

T.C.
CELAL BAYAR ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
DOKTORA TEZİ

144705

İMALATTA SİMÜLASYON KULLANIMI

VE

“ARENA” DİLİ İLE BİR UYGULAMA

MUSTAFA GERŞİL

144705

DANIŞMAN ÖĞRETİM ÜYESİ
PROF. DR. HÜLYA TÜTEK

MANİSA
2004

**YÜKSEKÖĞRETİM KURULU DOKÜMANTASYON MERKEZİ
TEZ VERİ FORMU**

Tez No:

Konu:

Üniv.Kodu:

Not: Bu bölüm merkezimiz tarafından doldurulacaktır.

Tezin yazarının

Soyadı: GERŞİL

Adı: MUSTAFA

Tezin Türkçe adı: İMALATTA SİMÜLASYON KULLANIMI VE ARENA DİLİ İLE BİR UYGULAMA

Tezin Yabancı adı: SIMULATION USE IN MANUFACTURING AND AN APPLICATION WITH ARENA LANGUAGE

Tezin yapıldığı Üniversite: CELAL BAYAR ÜNİVERSİTESİ Enstitü: SOSYAL BİLİMLER ENSTİTÜSÜ Yılı: 2004

Diğer kuruluşlar:

Tezin Türü: DOKTORA

Dili :TÜRKÇE

Sayfa sayısı :213

Referans sayısı:117

Tez Danışmanının

Ünvanı: Prof.Dr.

Adı: HÜLYA

Soyadı: TÜTEK

Türkçe anahtar kelimeler:

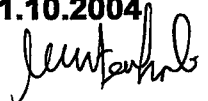
İngilizce anahtar kelimeler:

- 1- SİMÜLASYON
- 2- MODELLEME
- 3- ARENA
- 4- İMALAT
- 5-

- 1- SIMULATION
- 2- MODELING
- 3- ARENA
- 4- MANUFACTURING
- 5-

Tarih:11.10.2004

İmza :



ÖZET

Karmaşıklığın, dinamizmin ve değişimin egemen olduğu dünyada, sistemin davranışı ile performansa etki eden parametrelerin anlaşılması oldukça önemlidir. Bu durum genel olarak, karmaşıklık ve değişimi simgeleyen faaliyetleri içeren süreçlerin işlemlerinin gelişiminde gözlemlenebilir. Bu karmaşık, dinamik gerçekliği temsil edebilmek, analiz ve değerlendirmesini yapabilmek için bir modele olan ihtiyaç tanımlanmalıdır.

Modeller, imalat sistemi tasarımı ve işlevsel konuların analizinde geniş olanaklar sunar ve bu nedenle imalat sistemi tasarımı sürecinin bir çok aşamasında önemli araçlardır. Yeterli deneysel kanıtın olması durumunda ki bu tezin de belirttiği gibi, imalat endüstrisinde en güçlü modelleme tekniklerinden birisi simülasyondur.

Simülasyon, gerçek sistemin bir modelinin oluşturulması, sistemin davranışının incelenmesi ve farklı stratejiler altında nasıl işlediğini değerlendiren bir süreçtir. Böylece simülasyon, sistemlerin tanımlanması, değerlendirilmesi, analizini sağlar ve eğer başarılı bir şekilde uygulanabilirse, örgütsel çerçeveyi aşarak sistemdeki işbirliğini desteklemek suretiyle örgütte bilgi paylaşımı ve haberleşmeyi geliştirir.

Bu bağlamda, simülasyon sistem bilgisi birikimini önemli bir biçimde arttırabilir, sistemdeki hazırlık sürelerini kısaltabilir, kullanım oranı ve verimliliği arttırabilir ve örgütsel karar vermeyi büyük ölçüde destekleyebilir. Simülasyon, aynı zamanda performans değerlerinin izlenebilirliğini artırır ve örgüt içerisinde yer alan kişilere bu değerlerin ne kadar önemli olduğunu ifade eder.

Aslında simülasyon, doğada dinamik yapıda bulunan gerçek performans parametrelerini modelin analizinde kullanıp, stokastik yada deterministik modellere uygulayarak yeni bir imalat sistemi tasarımı ya da işlemler yönetimi politikasının değerlendirilmesinde temel oluşturur. Bunun amacı oldukça basittir: imalat sürecinde doğru kararları almak suretiyle bu kararların kalitesini arttırmaktır.

