

Kırıkkale Üniversitesi

# BENZETİM DİLLERİ KOMUTLAR RAPORU

**Benzetim dersi kapsamında öğrenilen komutları açıklayan kapsamlı bir rapordur.**

Mehmet PINARBAŞI

Syf 1

*Ahmet Kürsad TÜRKER*

[2007]

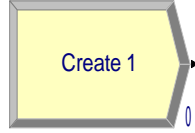
# İÇİNDEKİLER

1. ARENA KOMUTLARI.....	3
1.1. CREATE.....	3
1.2. DISPOSE.....	3
1.3. PROCESS.....	4
1.4. DECIDE .....	5
1.5. BATCH.....	8
1.6. SEPARATE.....	9
1.7. ASSIGN.....	10
1.8. RECORD.....	11
1.9. HOLD.....	11
1.10. MATCH.....	12
1.11. ROUTE.....	13
1.12. STATION.....	14
1.13. ACCESS.....	14
1.14. CONVEY .....	15
1.15. EXIT.....	16
1.16. REQUEST.....	17
1.17. TRANSPORT.....	17
1.18. FREE .....	18
1.19. SEQUENCE .....	19
2. ÖRNEKLER.....	19
2.1. BERBER ÖRNEĞİ.....	19
2.2. GELİŞTİRİLMİŞ BERBER ÖRNEĞİ .....	19
2.3. BEKLEME YERİ SINIRLI BERBER ÖRNEĞİ.....	20
2.4. ATÖLYE ÖRNEĞİ .....	22
2.5. TEKRAR İŞLEMELİ ATÖLYE ÖRNEĞİ.....	22
2.6. BEKLEME YERİ SINIRLI ATÖLYE ÖRNEĞİ .....	24
2.7. DECIDE ÖRNEĞİ.....	26
2.8. ROUTE ÖRNEĞİ.....	28
2.9. TRANSPORT ÖRNEĞİ .....	30
2.10. AKIŞLARI FARKLI ÜRÜNLER İÇİN ÖRNEK .....	32
2.11. ÖRNEK.....	34
2.12. ANİMASYON ÖRNEĞİ .....	36
2.13. BANKA ÖRNEĞİ .....	38
2.14. FABRİKA ÖRNEĞİ.....	40
2.15. ÖRNEK FİNAL SORUSU .....	42
3. SONUÇ .....	48

# 1. ARENA KOMUTLARI

## 1.1. CREATE

Bu modül, bir simülasyon modelinde varlıklar için başlangıç noktası tasarlar. Varlıklar, bir program kullanılarak ya da gelişler arası zamana dayanarak oluşturulur. Varlıklar, sistem içinden sürecin başlamasından sonra modülden ayrılırlar. Bu varlık tipi bu modülde belirtilmiştir. Simgesi aşağıdaki gibidir:



Varlık ismi yazılır.

Gelişler arası zaman değeri yazılır.

Üretilen varlığın tipinin adı.

İlk oluş zamanı ve geliş zamanları için kullanılan zaman birimi.

Üretilen geliş akımının tipi.

Sistem içine ilk varlığın gelmeye başlama zamanı.

Her gelişle verilen bir zamanda sisteme girecek varlık sayısı.

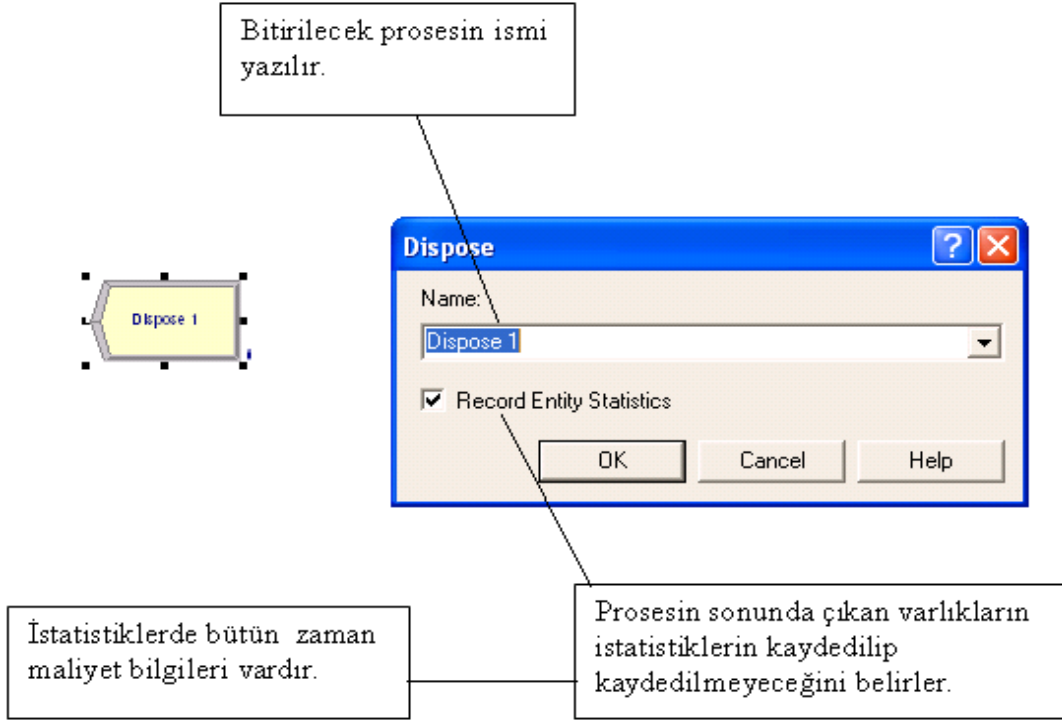
Bu modülde üretilecek max varlık sayısı. Bu değere ulaşıldığında, bu modül ile yeni varlık yaratılmayacak.

OK Cancel Help

Syf 3

## 1.2. DİSPOSE

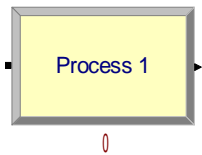
Bu modül, bir simülasyon modelinde varlıklar için son noktayı tasarlar. Varlıklar dağıtılmadan (çıkarılmadan) önce varlık istatistikleri kayıt edilebilir. Ayrıca modülün içinde sistemden ayrılan varlık sayıları da gösterilir. Simgesi ve modül ile ilgili açıklamalar aşağıdadır.



### 1.3. PROCESS

Bu modül, simülasyonda ana proses metodunu tasarlar. Kaynak kısıtlarını tutmak ve bırakmak için opsiyonlar (seçenekler) kullanılır. Ayrıca, bir submodel (alt model) ve kullanıcı tarafından tanımlanmış hiyerarşik mantık kullanım seçeneği vardır. Proses zamanı, varlıkları ayırt eder ve değer eklenen, değer eklenmeyen, taşıma, bekleme ve diğerlerini dikkate alabilir. İşlem önceliğini dikkate alır ve buna göre işlemleri kuyrukta bekletir. Simge üzerindeki çizgi kuyruğu ifade eder.

Simgesi ve açıklamalar aşağıdadır.



İşlem ismi yazılır.

Model tipini belirler. Standart veya alt model.

Belirlenmiş kaynak(lar) için bekleyen varlıkların öncelik değeri..

Modül içinde gerçekleşecek proses tipi. Gezen birimin sadece bekletilmesi, işlem görüp bekletilmesi, alınıp bir süreçten sonra bırakılması veya bekletilip bırakılması tipleri.

Varlık prosesi için kullanılan kaynak ya da kaynak setleri listesi.

Varlıklara proses zaman ve maliyetinin nasıl atanacağına karar verir.

Gecikme parametrelerini belirten dağılım tipi ya da metodu.

Proses için, rapor veritabanında istatistik birikip birikmeyeceği belirlenir.

Uniform ya da Triangular dağılımın her hangi biri için min değer belirlenmesi parametre alanı.

Gecikme parametreleri için zaman birimleri.

Normal dağılım için ortalama, bir sabit zaman gecikmesi için değer ya da Triangular dağılım için mod parametre alanı.

Uniform ya da Triangular dağılımın her hangi biri için max değer belirlenmesi parametre alanı.

#### 1.4. DECİDE

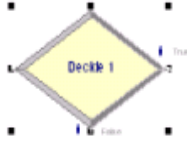
Bu modül sistemde karar verme prosesi için izin verir. Karar alınmasında bir veya daha fazla duruma(koşul) yada bir veya daha fazla olasılığa dayanarak seçmeyi içerir. Durumlar, özellik değerlerine, değişken değerlerine, varlık tiplerine yada bir ifadeye dayanabilir.

2'li ihtimal yada 2'li durumdan herhangi biri seçildiğinde, Decide modülünün iki çıkış noktası vardır. Doğru ve yanlışlar için birer çıkış noktası vardır. N'li ihtimal yada durum tipi belirlendiğinde, çoklu çıkış noktaları, her durum veya olasılık ve tek "else" çıkışı için gösterilir. 2'li ihtimal ve durum için yalnızca, her tipten (doğru/yanlış) çıkan varlıkların sayısı gösterilir. Simgesi ve açıklamaları aşağıdadır.

Seçim yapılacak durumun ismi.  
Örneğin; memnun olmuş müşteri.

Bu seçeneklerde:  
2-way by Condition: 2'li durum  
2-way by Change: 2'li seçim  
N-way by Condition: Çoklu durum  
N-way by Change: Çoklu seçim

Bu seçenekleri  
tektek açılacağız.



Decide

Name: Decide 1 Type: 2-way by Chance

Percent True (0-100): 50 %

OK Cancel Help

Kabul (true) seçeneğinin  
olasılık değeri.

Şimdi bu TYPE seçeneklerini açıklayalım:

TYPE: 2-way by Change = 2'li seçim.

Kabul (true) seçeneğinin  
olasılık değeri.

İkili durumlarda karar vermek için  
seçilir. (örneğin %90 sağlam % 10  
hatalı parçalar için.)

Decide

Name: Decide 1 Type: 2-way by Chance

Percent True (0-100): 90 %

TYPE: 2-way by Condition = **2'li Durum**

İkili durumlarda karar vermek için seçilir.

The screenshot shows the 'Decide' dialog box with the following fields:

- Name: Decide 1
- Type: 2-way by Condition
- If: Entity Type
- Named: çivi

Buttons: OK, Cancel, Help

If: seçim kısmında  
Variable: Değişkene göre  
Attribute: Varlık tipine göre  
Entity Type: isme göre örn: çivata  
Expression: Açıklamaya göre

Syf 7

TYPE: N-way by Change = **Çoklu seçim.**

Her bir seçimin % lik değerleri belirlenir.

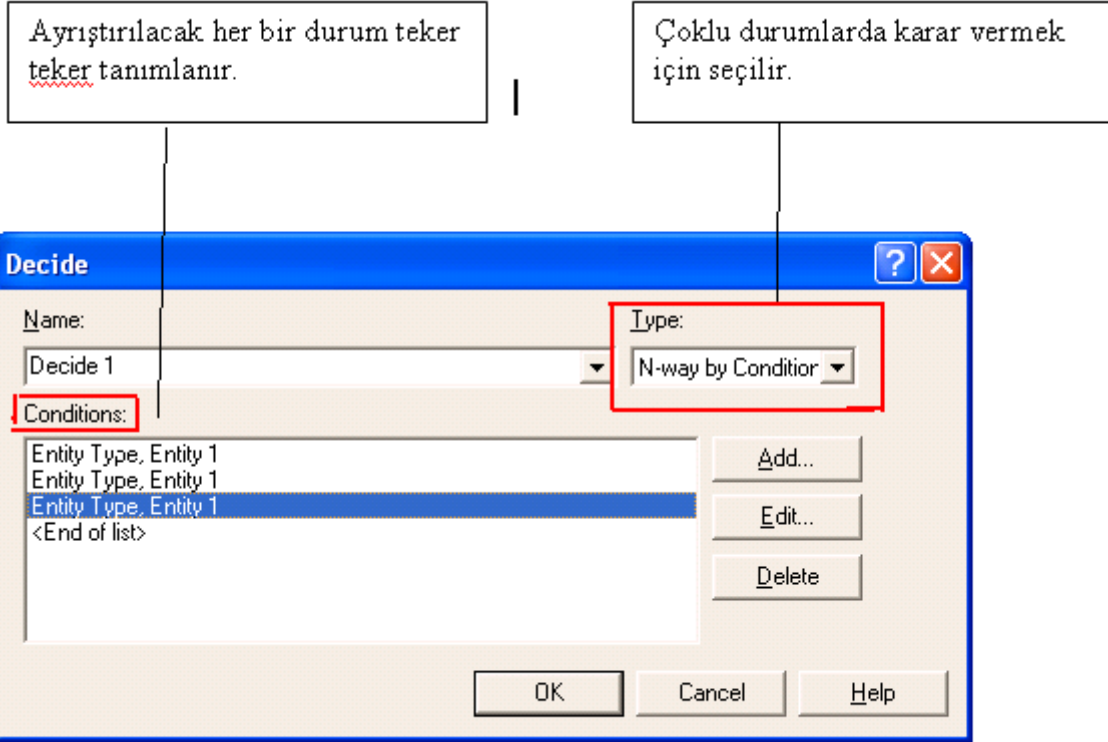
Çoklu seçimlerde karar vermek için seçilir.

The screenshot shows the 'Decide' dialog box with the following fields:

- Name: Decide 1
- Type: N-way by Chance
- Percentages: 50, 25, 25, <End of list>

Buttons: Add..., Edit..., Delete, OK, Cancel, Help

TYPE: N-way by Condition = **Çoklu Seçim**.



## 1.5. BATCH

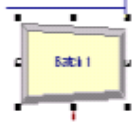
Syf 8

Bu modül, simülasyon modeli içinde gruplama mekanizmasını tasarlar. Batch'lar, sürekli yada geçici olarak gruplanabilir. Geçici Batch, kullanılan Separate (aşağıda açıklanmıştır) modülüyle daha sonra bölünmelidir.

Batch'lar, giren varlıkları belirtilmiş sayıyla yapabilir yada bir özelliğe dayanarak diğerleriyle eşleştirilebilir. Gerekli varlık sayısı birikene kadar, Batch modülünde varlık gelişleri bir kuyruğa yerleştirilir. İlk kümelenir, yeni bir temsilci varlık oluşturulur. Simgesi ve açıklamalar aşağıdadır.

Gruptaki parça miktarları  
(örneğin 20 şerli gruplar)

Batch türü:  
Temporary: Geçici  
Permanent: Sürekli



Grupların serbest bırakılma şeklini belirler.

Any Entity: Bütün parçalar gruplanıyorsa  
By Attribute: Aynı özellikte parçaların gruplanması isteniyorsa (Örneğin kırmızı parçalar)

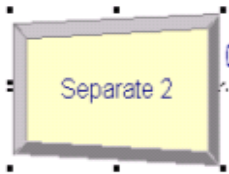
## 1.6. SEPARATE

Bu modül, çoklu varlıkların içine gelen bir varlığı kopyalamakta yada önceden oluşturulan bir varlık yığını bölme için kullanılabilir. Varlıklar çoğaltıldığında, modülden belirli sayıda kopya yapılır ve gönderilir. Gelen orijinal varlıklar aynı zamanda modülden ayrılır. Ayırma tipine bağlı olarak, belirlenmiş çıkış noktaları sayısı istenir. Varlıklar çoğaltıldığında, 2 çıkış noktası uygundur. Simgesi ve açıklamalar aşağıdadır.

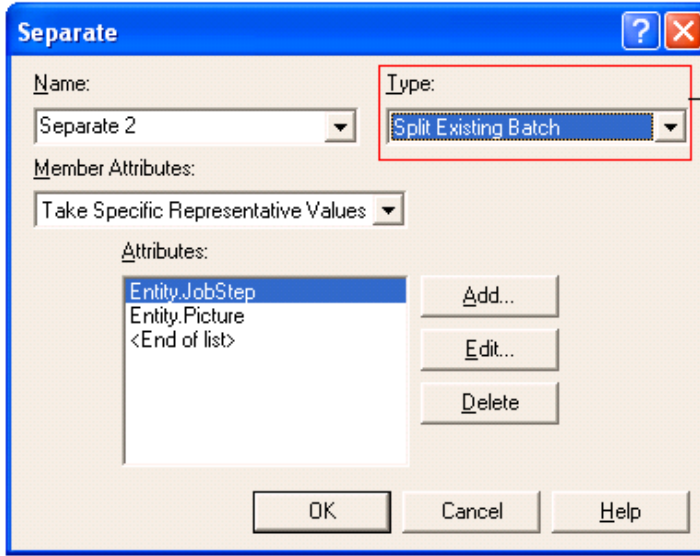
Syf 9

Ek çoğaltmaların (fatura) varlığın maliyetine/zaman yüzdesini belirler.

Duplicate Original: orijinal varlığı serbest bırakacak ve bu varlık için fatura teslim fişi gibi çoğaltmalar yapacak.



Kopya varlık sayısı



Split Existing Batch:  
Batch Modülünde  
birleştirilen grubu böler.

## 1.7. ASSIGN

Bu modül, değişkenlere varlık özelliklerine, varlık tiplerine, varlık resimlerine yada diğer sistem değişkenlerine yeni değer atanması için kullanılır. Tek bir Assign modülle çoklu atamalar yapılabilir. Simgesi ve açıklamalar aşağıdadır.

