğidir. Bu bilgi akış zamanlarını ayrı ayrı araba tipine göre kaydetmek için ayarlanır: TP, NP ve KG. Araba tipine göre sistem akış-zaman derlemesini yönetmek üzere üçlü bir çetele

istatistiği grubu kullanmaktadır.

Şekil 5-2'deki ücret gişeleri modelinin incelenmesi sonucunda, ciddi bir benzerlik ve üç tipdeki arabaların mantıksal sıralamasında küçük farklılıklar olduğunu görülmektedir. Sadece tek bir KG gişesi olduğu için, Model KG arabaları hariç hem TP hem NP arabaları için İstasyonSeç modüllerini kullanmaktadır. Bu farklılık modeli kolaylaştırmasına rağmen, aynı zamanda da modeli temel olarak tek bir KG gişesinde kısıtlar. TP ve NP yolları mantığını KG yolu için de tekrarlamak suretiyle modeli daha genel yapıda tasarlanmanın ciddi manada faydaları bulunmaktadır. Eklenen genellikler analiste kolaylıkla KG gişelerinin sayısını değiştirme imkanı sağlar, böylelikle modelin deneysel aralığını da genişletir.

5.5 Ücret Gişesi Modeli Benzetim Sonuçları

5.5.1 Temel Model

Ücret gişesi modeli $24 \times 60 \times 60 = 86,400$ saniye (bir gün) için çalıştırılmıştır ve aşağıdaki sonuçlar elde edilmiştir.

Kuyruk bölümü her 5 gişe kuyruğu için (2 TP kuyruğu, 2 NP kuyruğu ve 1 KG kuyruğu) gecikme istatistiklerini gösterir. Beklendiği gibi, NP kuyrukları en uzun kuyruklara sahiptir, TP kuyrukları daha kısa gecikmeler ve KG kuyruğu ise küçük gecikmelere sahiptir. Görülmelidir ki NPSQ_1 çok büyük bir gecikmeye sahiptir, çünkü bizim modelimiz NP_1 gişesini "talebin az olduğu periyod"un başlangıcında kapatmakta ve bir sonraki yoğunluk saatinin başlangıcına kadar kuyruğundaki arabaları zamanda "karaya oturmuş" şekilde orada beklemeye bırakmaktadır. Tabiki gerçekçi bir modelde, modelci modele NP_1 gişesini kapatmadan önce NPSQ_1 kuyruğunu bitirmesi için bir mantık eklemelidir.

9:37:44PM

Category Overview

September 26, 2000

Toll Booth

Replications: 1 Time Units: Seconds

Queue

Time

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
CRSQ_1	71.8380	94.00702	0.00	25243.36
CRSQ_2	16.7822	(Correlated)	0.00	94.5460
ECSQ_1	2.0543	(Correlated)	0.00	19.9613
ECSQ_2	1.7918	0.244268160	0.00	16.4772
EZP Queue	0.3657	(Correlated)	0.00	10.3250

Kaynak bölümü her 5 gişe sunucusu için sunucu kullanımlarını göstermektedir. NP_2 gişesi her zaman açık bulunurken NP_1 gişesi "talebin az olduğu periyod" boyunca kapalı olduğundan dolayı, sunucu NPS_2'nin kullanımı belirgin bir şekilde sunucu NPS_1'den daha yüksektir. Üstelik, gişe TP_1 , TP_2 gişesini ikisi de yoğun değilken üzerine aldığı için (bu, ilgili İstasyonSeç modülünde ilk durak olarak listelenmiştir), sunucu TPS_1 TPS_2'den daha yüksek bir kullanıma sahiptir.

Resource

Usage

Number Busy	Average	Half Width	Minimum Value	Maximum Value
CRS_1	0.2463	(Correlated)	0.00	1.0000
CRS_2	0.5513	(Correlated)	0.00	1.0000
ECS_1	0.5067	(Correlated)	0.00	1.0000
ECS_2	0.1614	(Correlated)	0.00	1.0000
EZPS_1	0.1640	(Correlated)	0.00	1.0000

Kullanıcı Tanımlı bölüm araba tipine göre araba akış zamanlarının kayıt sonuçlarını özetler. Genellikle NP_1 gişesinin yoğun olmayan saatlerdeki kapalılığından dolayı, NP tipinin akış zamanları diğerlerinden çok daha uzundur.

User Specified

Other

None	Average	Half Width	Minimum Value	Maximum Value
CR Flow Times	43.4896	29.88354	5.8445	25253.19
EC Flow Times	6.8030	(Correlated)	0.8936	24.8150
EZP Flow Times	3.3894	(Correlated)	1.1825	14.7619

5.5.2 Geliştirilmiş Model

NP_1 gişesinin yoğunluk saatleri dışındaki kapalılığının dramatik etkisi akıllara doğal bir soru getiriyor: Eğer NP_1 gişesi sürekli açık bulundursaydı, NP arabalarının akış zamanları ne ölçüde geliştirilebilirdi? Ücret gişeleri Arena modelini şu geliştirilmiş işletme kuralını yansıtaçak şekilde modifiye etmek kolaydır: Her seferde sunucu kaynağı NPS_1'in kapasitesini yalnızca 1'e ayarla. Geliştirilmiş ücret gişeleri modelinin simülasyon sonuçları ise aşağıda sunulmuştur.

Queue

Time

Waiting Time	Average	Half Width	Minimum Value	Maximum Value
CRSQ_1	14.6785	(Correlated)	0.00	130.01
CRSQ_2	22.7045	(Correlated)	0.00	126.81
ECSQ_1	2.0856	(Correlated)	0.00	19.9613
ECSQ_2	1.9014	0.280898281	0.00	16.4772
EZP Queue	0.3365	(Correlated)	0.00	8.9526

Resource

Usage

Number Busy	Average	Half Width	Minimum Value	Maximum Value
CRS_1	0.5360	(Correlated)	0.00	1.0000
CRS_2	0.2672	(Correlated)	0.00	1.0000
ECS_1	0.5043	(Correlated)	0.00	1.0000
ECS_2	0.1607	(Correlated)	0.00	1.0000
EZPS_1	0.1626	(Correlated)	0.00	1.0000

User Specified

Other

None	Average	Half Width	Minimum Value	Maximum Value
CR Flow Times	27.0212	(Correlated)	5.7100	139.10
EC Flow Times	6.8497	(Correlated)	1.2343	24.8150
EZP Flow Times	3.3520	(Correlated)	1.1819	12.0999

Beklenildiği üzere, geliştirilmiş ücret gişeleri önemli ölçüde daha düşük ortalama akış zamanı, NP arabaları için azami akış zamanı ve daha yüksek oranda gişe kullanımı vermektedir. TP ve KG arabalarının gişelerin istatistiklerindeki küçük sapmalar basit olarak rastgelelikten dolayıdır. Çünkü farklı tiplerin arabaları hatlar arası makaslamalar olmadığı sürece bir biriyle etkileşim göstermezler.



Bu örnekte
Arena'da

- Advanced: Expression Modülü
- Advanced: Advanced Set Modülü
- Delay
- Station
- Pick Station
- Seize
- Resources
- Schedule
- Release
- Record