Bu tezde, stokastik simülasyon modeli ile bir televizyon üretim tesisinde mevcut montaj hatlarından birisindeki çıktı miktarını arttırmak ve darboğaz sürecinin uygulanması ile hem kuyrukta bekleme sürelerini ve toplam sistem zamanını azaltmak amaçlanmıştır. Siman programlama dili temeline dayalı Arena dili kullanılarak, bir montaj bandı sisteminin modellemesi yapılış ve farklı işletim politikası önerilerinin modele uygulanması ile, üretim bandındaki çıktı miktarında %10 artış sağlanmıştır.

ABSTRACT

In a world where complexity, dynamism, and change dominate, it becomes vital to understand systems behavior and the parameters that affect performance. This situation can be generally observed in the development of manufacturing processes; activities in themselves characterized by complexity and change. To represent, analyze and evaluate this complex, dynamic reality; the need for a model should be recognized.

Models address a wide range of manufacturing system design and operational issues and are therefore essential tools in many facets of the manufacturing system design process. As seen from ample empirical evidence and as this thesis will show, one of the most powerful modeling techniques in the manufacturing industry is simulation.

Simulation is defined as the process of building a model of a real system and conducting experiments on this model for the purpose of understanding the behaviour of the system and evaluating various strategies for the operation of the system. Simulation thus provides analysis, description and evaluation capabilities of systems, and if successfully applied can support collaborative work across organizational boundaries and thereby improve information and communication.

By these means, simulation can significantly improve system knowledge, system lead time, increase utilization and productivity and support decision making throughout an organization. Simulation also increases the awareness of performance measurements and emphasizes the importance of those measures to the people involved in the organization.

In fact, simulation is fundamental to the assessment of a new manufacturing system design or operations management policy since by implementing stochastic or deterministic models, using real dynamic performance parameters. Its purpose here is simple: to support correct decisions throughout the manufacturing process, thereby increasing the quality of those decisions.

In this thesis, the aim is to increase the throughput of one of the assembly lines of a television production plant by determining the bottleneck process and minimizing both the queueing times and the total system time. Using Arena simulation language based on Siman, the assembly line is modelled and tested utilizing different operation policies and increase in production by 10% is realized.

YEMİN METNİ

Doktora tezi olarak sunduđum "İmalatta Simülasyon Kullanımı ve Arena Dili ile bir Uygulama" adlı çalışmanın, tarafımdan bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın yazıldığını ve yararlandığım eserlerin bibliyografyada gösterilen eserlerden oluştuđunu, bunlara atıf yapılarak yararlanmış olduğumu belirtir ve bunu onurumla doğrularım.

11/10/2004



Mustafa GERŞİL

TEZ SAVUNMA SINAV TUTANAĞI

Celal Bayar Üniversitesi Sosyal Bilimler Enstitüsü 11/02/2005 tarih ve 3/21 sayılı toplantısında oluşturulan jürimiz tarafından Lisans Üstü öğretim Yönetmeliği'nin 24. Maddesi gereğince Enstitümüz İşletme Anabilim Dalı Doktora Programı öğrencisi Mustafa GERŞİL'in "İmalatta Simulasyon Kullanımı ARENA dili ile bir uygulama " Konulu tezi incelenmiş ve aday 25/02/2005 tarihinde saat 14.00 de jüri önünde tez savunmasına alınmıştır.

Adayın kişisel çalışmaya dayanan tezini savunmasından sonra 30 dakikalık süre içinde gerek tez konusu, gerekse tezin dayanağı olan anabilim dallarından jüri üyelerine sorulan sorulara verdiği cevaplar değerlendirilerek tezin,

BAŞARILI olduğuna	<input checked="" type="checkbox"/>	<u>OY BİRLİĞİ</u>	<input checked="" type="checkbox"/>
DÜZELTME yapılmasına	<input type="checkbox"/> *	<u>OY ÇOKLUĞU</u>	<input type="checkbox"/>
RED edilmesine	<input type="checkbox"/> **	ile karar verilmiştir.	