Varlığa istenen atamalar yapılır.

Type kısmına varlığa ne tür bir atama yapılacağı belirlenir.

Variable: Varlığa atanacak değişkenler.

Attribute: Varlığa atanacak özellikler.

Entity Type: Varlığa atanacak yeni varlık tipi

Entity Picture: Varlığa atanacak yeni varlık resmi.

Other: Yeni sistem değişkenleri tanımlanır.

Özelliğe, değişkenlere yada diğer sistem değişkenlerine **değer atar**.

## 1.8. RECORD

Bu modül, simülasyon modelinde istatistikleri biriktirmekte kullanılır. Çeşitli gözlemsel istatistikler kullanılır, modül içinde çıkışlar arası zaman, varlık istatistikleri (zaman, maliyet vb.), genel gözlemler ve ara (interval) istatistikleri içerir. Sayılabilir tip bir istatistik olduğu gibi kullanılır. Tally ve Counter setleri aynı zamanda belirlenebilir. Simgesi ve açıklamalar aşağıdadır.

**Count:** İstenilen değeri sayacak.

**Entity Statistics:** Varlıkların istatistiklerini zaman ve maliyet/süre oranı bilgisi şeklinde üretir.

**Time Interval:** Belirli bir özellik değeri ile geçerli simülasyon zamanı arasındaki farkı hesaplar ve kaydeder.

**Time Between:** Varlıkların modüle girişler arası zamanını hesaplar.

**Expression:** Belirli bir ifade değerini kaydeder.



The screenshot shows the 'Record' dialog box with the following fields and values:

- Name: Record 1
- Type: Count
- Value: 1
- Counter Set Name: Counter Set 1
- Set Index: Entity.Type
- Record into Set:

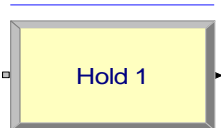
**Counter Name:** Bu alan, artma-azalma sayaçlarının sembol adını tanımlar.

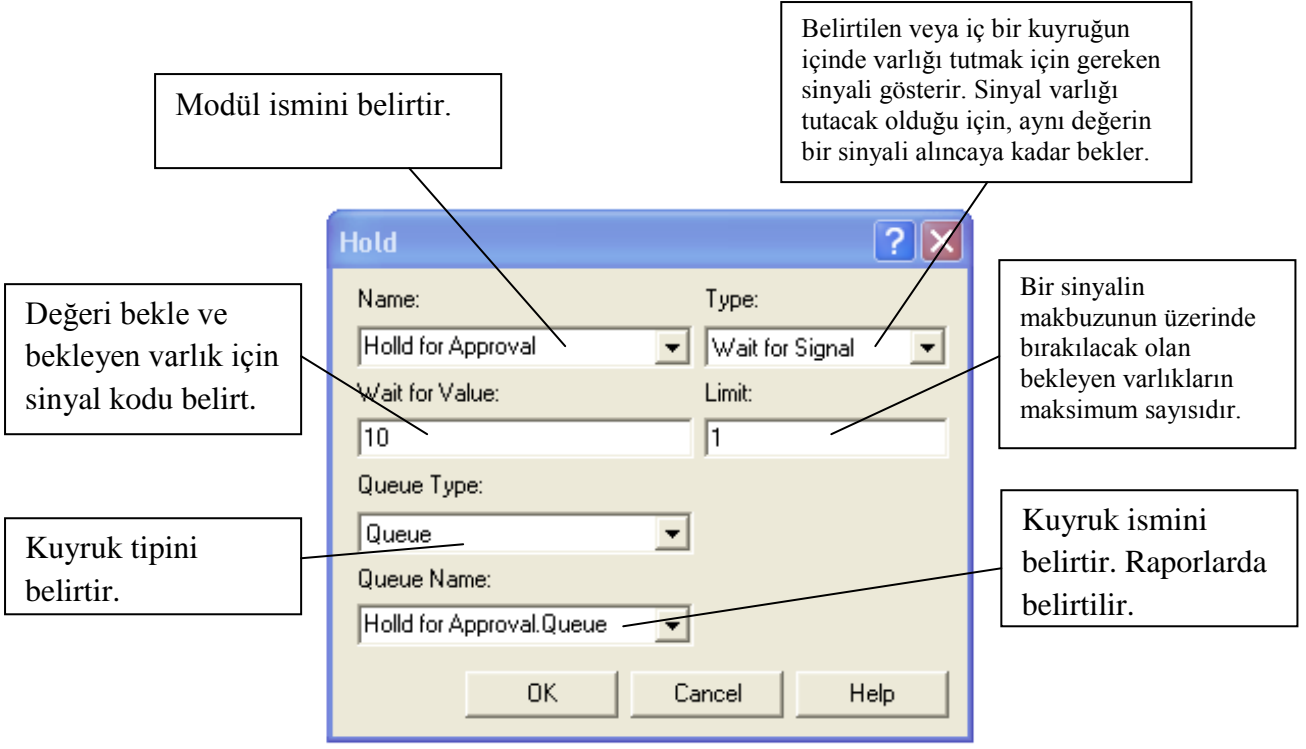
**Tally Name:** Bu alan, kaydedilecek gözlem içinde TALLY sembol adını tanımlar.

**Attribute Name:** Aralık istatistikleri için kullanılacak özellik değerinin adıdır.

## 1.9. HOLD

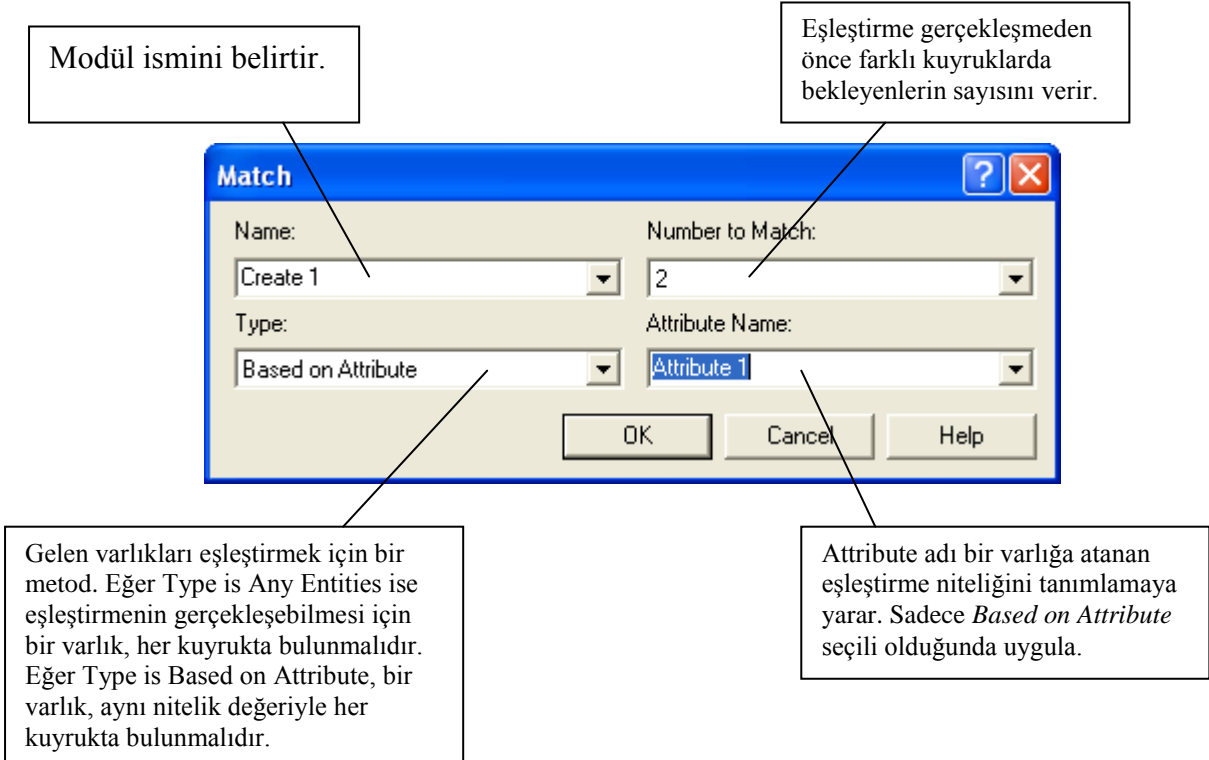
Bu modülde eğer varlık bir sinyal için tutuluyorsa, sinyal modülü varlığa sonraki modüle geçmek için izin vermede kullanılır. Eğer varlık, verilmiş bir halin doğru olması için tutuluyorsa, varlık (tanımlanan veya iç kuyruktan her biri) hal doğru oluncaya kadar modülde kalacak. Varlık sınırsız bir tutmada olduğu zaman taşıma modülü, işleme tabi tutmaya devam etmek için izin vermede kullanılır. Simgesi ve açıklamalar aşağıdadır.





## 1.10. MATCH

Match modülü farklı kuyruklarda bekleyen varlıkları belli sayılarda gruplar, bir araya getirir. Match komutunun işlem görebilmesi için belirtilen(tanımlanan) kuyruklarda en azından bir varlık olması gerekir. Ayrıca match komutu kullanılmadan önce, kuyruklarda beklemekte olan varlıkların en az bir ortak Attribute(özellik)leri olmalıdır.



## 1.11. ROUTE

Route (Rota) modülü, belirtilen bir istasyona bir varlığı transfer eder veya istasyona ziyaret sırasında, sonraki istasyona gezen birimi tanımlamak için kullanılır. Bir gezen birim rota modülüne girdiği zaman, onun istasyon niteliği (varlık, istasyon), varış yeri istasyonuna girendir. Gezen birim varış yeri istasyonuna rota zamanı kullanılarak yollar. Eğer istasyon varış yeri, öyle sıra ile girilirse, sonraki istasyon varlığın sırası ile kararlaştırılır ve takımın içinde adım özel-maksat nitelikler varlığı tarafından tanımlanır. Simgesi ve açıklamalar aşağıdadır.

Route 1

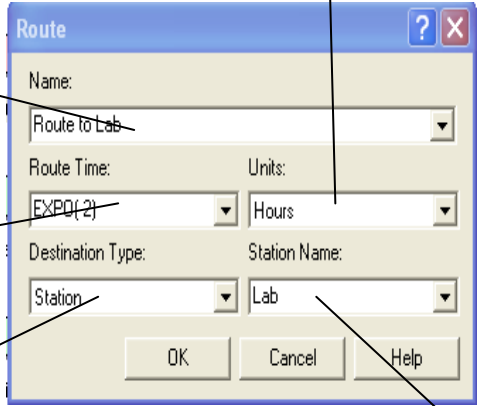
Gezen birimin rota ismi tanımlanır.

Laboratuar istasyonuna varması için varlık için gecikme zamanı, 2 zaman ünitesinin bir ortalamasıyla üstel bir dağıtmada temel alınır.

Gezen birimin varış yerinin istasyon olduğu belirtilmektedir.

Taşıma zamanı için zaman birimi.

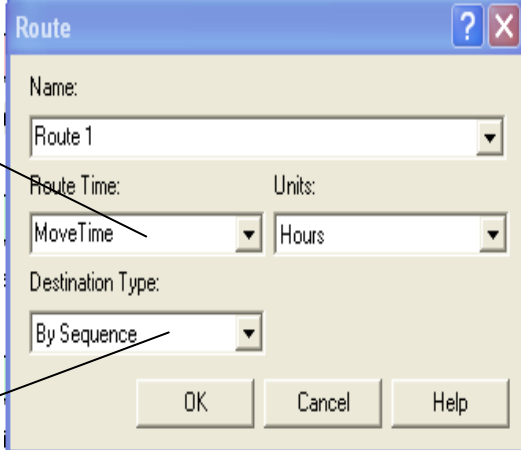
Gezen birimin varış yeri istasyon isminin laboratuar olduğu belirtilmektedir.



Syf 13

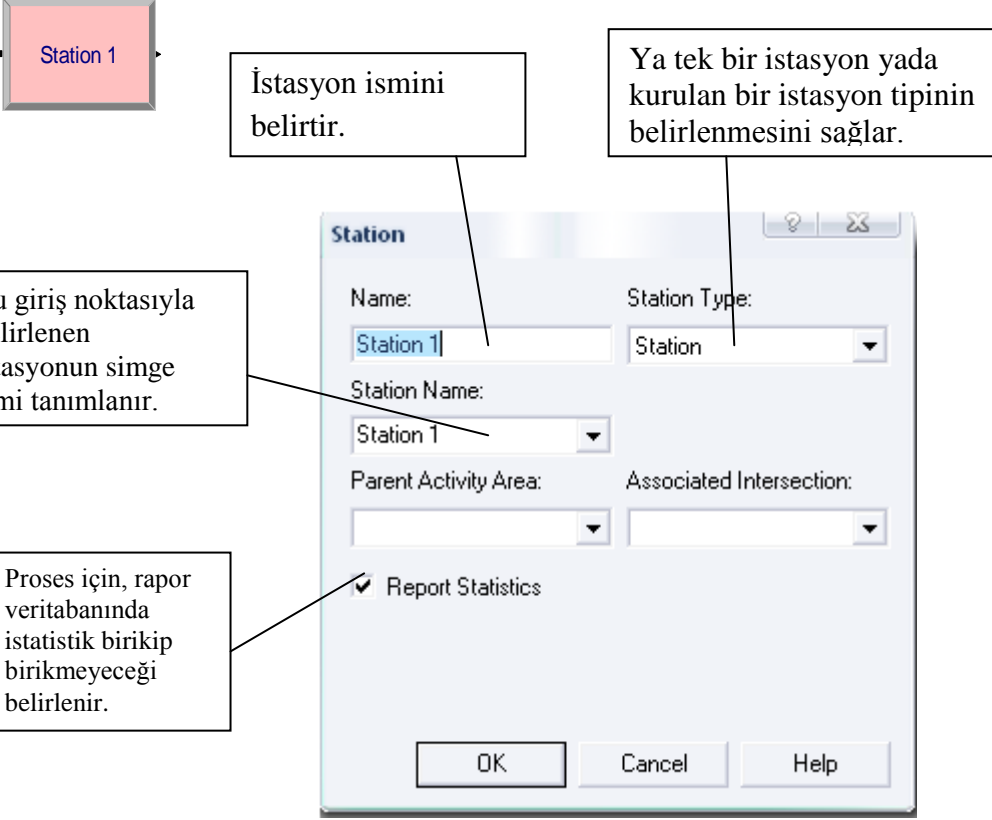
Transfer gecikme zamanı için bir nitelik olabilen hareket zamanı temel alınır ve bu zaman varlığın sıra tayinlerinde tanımlanır.

Gezen birim, varlığın sırası ile belirtilen sıra takımında sonraki istasyona gönderilir.



## 1.12. STATION

Route komutu kullanıldığında gezen birimin gideceği yerleri tanımlamak için kullanılır. Station modülü hareketli kaynakları veya durağan olmayan kaynakların olduğu ilgili bir park alanına sahip olabilir. Simgesi ve açıklamaları aşağıdadır.



The diagram illustrates the configuration of a station. On the left, a red square symbol labeled "Station 1" is shown. To its right, a dialog box titled "Station" is displayed. The dialog box contains the following fields and options:

- Name:** A text input field containing "Station 1".
- Station Type:** A dropdown menu set to "Station".
- Station Name:** A dropdown menu set to "Station 1".
- Parent Activity Area:** A dropdown menu.
- Associated Intersection:** A dropdown menu.
- Report Statistics:** A checked checkbox.

Four callout boxes provide additional information:

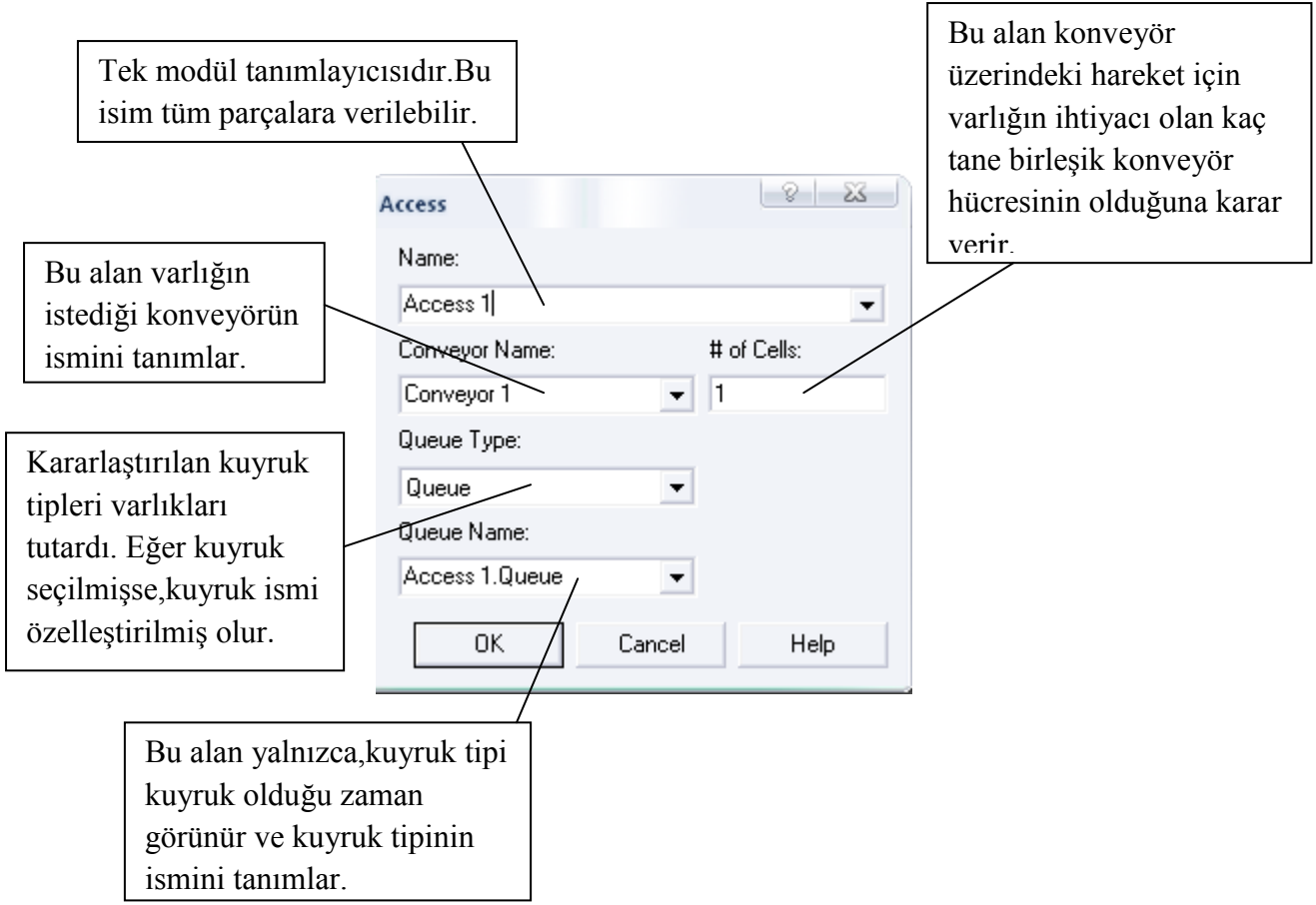
- Top left: "İstasyon ismini belirtir." (Indicates the station name.)
- Top right: "Ya tek bir istasyon yada kurulan bir istasyon tipinin belirlenmesini sağlar." (Allows for either a single station or the selection of a station type.)
- Middle left: "Bu giriş noktasıyla belirlenen istasyonun simge ismi tanımlanır." (The symbol name of the station is defined by this input point.)
- Bottom left: "Proses için, rapor veritabanında istatistik birikip birikmeyeceği belirlenir." (Determines whether statistics will be accumulated in the report database for the process.)

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

## 1.13. ACCESS

Access modülü, varlığın bir istasyondan diğerine hareketi için konveyörün bir yada daha fazla hücrelerine yer tahsis eder. Varlık bir kez konveyör üzerindeki hücrelerin kontrolüne sahiptir. Belki varlık gelecek istasyona taşınacak olabilir. Varlık Access modülüne geldiği zaman konveyör üzerindeki bitişik hücrelerin uygun numaralarına kadar bekleyecektir ve bu numaralar boş ve diğer varlık istasyon yeriyile sıraya dizilmiştir. Simgesi ve açıklamalar aşağıdadır.





#### 1.14. CONVEY

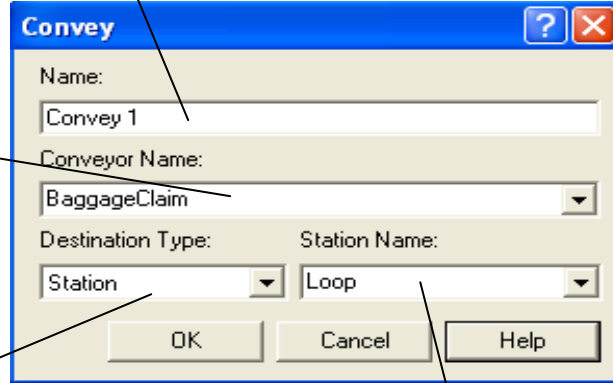
Convey modülü aracılığı ile bulunduğu istasyondan belirtilen varış istasyonuna taşır. Bir sonraki istasyona varlığın taşınması esnasında oluşan zaman gecikmesi taşıyıcı hızından ve istasyonlar arası mesafeden kaynaklanır. Bir varlık convey modülüne girdiği zaman, onun istasyon niteliği de varış istasyonuna taşınır. Varlık varış istasyonuna özellikten sonra taşınır. Eğer varış yeri tipi By Sequence olarak belirtilirse, sonraki istasyon varlığın Sequence'si ve bunun içinde yer alan adımlar ile belirlenir. Simgesi ve açıklamalar aşağıdadır.



Benzersiz modül tanıyıcısı. Bu isim, modül şeklinde gösterilir.

Bu alan, kendisinde varlığın, taşıyacak olduğu taşımacının ismini tanımlar.

Bu, varlık varış yerini belirtmek için metodu kararlaştırır.



Bu alan, varış yeri tipinin, istasyon olduğu zaman görünür tektir, ve o, kendisine varlığın, taşıyacak olduğu istasyonun ismini tanımlar.

### 1.15. EXIT

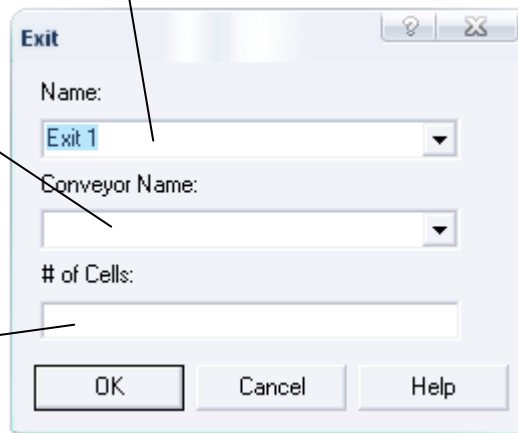
Bu modül Access modülü ile Conveyöre alınan gezen birimi herhangi bir işlem için konveyörden almaya yarar. Convey komutu ile taşınan bir gezen birim mutlaka ilgili istasyona geldiğinde prosese girmeden önce conveyörden alınmalıdır. Aksi halde taşıyıcı sürekli dolu görünecek bu da yanıltıcı sonuçlar doğuracaktır. Modülün simgesi ve açıklamaları aşağıda verilmiştir.



Modül ismini belirtir.

Gezen birimin alınacağı conveyor ismini belirtir.

Bu alan konveyör üzerindeki hareket için varlığın ihtiyacı olan kaç tane birleşik konveyör hücresinin olduğuna karar verir.



## 1.16. REQUEST

İstek modülü, bir varlığa bir taşıyıcı ünitesini tayin eder ve varlığın yerine üniteye hareket eder. Özel bir taşıyıcı ünitesi belirtilebilir veya seçim bir kural temel alınarak meydana gelebilir. Varlık, istek modülüne vardığı zaman, biri müsait olduğu zaman bir taşıyıcı ayrılır. Taşıyıcı ünitesi, varlık yerine uzanincaya kadar istek modülünde kalır. Varlık sonra, istek modülünden dışarı hareket eder.

İlgili açıklamalar aşağıdadır.



Benzersiz modül tanıyıcısı. Bu isim, modül şeklinde gösterilir.

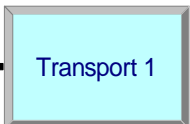
Rica etmesi için taşıyıcının ismini tanımlar.

Varlıkların ünite zamanı başına uzunluk ünitelerinde yerine hareket ettikleri zamandaki hızıdır.

Bu modül hakkındaki diğer bilgiler pek kullanılmayan özellikler olduğu için üzerinde durulmamıştır.

## 1.17. TRANSPORT

Bu modül yine gezen birimin taşınmasında kullanılır. Bu modülde taşıyıcı sınırlaması vardır. İstedığımız kadar taşıyıcıyı biz tanımlarız. Request komutu ile çağırılan taşıyıcı Transport modülü ile ilgili istasyona gittikten sonra Free modülü ile mutlaka boşaltılmalıdır. Modül ile ilgili açıklamalar aşağıdadır.



Benzersiz modül tanıyıcısı. Bu isim, modül şeklinde gösterilir.

Elimizde olan taşıyıcı miktarını buraya yazarak belirleyebiliriz.

Taşıyıcının sistem içerisindeki ismi.

Gezen birimin varacağı yeri belirlemek için kullanılır.

Varlıkların ünite zamanı başına uzunluk ünitelerinde yerine hareket ettikleri zamandaki hızıdır.

Modül ile ilgili diğer açıklamalar detay oluşturacağından verilmemiştir. Fakat Destination Type bölümüne attribute veya sequence tanımlanarak da taşıma sağlanabilir. Bunlar yukarıda anlatılan benzer modüller ile aynıdır.

### 1.18. FREE

Bu modül varlığın en son pay edilmiş taşıyıcısını salıvermek için kullanılır. Eğer sırada taşıyıcı istemek veya pay etmek için bekleyen bir varlık varsa, taşıyıcı o varlığa verilir. Taşıyıcı modülünde başka türlü ayarlanmadığı sürece; taşıyıcının boşaldığı anda bekleyen başka varlıklar yoksa, taşıyıcı ünite serbest bırakan varlığın istasyonunda boşta bekletilir. İlgili açıklamalar aşağıdadır.

Free 1

Benzersiz modül tanıyıcısı. Bu isim, modül şeklinde gösterilir.

Boşaltılması gereken taşıyıcının ismini tanımlar.

## 1.19. SEQUENCE

Bir modül olmasa da özellikle taşıyıcı görevini yapan modüllerde kullanılan bu komut, taşıma faaliyetlerini sistem tasarımı sırasında epeyce kolaylaştırmaktadır. Advanced Transfer başlığı altındaki modüllerden Sequence seçildiğinde açılan kısma sırası ile ilgili gezen birimin işlem sırası girilirse sistem tasarımı içerisinde tekrar tekrar işlem öncelikleri ile ilgilenmeye gerek kalmayacaktır. Bu da bize büyük kolaylıklar sağlayacaktır.

## 2. ÖRNEKLER

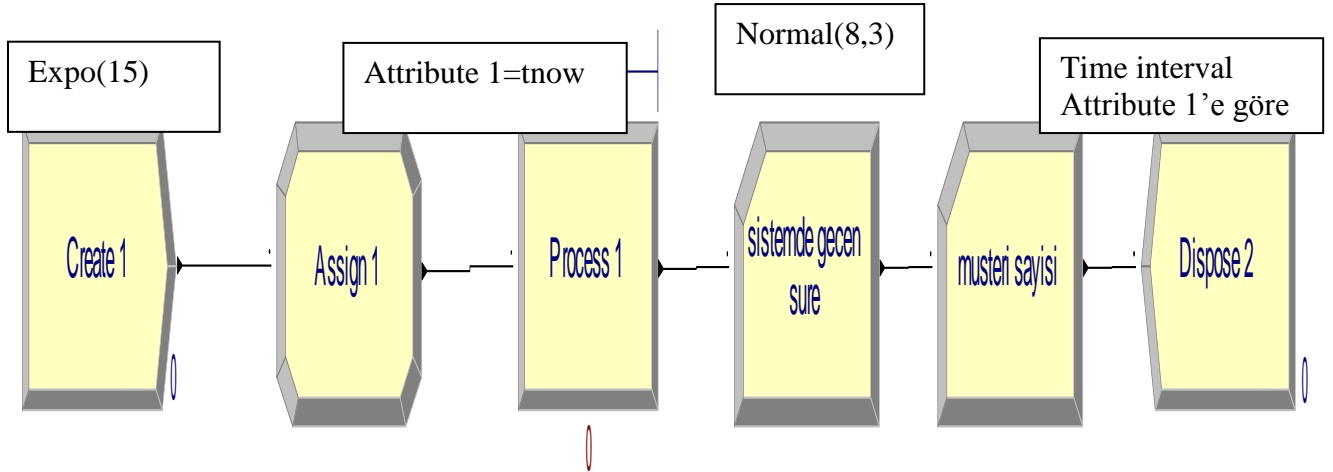
Örnekler geçmiş yıllarda çözülen örnekler, ödevler veya sınav soruları incelenerek ve sorular üzerinde çeşitli deęiřtirmeler yapılarak hazırlanmıştır.

### 2.1. Berber Örneęi

Geliřler arası süre  $Ex(15)$  olan ve tırař süreleri  $normal(8,3)$  daęılımına uyan bir berberde 200 dakika çalıřılması durumunda oluřacak hizmet gören müşteri sayısı, hizmet görenlerin ortalama sistemde geçirdięi süre ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

**Çözüm:** Sorunun arena çıktıları ařaęıda verilmiştir.

Syf 19

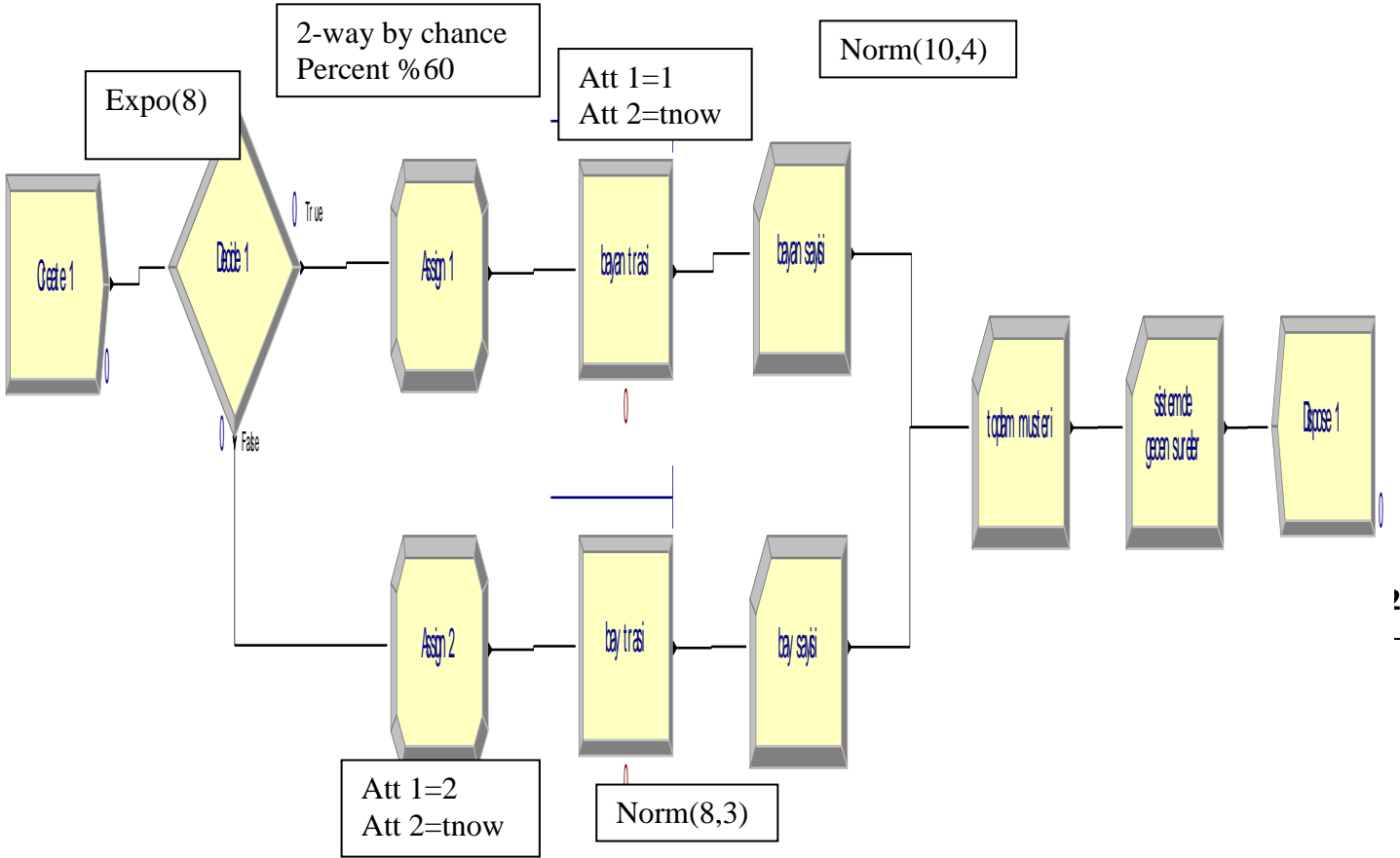


### 2.2. Geliřtirilmiř Berber Örneęi

Geliřler arası süre  $Ex(8)$  olan, gelen müşterilerin %60'nın bayan %40'nın bay olduęu, bayan ve bay müşteriler için ayrı ayrı tırař koltuklarının bulunduęu, ve tırař sürelerinin bayanlar için  $normal(10,4)$ , baylar için  $normal(8,3)$  daęılımına uyan bir berberde 200 dakika çalıřılması durumunda oluřacak hizmet gören müşteri sayısı, hizmet gören bay veya bayan sayılarını,

hizmet görenlerin ortalama sistemde geçirdiği süreleri (bay veya bayan için ayrı ayrı), kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