* Bu halde adaya 3 ay süre verilir.

** Bu halde adayın kaydı silinir.


BAŞKAN

Prof. Dr. Hülya TÜTEK
(Danışman)



ÜYE

Prof. Dr. Cengiz YILMAZ

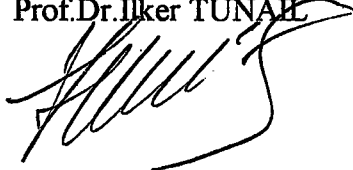


ÜYE

Prof. Dr. Semra ÖNCÜ

ÜYE

Prof. Dr. İlker TUNAİL



ÜYE

Doç. Dr. Altay Uğur GÜL



Evet Hayır

*** Tez, burs, ödül veya Teşvik prog. (Tüba, Fullbright vb.) aday olabilir

Tez, mutlaka basılmalıdır

Tez, mevcut haliyle basılmalıdır

Tez, gözden geçirildikten sonra basılmalıdır. ^{abilir}

Tez, basımı gereksizdir.

İÇİNDEKİLER

YÜKSEKÖĞRETİM KURULU DÖKÜMANTASYON MERKEZİ

TEZ VERİ FORMU.....	i
ÖZET.....	ii
ABSTRACT.....	iii
YEMİN METNİ.....	iv
TEZ SAVUNMA TUTANAĞI.....	v
TABLolar LİSTESİ.....	xi
ŞEKİLLER LİSTESİ.....	xii
ÖNSÖZ.....	xiv
GİRİŞ.....	1

BİRİNCİ BÖLÜM

SİMÜLASYON KAVRAMI

1.1 SİMÜLASYONA GİRİŞ.....	4
1.2 SİMÜLASYONUN AMACI VE ÖZELLİKLERİ.....	8
1.2.1 Simülasyonun Özellikleri.....	9
1.2.2 Simülasyon Kullanımının Uygun Olduğu Durumlar.....	10
1.3.SİMÜLASYONUN AVANTAJ VE DEZAVANTAJLARI.....	11
1.3.1 Simülasyonun Avantajları.....	11
1.3.2 Simülasyonun Dezavantajları.....	12
1.4 İMALATTA SİMÜLASYON KULLANIMI.....	13
1.5 SİMÜLASYON MODELİNİN YAPISI.....	18
1.6 SİMÜLASYON MODELİNİN NİTELİKLERİ.....	21
1.7 BİLGİSAYARLI SİMÜLASYONUN TEMELLERİ.....	22
1.7.1 Bilgisayar Simülasyonunun Karakteristikleri.....	23
1.7.2 Simülasyon Modellerinin Sınıflandırılması.....	24
1.7.2.1 Statik ve Dinamik Simülasyon Modelleri.....	25

1.7.2.2	Stokastik (Olasılıklı) ve Deterministik (Belirli) Simülasyon Modelleri.....	25
1.7.2.3	Kesikli (Discrete) Simülasyon Modelleri.....	26
1.7.2.3.1	Kesikli Simülasyonda Zaman İlerletme Mekanizması.....	27
1.7.2.3.2	Kesikli Simülasyonun Elemanları.....	28
1.7.2.4	Sürekli (Continuous) Simülasyon.....	29

İKİNCİ BÖLÜM

SİMÜLASYON VE MODELLEME

2.1	EL İLE SİMÜLASYON (MONTE CARLO SİMÜLASYONU).....	32
2.2	VERİ TOPLAMA VE ANALİZİ.....	39
2.3	TESADÜFİ SAYILARIN TÜRETİLMESİ.....	40
2.4	RASTGELELİK TESTLERİ.....	41
2.5	İSTATİSTİKSEL DAĞILIMLAR VE UYGUNLUK TESTLERİ.....	42
2.5.1	Dikdörtgen Dağılım.....	43
2.5.2	Üçgen Dağılımı.....	44
2.5.3	Geometrik Dağılım.....	45
2.5.4	Hipergeometrik Dağılım.....	46
2.5.5	Bernoulli Dağılımı.....	46
2.5.6	Binom Dağılımı.....	47
2.5.7	Poisson Dağılımı.....	47
2.5.8	Normal Dağılım.....	48
2.5.9	Gamma Dağılımı.....	49
2.5.10	Üstsel Dağılım.....	49
2.5.11	Erlang Dağılım.....	51
2.5.12	Dağılımın Uygunluğunun Testi.....	52
2.6	SİMÜLASYONDA MODELLEME YAKLAŞIMLARI.....	54
2.7	SİMÜLASYON AŞAMALARI.....	57
2.7.1	Problemi Tanımlama.....	58
2.7.2	Girdilerin Toplanması ve İstatistiksel Analizi.....	58