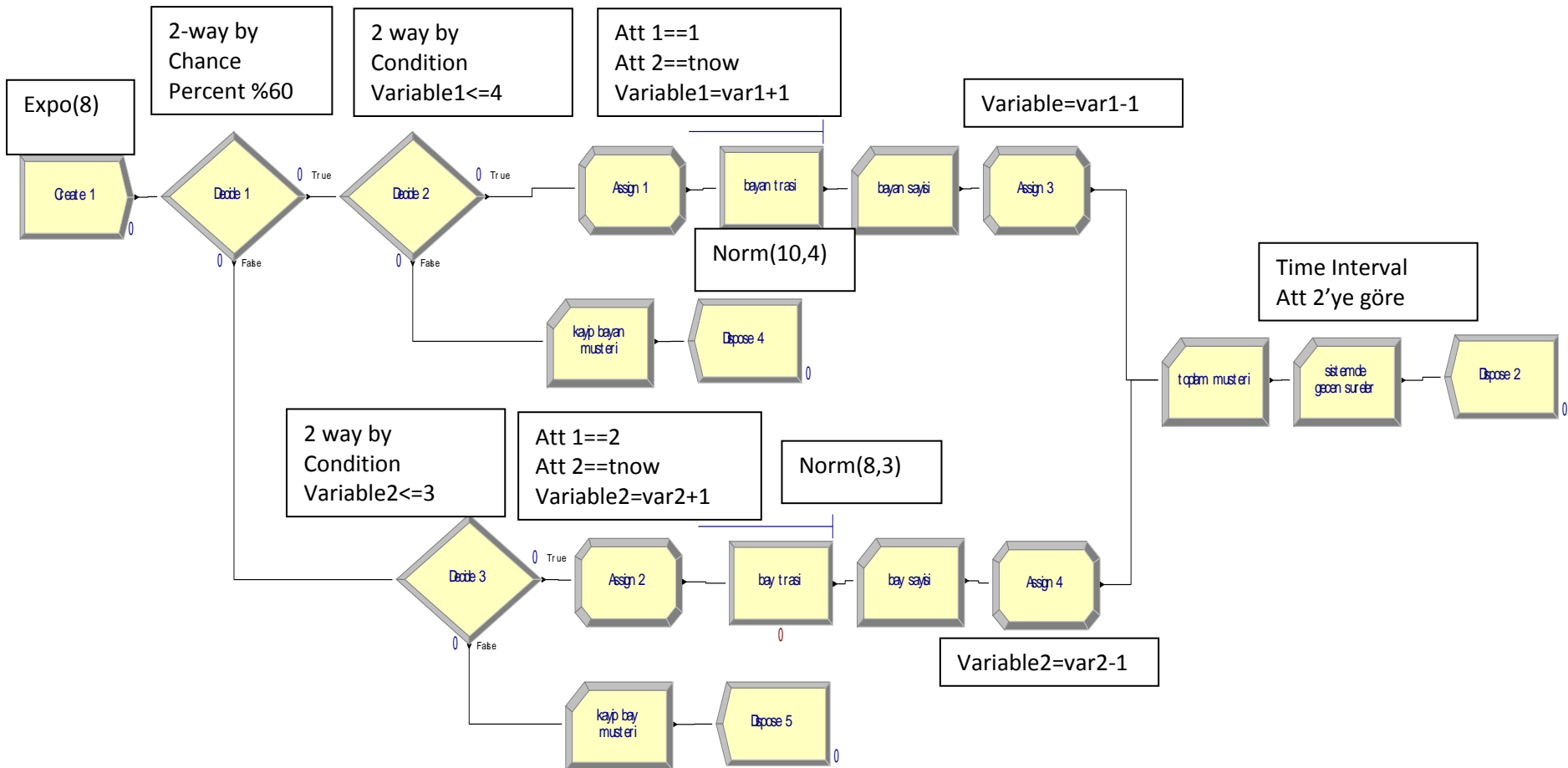
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



### 2.3. Bekleme Yeri Sınırlı Berber Örneği

Gelişler arası süre  $Ex(8)$  olan, gelen müşterilerin %60'nın bayan %40'nın bay olduğu, bayan ve bay müşteriler için ayrı ayrı tıraş koltuklarının bulunduğu, bayanlar için bekleme salonu kapasitesinin 5 kişi, baylar için 4 kişi olduğu ve tıraş sürelerinin bayanlar için normal(10,4), baylar için normal(8,3) dağılımına uyan bir berberde 200 dakika çalışılması durumunda oluşacak hizmet gören müşteri sayısı, hizmet gören bay veya bayan sayılarını, hizmet görenlerin ortalama sistemde geçirdiği süreleri (bay veya bayan için ayrı ayrı), kayıp müşteri sayısını ve kuyruk durumlarını gösteren Arena simülasyonunu yapınız.

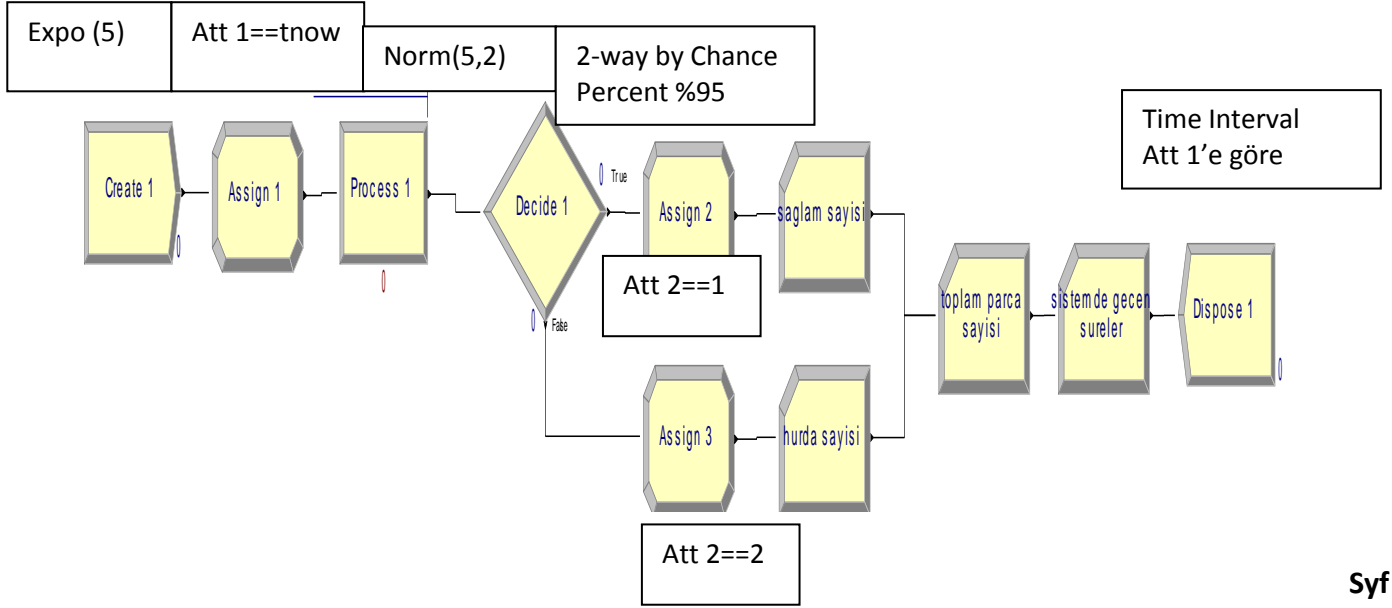
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.4. Atölye Örneği

Gelişler arası süre  $Ex(5)$  olan, gelen parçaların %5'inin hurda %95'inin sağlam olduğu, işlem süresinin  $normal(5,2)$  dağılımına uyan bir atölyede işlem gören parça sayıları, sistemde geçen süreler ve kuyruk durumunu veren Arena simülasyonunu yapınız.

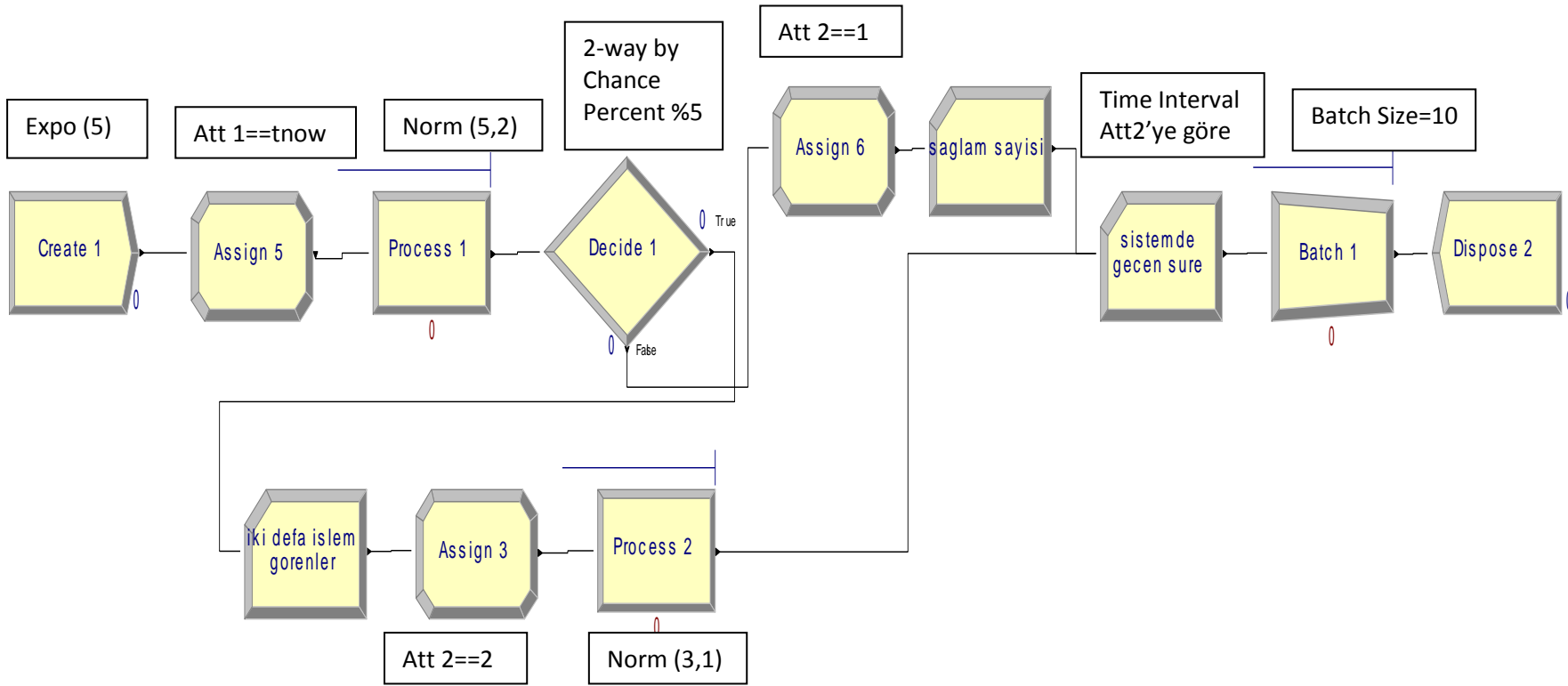
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.5. Tekrar İşlemeli Atölye Örneği

Gelişler arası süre  $Ex(5)$  olan, gelen parçaların %5'inin hurda %95'inin sağlam olduğu, işlem süresinin  $normal(5,2)$  dağılımına uyan, hurda parçaların  $normal(3,1)$  işlem süresi ile tekrar işlemeye alındığı ve atölyeden çıkışların 10'arlı partiler halinde olduğu bir atölyede işlem gören parça sayıları, iki defa işlem gören parça sayıları, sistemde geçen süreler ve kuyruk durumunu veren Arena simülasyonunu yapınız.

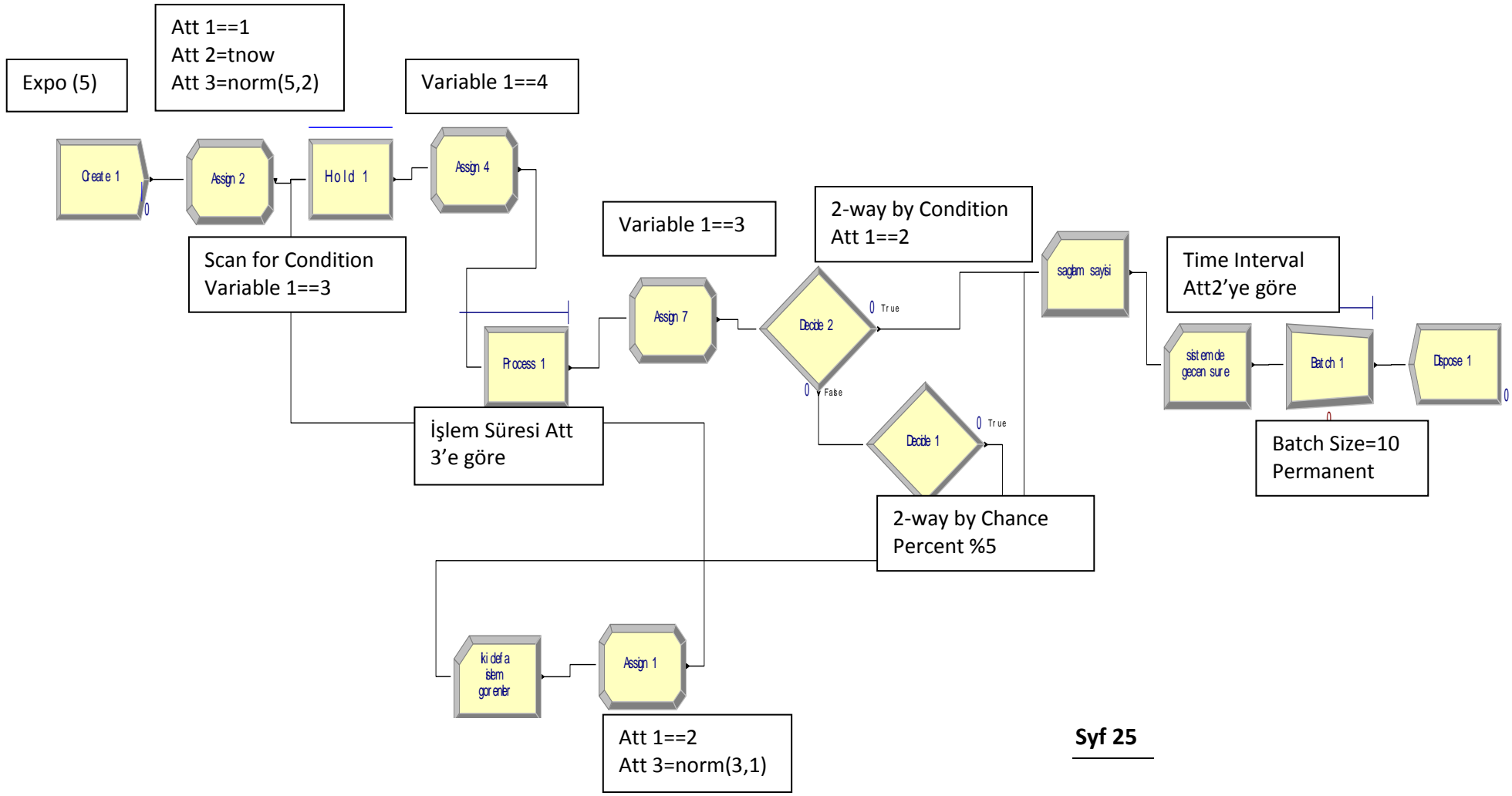
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.6. Bekleme Yeri Sınırlı Atölye Örneđi

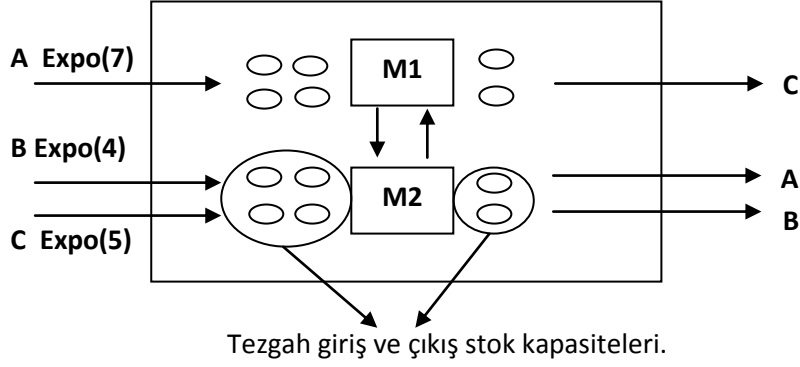
Gelişler arası süre  $Ex(5)$  olan, gelen parçaların %5'inin hurda %95'inin sağlam olduđu, işlem süresinin  $normal(5,2)$  dağılımına uyan, hurda parçaların  $normal(3,1)$  işlem süresi ile tekrar işlemeye alındığı, tezgah önünde bekleyecek parça sayının 4 ile sınırlandırıldığı ve atölyeden çıkışların 10'arlı partiler halinde olduđu bir atölyede işlem gören parça sayıları, iki defa işlem gören parça sayıları, sistemde geçen süreler ve kuyruk durumunu veren Arena simülasyonunu yapınız.

**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.7. Decide Örneği

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

A — M1[norm(4,2)]→M2[norm(5,2)]→çıkış

B — M2[norm(6,1)]→çıkış

C — M2[norm(7,3)]→M1[norm(7,3)]→çıkış

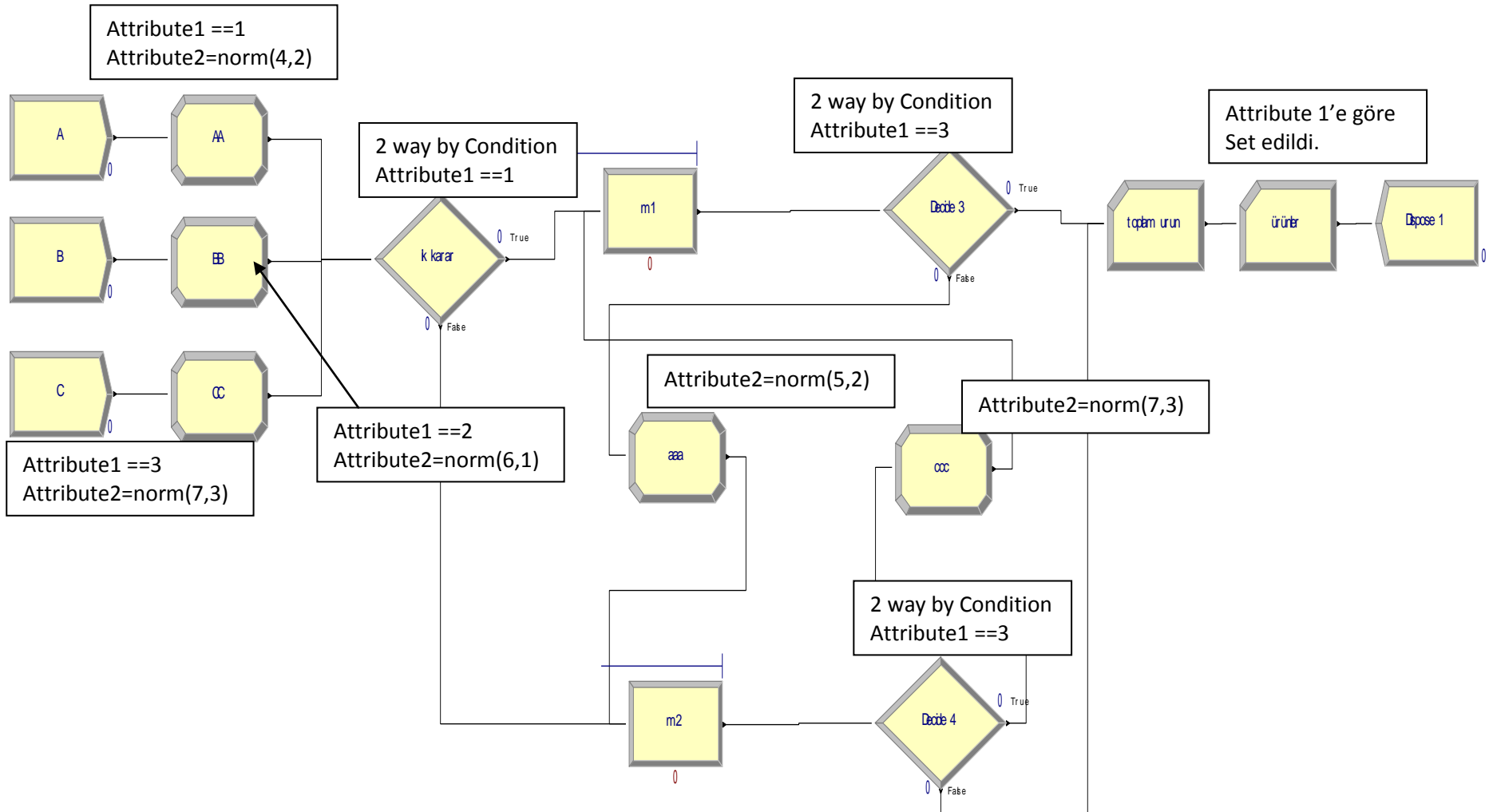
Soru Decide komutu kullanılarak çözülmüştür.

NOT: Attribute 1= Parça Tipi

Attribute 2= İşlem Süresi

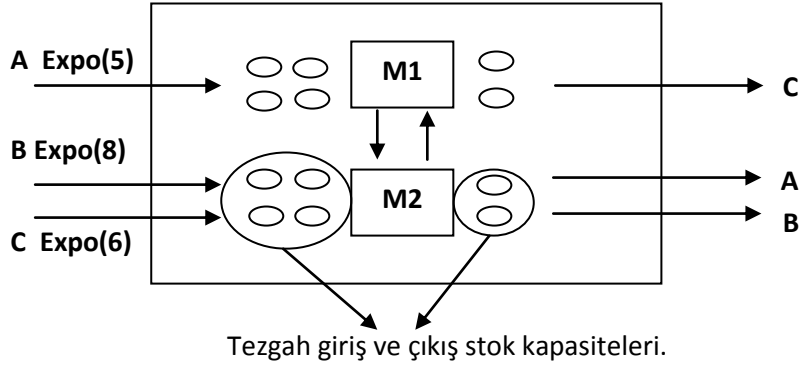
NOT: Soruda verilen süreler tesadüfi olarak seçilmiştir.

**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.8. Route Örneği

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

A — M1[norm(5,2)]→M2[norm(3,1)]→çıkış

B — M2[norm(4,2)]→çıkış

C — M2[norm(5,1)]→M1[norm(6,1)]→çıkış

Soru Route komutu kullanılarak çözülmüştür.

NOT: Route için Velocity=10 ve Taşıma Süresi olarak da=2 alınmıştır.

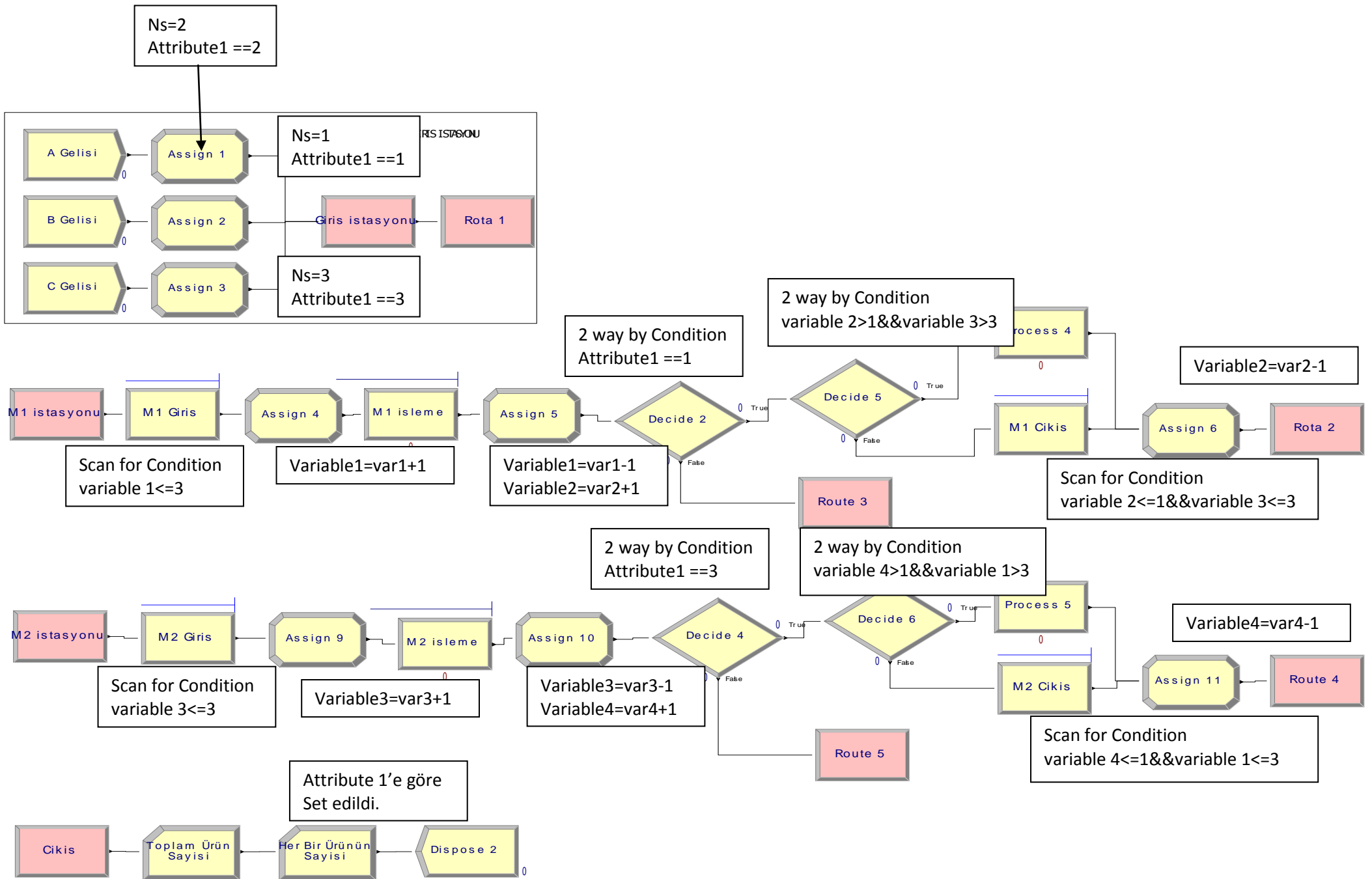
Attribute 1= Parça Tipi

Attribute 2= İşlem Süresi

Attribute 3=Taşıma Süresini göstermektedir.

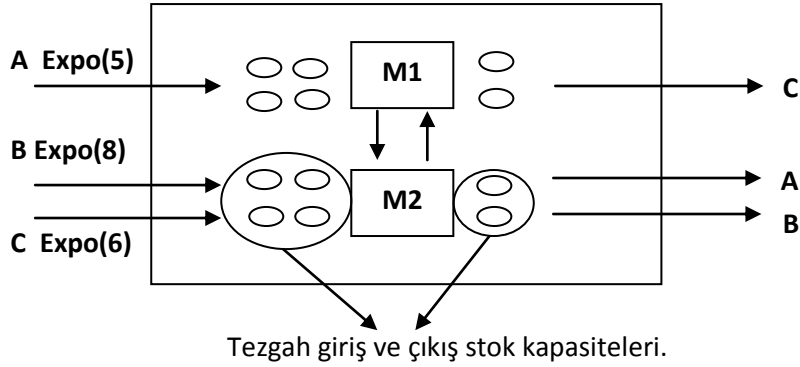
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.

NOT: Soruda verilen süreler tesadüfi olarak seçilmiştir.



## 2.9. Transport Örneği

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

A — M1[norm(5,2)]→M2[norm(3,1)]→çıkış

B — M2[norm(4,2)]→çıkış

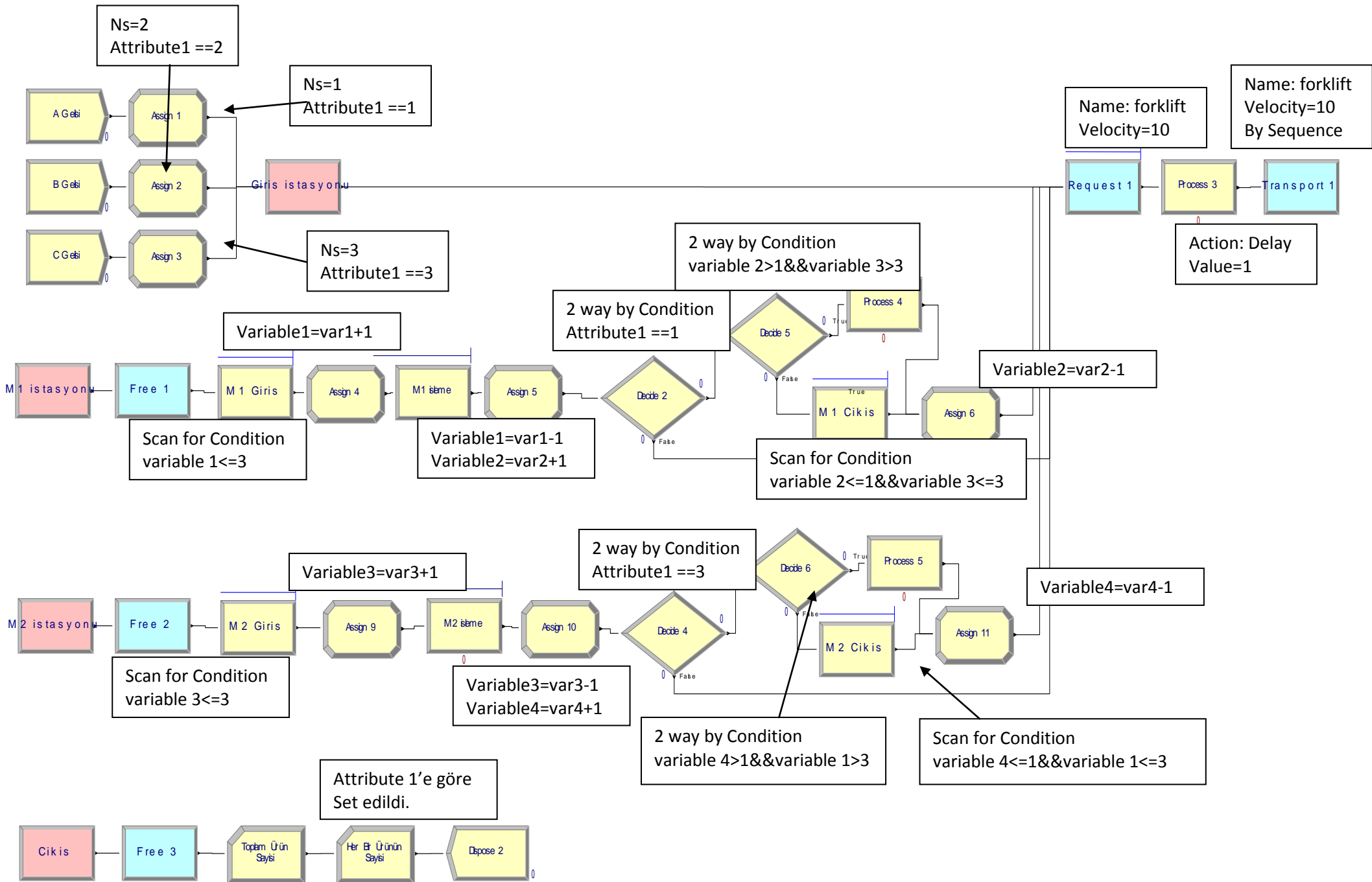
C — M2[norm(5,1)]→M1[norm(6,1)]→çıkış

Soru Transport komutu kullanılarak çözülmüştür.

NOT: Transport için Velocity=10 ve Taşıma Süresi olarak da=2 alınmıştır. Taşıyıcının gelme süresi ilgili proste sabit 1 br alınmıştır. Transport için Distance(gidilecek yerler arası uzaklıklar) hepsi 10 alınmıştır. Attribute değerleri bir önceki soru ile aynıdır.

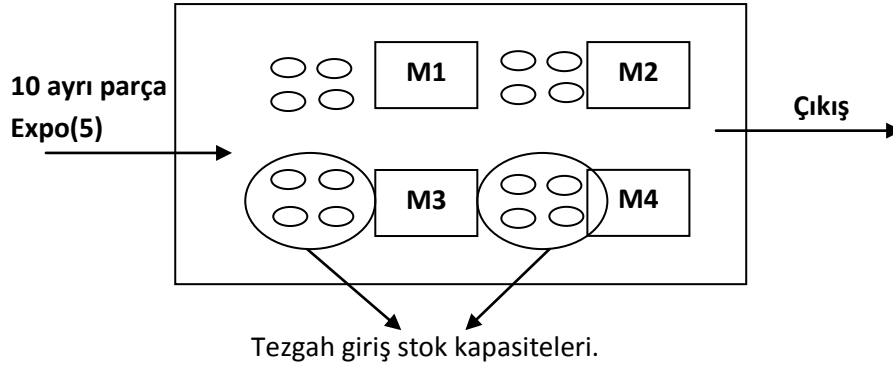
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.

NOT :Soruda verilen süreler tesadüfî olarak seçilmiştir.



## 2.10. Akışları Farklı Ürünler İçin Örnek

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

1	=	M1[norm(2,1)]→M3[norm(4,8)]→çıkış	<u>NOT:</u> Soruda verilen süreler tesadüfi olarak seçilmiştir.
2	=	M4[norm(4,4)]→çıkış	
3	=	M4[norm(3,0)]→M3[norm(5,8)]→M2[norm(6,1)]→çıkış	
4	=	M3[norm(9,0)]→M1[norm(8,9)]→çıkış	
5	=	M3[norm(1,7)]→M2[norm(2,3)]→M1[norm(4,4)]→çıkış	
6	=	M1[norm(7,3)]→çıkış	
7	=	M4[norm(9,2)]→M3[norm(4,6)]→çıkış	
8	=	M2[norm(3,9)]→M4[norm(1,7)]→çıkış	
9	=	M3[10]→çıkış	
10	=	M4[norm(8,0)]→M1[norm(7,2)]→M2[norm(4,1)]→M3[norm(9,9)]→çıkış	

Syf 32

Soru Transport komutu kullanılarak çözülmüştür.

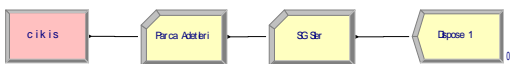
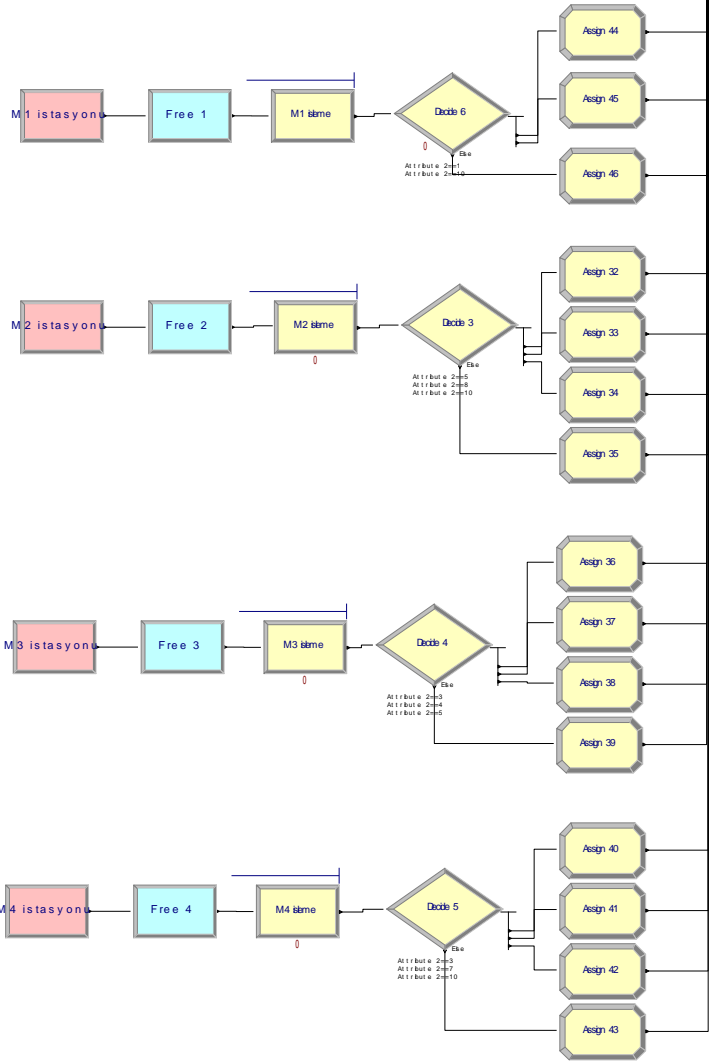
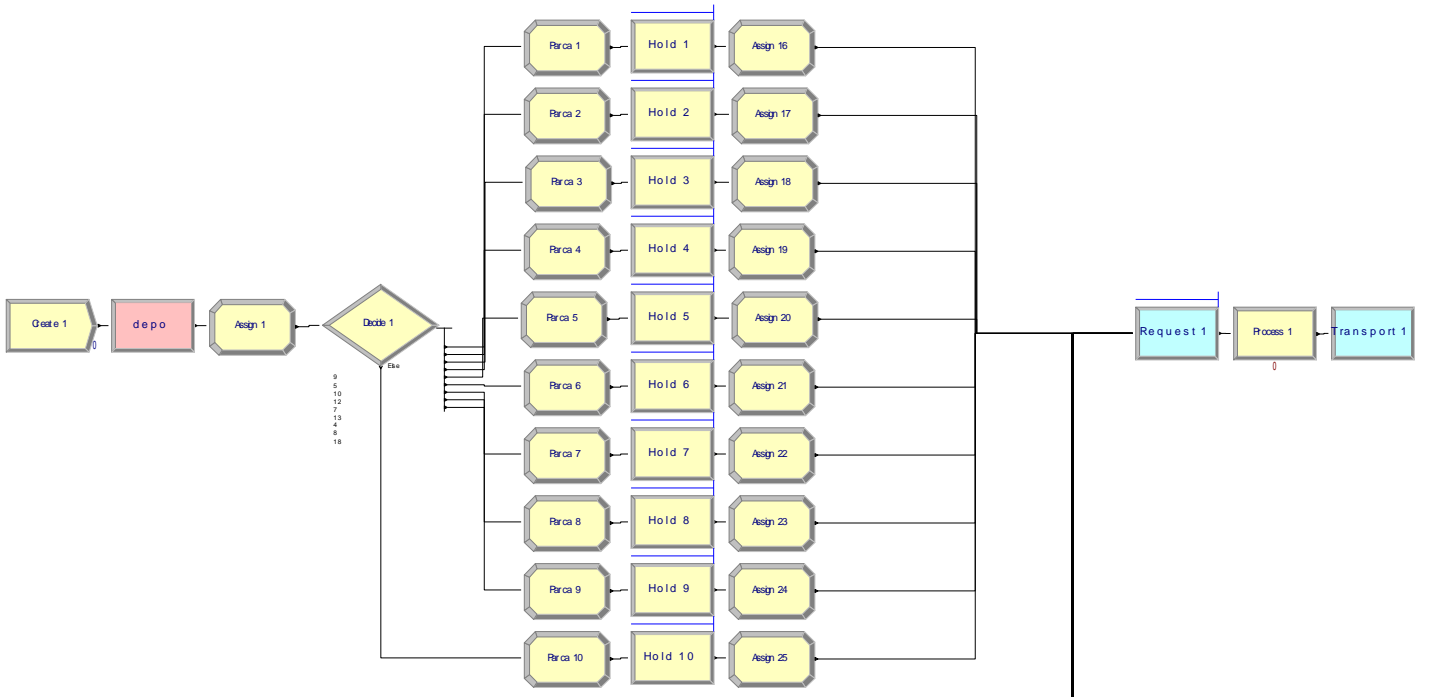
Attribute 1=tnow

Attribute 2=Parça Tipi

Attribute 3=İşlem Süresi

NOT: Transport için Velocity değeri 10 br alınmıştır. Taşıyıcının gelme süresi ilgili proseste sabit 1 br alınmıştır. Transport için Distance(gidilecek yerler arası uzaklıklar) hepsi 10 br alınmıştır.

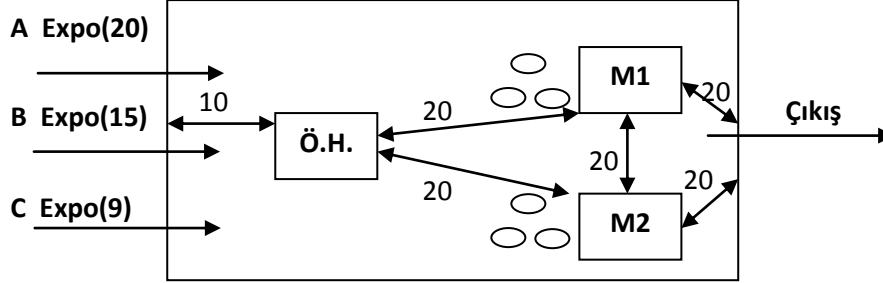
**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



NOT: Çıktı üzerine önceki sorularda olduğu gibi açıklamalar karmaşıklık yaratacağı gerekçesiyle eklenmemiştir.

## 2.11. Örnek

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

- A = Ö.H.[norm(2,1)]→M1[norm(4,8)]→M2[norm(5,4)]→çıkış  
B = Ö.H.[norm(4,4)]→M2[norm(3,5)]→M1[norm(3,2)]→çıkış  
C = Ö.H.[norm(3,0)]→M1[norm(5,8)]→çıkış

NOT: Soruda verilen süreler tesadüfi olarak seçilmiştir.

Soru Transport komutu kullanılarak çözülmüştür.

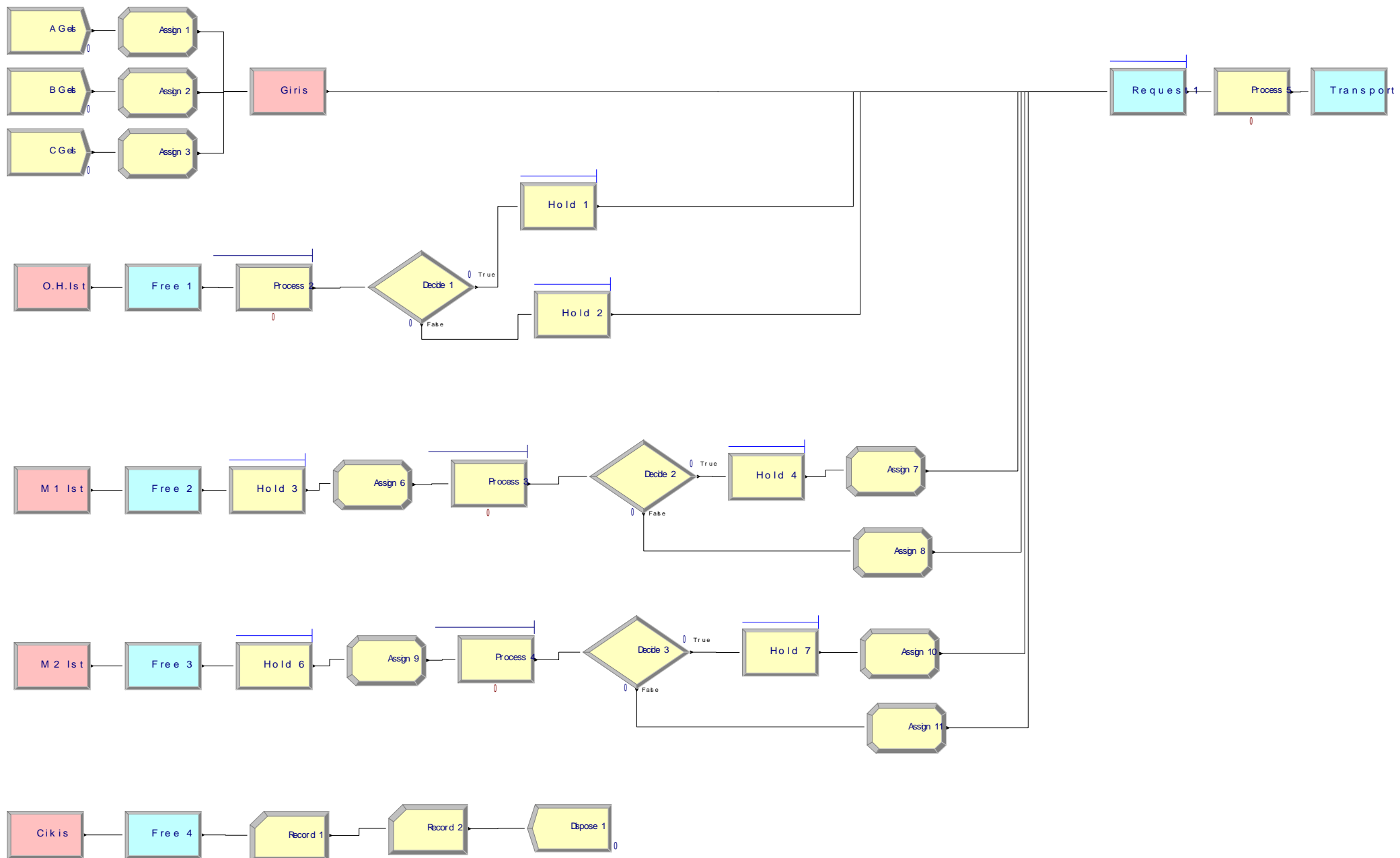
Attribute 1=tnow

Attribute 2=Parça Tipi

Attribute 3=İşlem Süresi

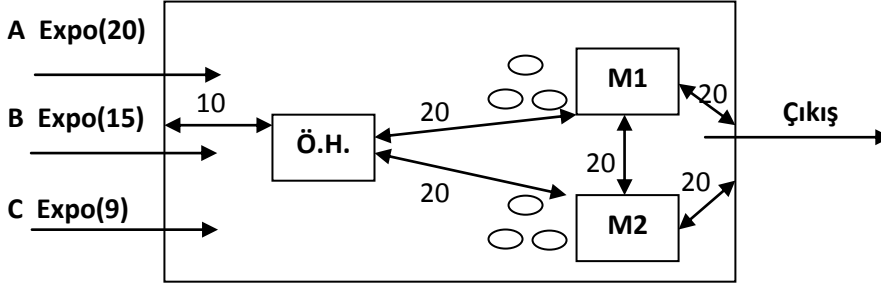
NOT: Transport için Velocity değeri 10 br alınmıştır. Taşıyıcının gelme süresi ilgili proseste sabit 1 br alınmıştır. Transport için Distance(gidilecek yerler arası uzaklıklar) şekilde  $\longleftrightarrow$  ile gösterilen mesafeler olarak alınmıştır.

**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



## 2.12. Animasyon Örneği

Aşağıda şekli ve ilgili bilgilerin verildiği atölyede parçalara ait Arena simülasyonunu ve animasyonunu kuyruk durumları ve işlem gören parça adetlerini verecek şekilde hazırlayınız.



### Rotalar ve İşlem Süreleri

- A = Ö.H.[norm(2,1)]→M1[norm(4,8)]→M2[norm(5,4)]→çıkış  
B = Ö.H.[norm(4,4)]→M2[norm(3,5)]→M1[norm(3,2)]→çıkış  
C = Ö.H.[norm(3,0)]→M1[norm(5,8)]→çıkış

NOT: Soruda verilen süreler tesadüfi olarak seçilmiştir.

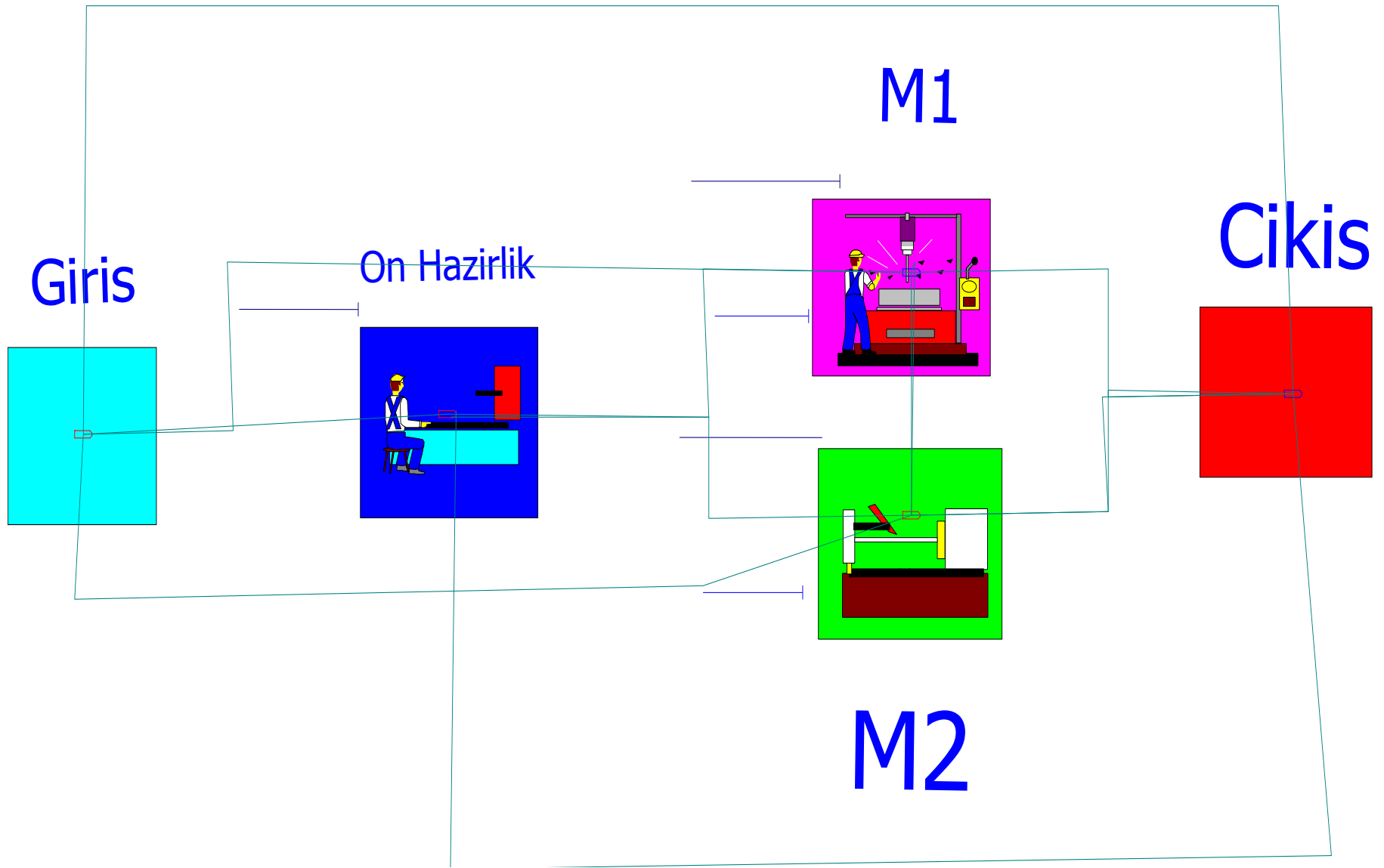
Soru Transport komutu kullanılarak çözülmüştür.

- Attribute 1=tnow  
Attribute 2=Parça Tipi  
Attribute 3=İşlem Süresi

**Syf 36**

NOT: Transport için Velocity değeri 10 br alınmıştır. Taşıyıcının gelme süresi ilgili proseste sabit 1 br alınmıştır. Transport için Distance(gidilecek yerler arası uzaklıklar) şekilde  $\longleftrightarrow$  ile gösterilen mesafeler olarak alınmıştır.

**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.



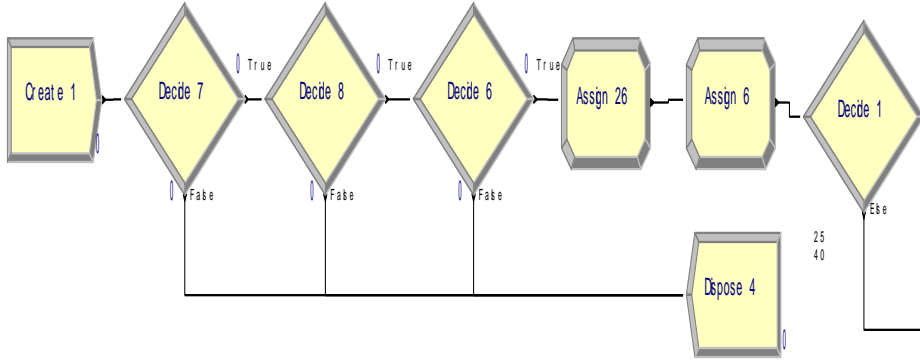
### 2.13. Banka Örneđi

Bir bankaya müşteriler random(exp) şeklinde gelişler arası süre 1 dk olarak gelmektedirler. Gelen Müşteriler önce numaratóre yönelmekte ve orada ortalama (3:1) saniye kadar zaman kaybettikten sonra gişelere yönelmektedir. Yapacakları işleme göre numaratórden numara alan müşteriler havale için 1ve 2 nolu gişeye hesap işlemleri için 3 ve 4 nolu gişeye fatura ödemeleri için 5 ve 6 nolu gişeye yönelmektedir. Gişe seçimleri numaratór tarafından sırayla yapılmaktadır. Örneđin ilk gelen kişi havaleyi seçerse numaratórde 1 nolu kasaya daha sonra gelen kişi fatura ödemelerini seçerse 5 nolu kasaya üçüncü kişi havale yi seçerse 2 nolu kasaya yönlendirilecektir. Bankaya aynı anda en fazla 50 kişi gelebilmekte ve banka günde 8 saat haftada 7 gün çalışmaktadır. Banka prensipleri geređi günlük 200 müşteriye hizmet vermezse çalışma süresini uzatmakta ve bu sayıya ulaşmaya çalışmaktadır ancak yapılan sözleşmeye göre de gişe görevlileri günde en fazla 10 saat çalışmaktadır. Sisteme girişler bittikten sonra banka görevlileri bankadaki müşterilerin işlemlerini bitirebilmek için maksimum 1 saat daha çalışmaktadırlar ve yine işlem için bekleyen müşteri var ise işlemlerini tamamlamadan sistemi terk etmektedirler.

Bankanın müşteri portföyü %25 havale için gelenler %40 fatura ödemeleri için gelenler %35 de hesap işleri için gelenlerden oluşmaktadır. Bunların işlem süresi sırasıyla (3:2),(4:1),(5:1) dakikadır. Bu problemin arena da simülasyonunu yapınız.(dağılım olarak normal dağılım kullanınız)

**Çözüm:** Sorunun arena çıktıları aşağıda verilmiştir.

08:30:00

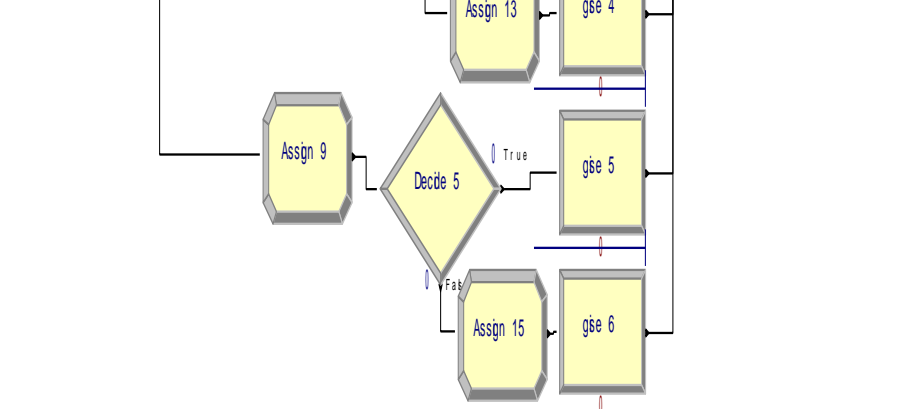


08:30:00

num arator

Decide 2

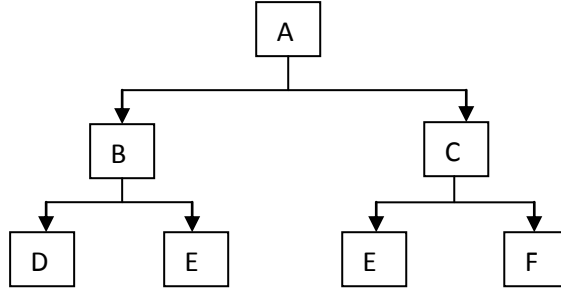
Entry.Type=-havab  
Entry.Type=-fatura



08:30:00

## 2.14. Fabrika Örneđi

Bir fabrikada üretilen ürünlere ait ürün ağacı aşağıdaki gibidir.



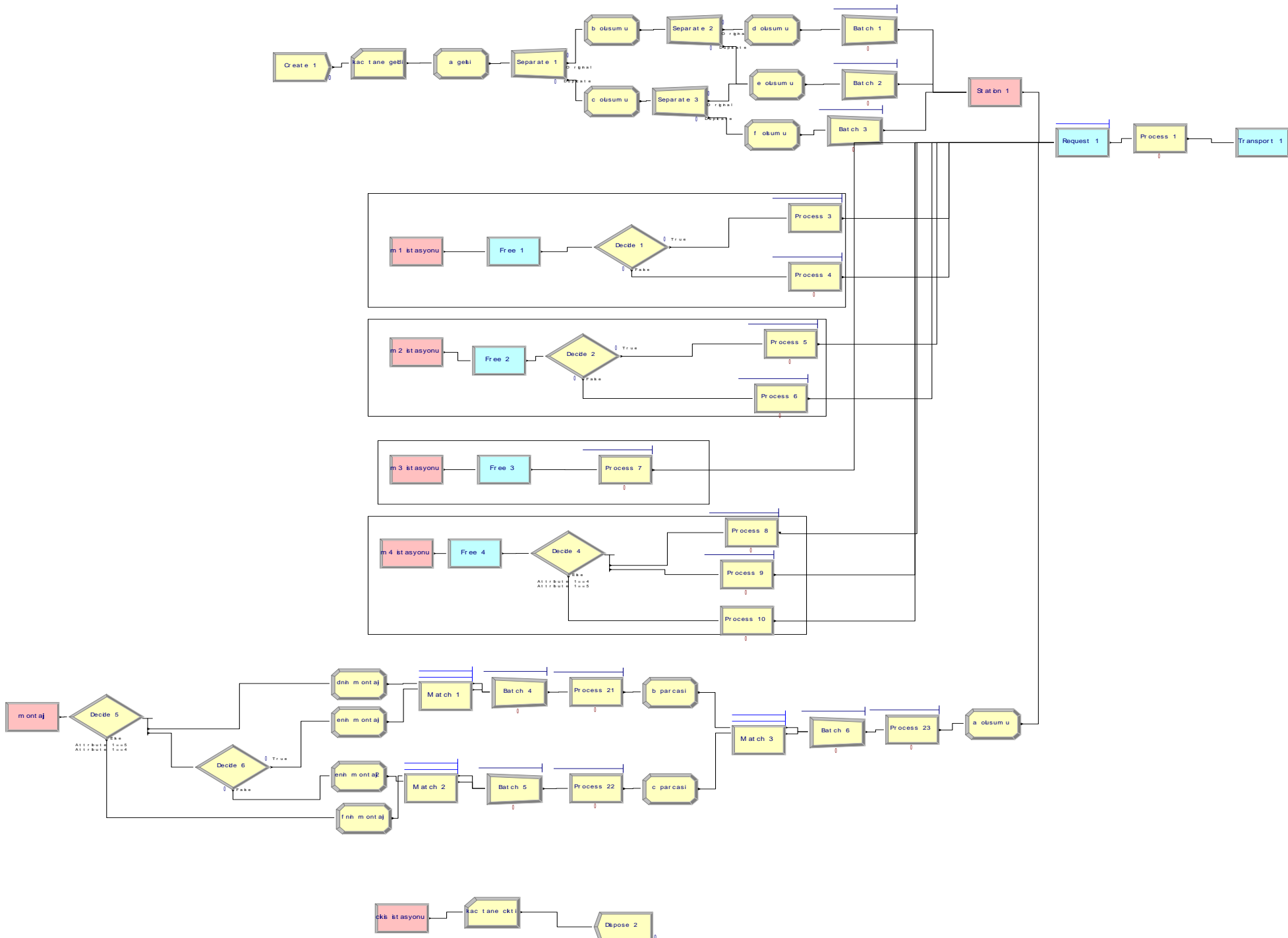
Fabrika yapılan 100 birimlik A ürünü üretimini tamamlamak istemektedir. Ürünlerin üretimi için atölye içerisinde izlemeleri gereken rotalar ile işlem süreleri aşağıda verilmiştir. Ürünler 10'arlı partiler halinde atölyede dolaşacak ve bir tezgahdaki partinin işi bitmeden diğer parti atölyeye girmeyecektir. Taşıma süreleri 5 birim alınırken montaj süreleri 0 birim olarak alınacaktır. Atölyede sadece bir taşıyıcı bulunmaktadır. Atölyede giriş çıkış aynı kapıdan olacak, giriş çıkış ile tezgahlar ve tezgahların kendi aralarındaki tüm mesafeler 10 birim olarak alınacaktır. Atölyeye kaç ürün geldiğini ve kaç ürün çıktığını bulan arena örneğini yapalım.

Rotalar: D → M1(3) - M2(5) - M4(7) \*Parantez içindekiler işlem süreleridir.

E → M2(4) - M3(9) - M4(4)

F → M1(11) - M4(3)

**Çözüm:** Problemin çözümü aşağıdadır.



## 2.15. Örnek Final Sorusu

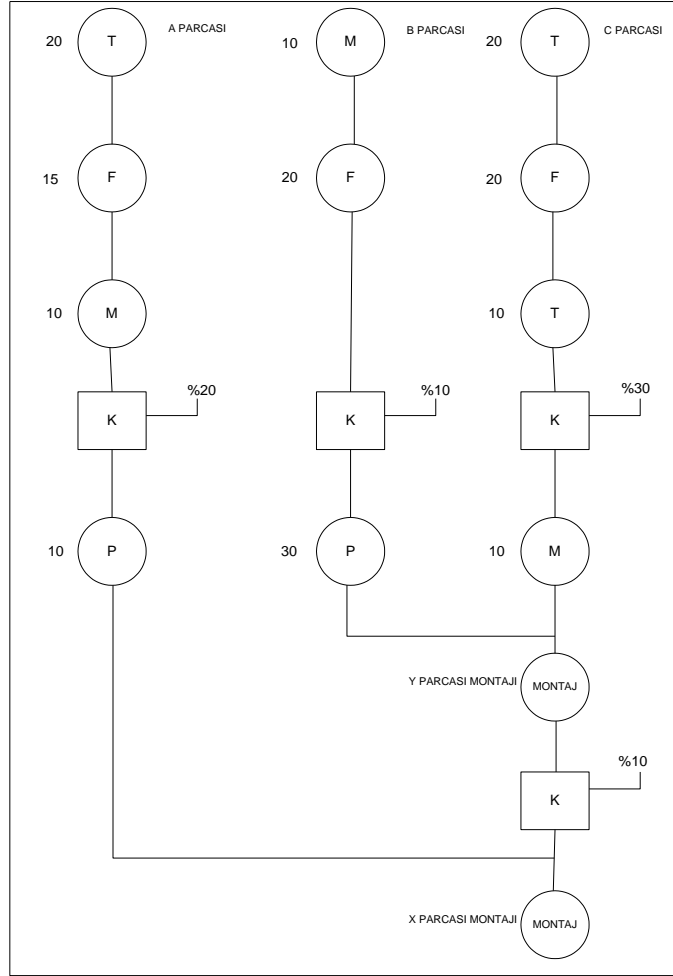
Bir işletme A, B, C ürünlerinin montajından X ürününü; B, C parçalarının montajından ise Y ürününü üretmek istemektedir. İşletme ayda 300 saat çalışmaktadır.

İşletmede A ve B ürünleri sürece göre yerleştirilmiş bir atölyede, C ürünü ise ürüne göre yerleştirilmiş bir atölyede imal etmektedir.

**NOT:** 1) C ürününün imalinde ikinci Torna tezgâhından sonra çıkan hatalı parçalar sürece göre yerleştirilmiş atölyedeki torna tezgâhlarında 10 dakikalık bir operasyonla %50 si kurtarılmakta ve ürüne göre yerleşimdeki matkap tezgâhına gönderilmektedir.

2) C ürününün imalatında konveyör kullanılmaktadır. Atölyede 2 adet forklift bulunmaktadır.

Ürünlere olan aylık talep miktarları ile ürünlerin üretiminde atölye içerisinde ne kadarlık partiler ile gezeceği aşağıda verilmiştir.



	X	Y	A	B	C
Aylık talep	50	10	10	5	5
Parti büyüklüğü	10	5	5	5	5

a) Arena ile sistemin simülasyonunu yapınız (Animasyonlu olarak).

b) Uygun tezgâh sayısının belirleyiniz.

## ÇÖZÜM

**Genel Bilgiler:** Simülasyon sürecinde kullanılan genel bazı bilgilerden bahsedilecektir.

Değer	Görevi	Değer	Görevi
<u>Attribute 1</u>	Parça Tipi	<u>Attribute 5</u>	Y Montajı için
<u>Attribute 2</u>	İşlem Süreleri	<u>Attribute 6</u>	Montaj sonrası batch için X veya Y kontrolü
<u>Attribute 3</u>	Arızalı ürünün C olup olmadığının kontrolü için	<u>Variable 1</u>	Montaj için sistemde A olup olmadığının kontrolü için
<u>Attribute 4</u>	X Montajı için	<u>Montaj süresi</u>	Norm(8,3)

- a) Problemin arena benzetim dili ile atelye atelye anlatımı aşağıda verilmiştir. Animasyonu ise ayrıca anlatılmıştır.

### **Sürece Göre Yerleşim Atölyeleri:**

Aşağıda sürece göre yerleşim yapılacak yerleşkenin atölyeleri ve bu atölyelerde uygulanan simülasyon yaklaşımı anlatılmıştır.

➤ *Sürece Göre Giriş Atelyesi:*

Bu atölyede A ve B girişlerinin sağlanmasının yanı sıra bu yarı ürünlere ait özellikler atanmıştır.

A Expo(5), B Expo(7) girişler arası süreye sahip girişler ile atölyeye girmektedir. Burada Create modülünde max arrivals A için 75 ve B için 80 alınmalıdır. Çünkü ilgili taleplerin tam karşılanması gerekiyor. Daha sonraki Assign modülünde ilgili gezen birim için gerekli attribute değerleri tanımlanmıştır. Batch modülünde ise Attribute 1 değerine göre 5'erli partiler oluşturulmuştur. Daha sonra ürünler Taşıyıcı Bölümüne gönderilir.

➤ *Taşıyıcı Bölümü:*

Burada taşıma işlemi transport ile sağlanmaktadır. Taşıma süresi olarak 0.1 br, taşıyıcıların hızları da 30 br alınmıştır. Ayrıca transport modülü için Destination Type By Sequence seçilmiştir.

➤ *Torna Atelyesi:*

Taşıyıcı ile gelen ürün Free modülü ile boşaltılır. C üretiminde oluşan hasarlı parçalar torna atelyesinde işleneceği için buraya gelen yarı ürünün C mi yoksa diğer ürünler mi olduğuna karar verilmelidir. Gelen ürün C ise direkt işlem göreceken A veya B ise önce Separate modülü ile partiler ayrılmalıdır. İşlemden

sonra yine ürünün C olup olmadığının kontrolü yapılır. Daha sonraki Decide modülü ile tornadan sonra ıskartaya ayrılan C miktarı belirlenir (2 way By Chance-Percent True 50). Eğer ürün arızalı C ise ıskarta sayısını belirlemek için Record modülü ile sayıldıktan sonra Dispose ile sistemi terk eder. Ürün sağlam C ise direkt transport ile ilgili atelyeye taşınır. Ürün A veya B ise birinci decide modülünden sonra 5'erlik partiler haline getirilip taşıyıcıya iletilir.

➤ *Freze Atelyesi:*

Taşıyıcı ile gelen ürünler boşaltıldıktan sonra ürünler partilerinden ayrıştırılır. Process ile gerekli işlemi görür ve ilgili ürünün B olup olmadığının kontrolü yapılır. Çünkü B ürünü freze işleminden sonra kalite kontrole girmekte ve %10'u ıskartaya çıkmaktadır. Eğer ürün B ise ikinci Decide modülü ile B'nin kalite kontrol işlemi gerçekleştirilir. Ürün A ise direkt olarak Batch modülü ile %5'erli partiler halinde taşıyıcı bölümüne iletilir. Ürün ıskartaya ayrılacak bir B ise record komutu ile sayıldıktan sonra sistemden çıkarılır, sağlam B ise partiler halinde taşıyıcıya iletilir.

➤ *Matkap Atelyesi:*

Ürünler atelyeye boşaltıldıktan sonra partiler ayrıştırılır ve ürünler işleme girerler. Matkap işlemi sonrası A ürününün kalite kontrol faaliyeti için ürünün A olup olmadığı kontrol edilir. Ürün A değilse partiler halinde taşıyıcıya iletilir. Eğer ürün A ise %20'lik bir ıskarta oranı ile Decide modülünden geçer. Ürün ıskartaya ayrılacaksa Record modülü ile sayıldıktan sonra sistemi terk eder. Sağlam A ise partiler haline getirilip taşıyıcıya iletilir.

➤ *Planya Atelyesi:*

Ürünler taşıyıcıda indirilip ayrıştırıldıktan sonra gerekli işlemi görürler ve 5'erli partiler halinde taşıyıcıya iletilirler.

➤ *Montaj Atelyesi:*

Atelye girişine gelen ürünün C olup olmadığı kontrol edilir. Çünkü ürün C ise Exit modülü ile konveyörden alınacakken A veya B ise Free ile Transport modülünden alınacaktır. Ürün C ise Decide 19 (Percent True=8,4) modülü ile montaja gönderilecek ve üretilecek C ayırımı yapılır. Eğer ürün A veya B ise partiler ayrıştırıldıktan sonra Attribute 4 (X Montajını gerçekleştirecek attribute değeri) 1 olarak atanır. Montajın X mi yoksa Y mi olacağı sistemde A ürününün bulunup bulunmadığına bağlıdır. Bu yüzden bu aşamada ürünün Decide 12 modülü ile A mı olup olmadığı kontrol edilir. Ürün A ise önce Decide 17 (Percent True=16,6) modülü ile üretilecek ve montaja gönderilecek A ürünü sayısı belirlenir, sonra sistemde A olduğunu göstermesi için atanan Variable 1 değeri 1 artırılır ve Match 5 (Attribute Name=Attribute 4) modülü ile X montajına gönderilir. Ürün A değilse yani B ise önce sistemde üretilecek ve montaja gidecek B sayısını belirlemek için Decide 18 modülü ile (Percent True=91,6) B' nin bir kısmı taşıyıcıya iletilirken bir kısmı montaja gönderilir, sonra yine sistemde A olup olmadığının kontrolü için Variable 1 değerinin sıfırdan büyük olup olmadığı kontrol edilir (Decide 14). Eğer sistemde A varsa Match 5 modülüne yoksa Assign 8 modülüne gönderilerek Y montajı için Attribute 5 değeri 1 atanır ve Match 4 modülüne gönderilir. Batch 10 ve 11 modülleri ile montaj gerçekleştirilir.

Assign 11-12 modülleri ile montaj sonrası parti miktarları ayrı olduğu için bunu belirleyecek Assign 6 modülleri atanır. Process 13 ile montaj işlemi süresi ile gerçekleştirilir. Assign 10 modülü daha önce artırdığımız Variable 1 değerini 1 azaltır, sistemden bir A ürünü çıktığı için. Decide 15 ile parti miktarlarını belirleyecek Attribute 6 değeri değerlendirilmiştir. Ürün Y ise Decide 16 ile montaj sonrası % 10'luk ıskarta sayısı belirlenmiştir. Daha sonra batch 12-13 modülleri ile partiler halinde ürünler taşıyıcıya taşınmıştır.

### **Ürüne Göre Yerleşim:**

Aşağıda ürüne göre yerleşim yapılan konveyör hattı segmentleri ile anlatılmıştır.

- Bu hat Expo(6) ve Max Arrivals 103 olan Create komutu ile başlamaktadır. Bu hatta Access modülü ile ürünler konveyör hattına alınmakta Convey modülü ile ilgili birime taşınmakta ve Exit modülü ile ilgili tezgâha geldiğinde hattan alınmaktadır.

Bu hatta dikkat edileceklerden biri ikinci torna tezgâhından sonraki kalite kontrol işlemidir. Bu arda Decide modülü ile %30 C ürünün ayrılır. Eğer ürün hasarlı ise Assign 4 komutu ile Attribute 3 değeri 2 ve sequence numarası farklı bir rota izleyeceği için 4 olarak atanır. Daha sonra taşıyıcıya iletilir. Ürün hasarlı değilse Access ve Convey modülleri ile olağan akışına devam etmektedir.

Dikkat edilecek bir diğer nokta ise matkap işlemine gelen ürünün ikinci işlemi görüp gelen C mi yoksa sağlam C mi olduğudur. Çünkü ikinci işlemden gelen C transport ile gelirken sağlam C hattan gelmektedir. Decide komutu ile (Attribute 3 ==2) ürünün ikinci işlemden gelip gelmediği kontrol edilmiş; eğer ikinci işlem ürünü ise Free, değilse Exit edilerek tezgâha alınmıştır. Daha sonra gerekli işlemleri görmüş ve hattaki olağan akışı devam etmiştir.

### **Çıkış atelyesi:**

Bu atelyede gelen ürünler free modülü ile taşıyıcıdan alınarak sistemi terk etmektedir. Record modülü ile sistemden çıkan parti miktarı sayılmaktadır.

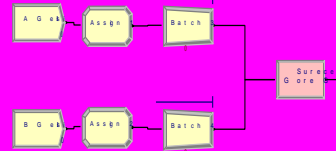
b) Sistemin Arena Benzetim dili ile animasyonu yapılmış ve aşağıda verilmiştir.

**NOT:** Aşağıda verilen şekiller ve sistemin nasıl çalıştığı rapor ile verilecek olan CD içerisinde bulunan Arena Program çıktısı ile çalıştırılıp görülebilir. Ayrıca sisteme ait sonuçlar rapora ek olara verilecekse de program çalıştırılıp da görülebilir.

Çıktılar CD' de bulunan "Ödev 14 Animasyonlu" programından alınmıştır. Asıl akış şeması "Ödev 14 Animasyonlu2" programıdır. Fakat deneme sürümü Arena kullanıldığı için kullanılması izin verilen maksimum modül sayısına ulaşıldığından ARENA programının çalışmasına izin vermemektedir. Ayrıca bu programın kaydedilmesine de izin verilmediği için sadece aşağıda akış şekli olarak verilmiştir. Montaj atelyesine eklenen Decide 16 modülünden sonraki modüller limiti aşmaktadır.

# AKIŞ ŞEMASI

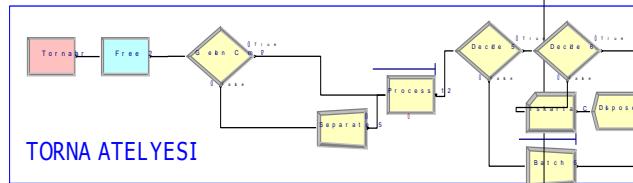
## SURECE GORE GIRIS ATELYESI



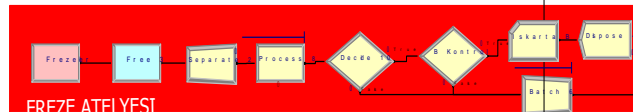
## TASIYICI BOLUMU



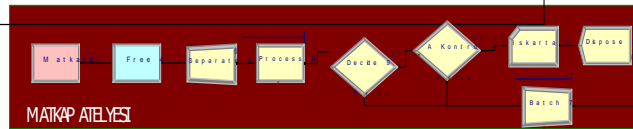
## TORNA ATELYESI



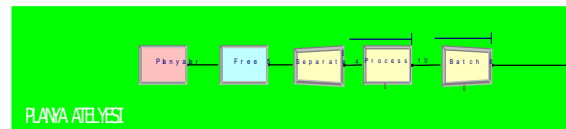
## FREZE ATELYESI



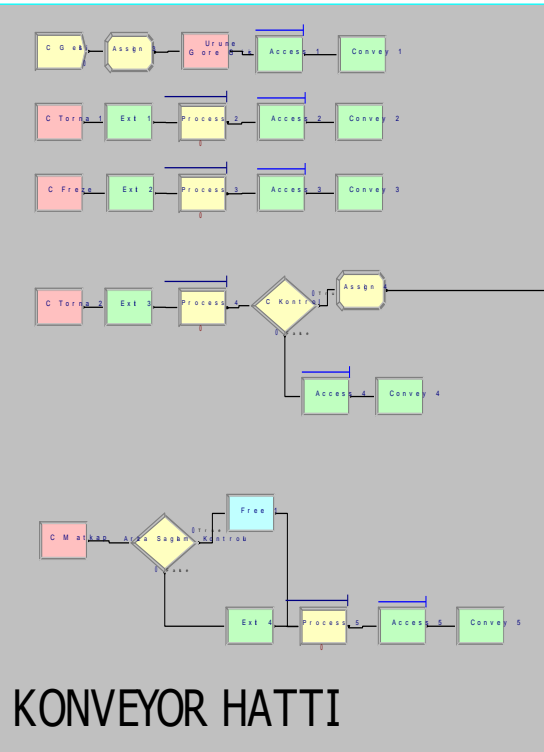
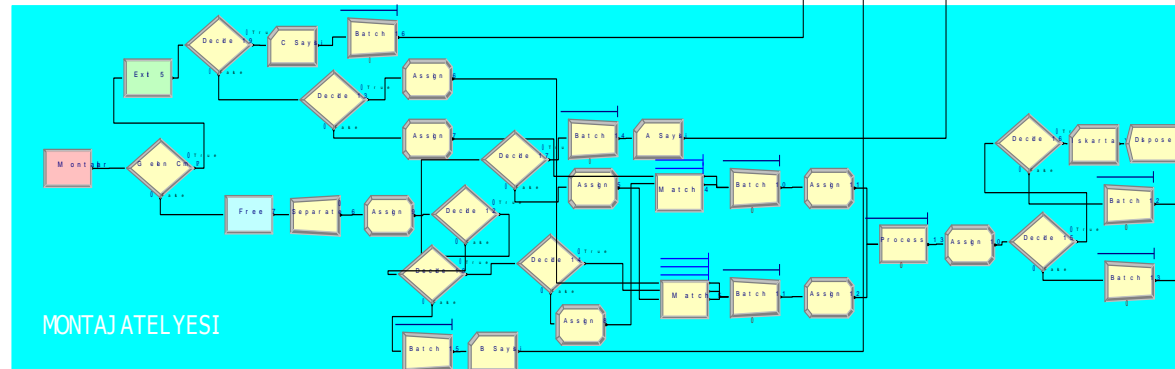
## MATKAP ATELYESI



## PLANA ATELYESI

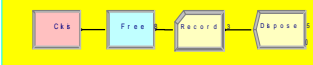


## MONTAJ ATELYESI

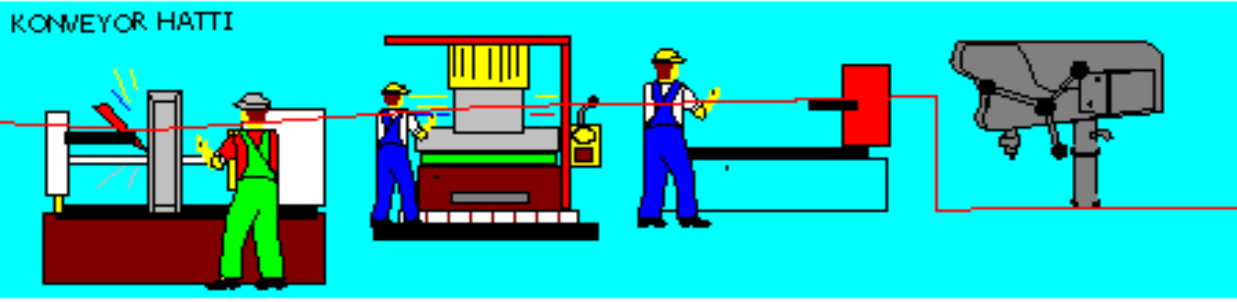


## KONVEYOR HATTI

## CIKIS

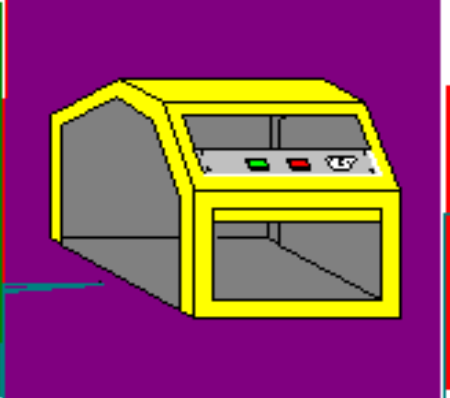
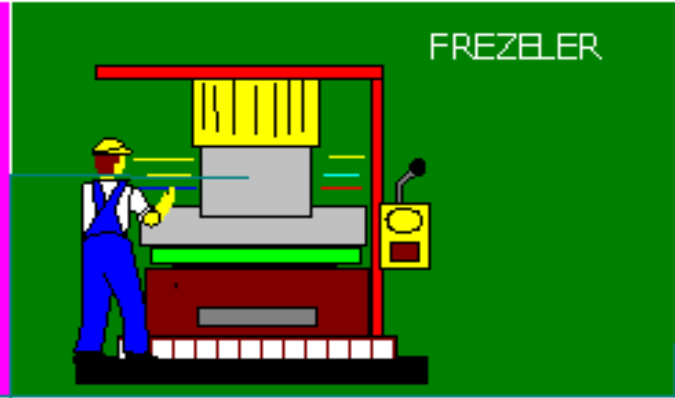
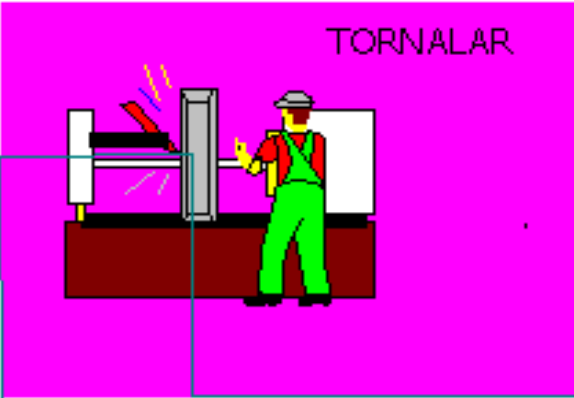


UG Giriş

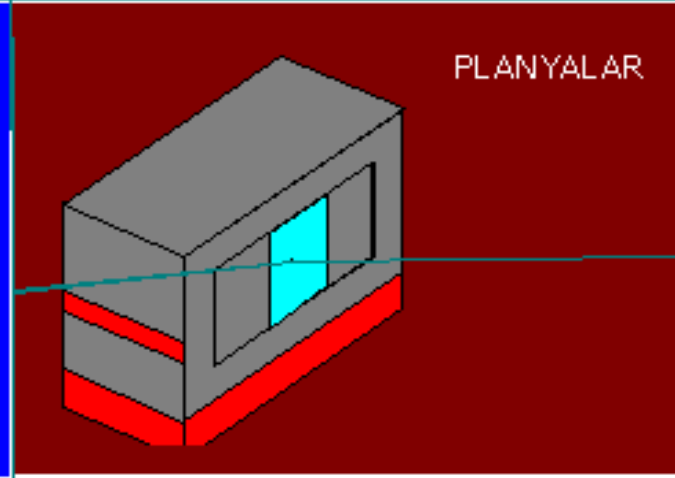
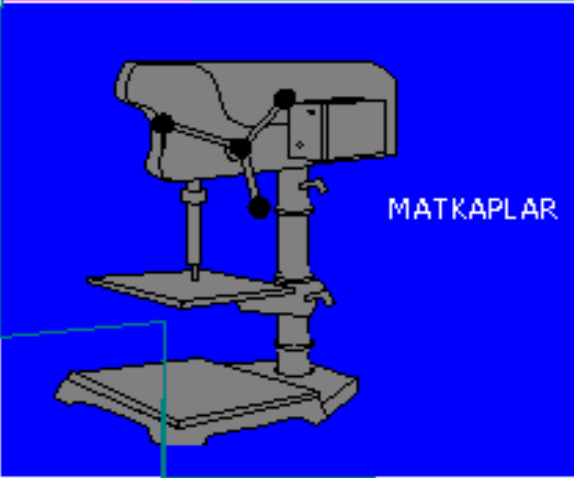


MONTAJ

9GS Giriş



ÇIKIŞ



### 3. SONUÇ

Raporun hazırlanma amacı 2006-2007 öğretim yılı benzetim dersi kapsamında öğrenilen komutları tekrar etmek ve Benzetim Dilleri dersine ön hazırlık yapmaktır. Komutlar hakkında sadece temel bilgiler vermeye çalışılmış fazla detaylara girilmemiştir. Tüm komutlar hakkında yol gösterici olması açısından örnekler geliştirilmeye çalışılmıştır. Verilen örnekler tüm komutları kapsamasa da tekrar anlamında diğer komutların da nasıl çalıştırıldığının hatırlanmasını sağlamıştır.

Yapılan çalışmalar ve araştırmalar ışığında Benzetim Dilleri dersinin daha verimlilik seviyesi yüksek geçeceği kanaatindeyim.