2.7.3 Yazılımı Seçme.....	60
2.7.4 Simülasyon Modelini Geliştirme.....	61
2.7.5 Simülasyon Modelini Doğrulama(Verification).....	62
2.7.5.1 Doğruluğu Sınama Teknikleri.....	65
2.7.6 Simülasyon Modelinin Geçerliliğini Test etme(Validation).....	66
2.7.7 Simülasyon Deneylelerinin Tasarımı.....	68
2.7.8 Çıktıların İstatistiksel Analizi.....	69
2.8 VARYANS AZALTMA TEKNİKLERİ (VT).....	71

ÜÇÜNCÜ BÖLÜM

SİMÜLASYON DİLLERİ VE KULLANIM ALANLARI

3.1 SİMÜLASYON DİLLERİNİN SINIFLANDIRILMASI.....	75
3.1.1 Simülasyon Yazılımları.....	75
3.1.1.1 Simülasyon Yazılımı Seçimi.....	77
3.1.1.2 Simülasyon Paketleri.....	78
3.1.1.2.1 .Arena Simülasyon Paketi.....	78
3.1.1.2.2 .AutoMod.....	79
3.1.1.2.3 Quest.....	80
3.1.1.2.4 Extend.....	80
3.1.1.2.5 Microsaint.....	81
3.1.1.2.6 Promodel.....	81
3.1.1.2.7 Taylor Enterprise Dynamics.....	82
3.1.1.2.8 .Witness.....	82
3.1.1.2.9 GPSS.....	83
3.1.1.2.10 Siman.....	84
3.1.1.2.11 Simscript.....	85
3.1.1.2.12 Slam.....	85
3.1.2 Simülatörler.....	86
3.3 SİMÜLASYON DİLLERİNİN KARŞILAŞTIRILMASI.....	87
3.4 İMALAT SİSTEMLERİNİN SİMÜLASYONUNDA GELİŞEN YAKLAŞIMLAR.....	89

3.4.1 Paralel ve Dağıtılmış Sistemlerinin Simülasyonu.....	89
3.4.2 Yapay Zeka ile Modelleme ve Simülasyon.....	91
3.4.2.1 İnsan Gibi Düşünen Sistemler.....	94
3.4.2.2 İnsan Gibi Davranan Sistemler.....	94
3.4.2.3 Rasyonel Düşünen Sistemler.....	94
3.4.2.4 Rasyonel Davranan Sistemler.....	95
3.4.2.5 Yapay Zeka Türleri.....	95
3.4.2.5.1 Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler...	96
3.4.2.5.2 Doğal Diller.....	96
3.4.2.5.3 İnsan Algılama Yeteneklerinin Benzetimi.....	97
3.4.2.5.4 Robotikler.....	99
3.4.2.5.5. Sinirsel Ağlar.....	100
3.4.2.5.6 Bulanık Mantık.....	100
3.4.2.5.7 Sanal Gerçeklik.....	101
3.4.3 Nesneye Yönelik Programlama.....	102

DÖRDÜNCÜ BÖLÜM

UYGULAMA: VESTEL ELEKTRONİK A.Ş DE BİR ÜRETİM BANDI SİMÜLASYONU

4.1 ŞİRKETİN TANITIMI.....	104
4.2 ÜRETİM HATTININ TANITIMI.....	111
4.3 PROBLEMİN TANITIMI.....	113
4.4 MODELİN HAZIRLANMASINDA YAPILAN VARSAYIMLAR.....	118
4.5 MODELİN KURULMASI.....	119
4.6 MODELDE KULLANILAN ARENA ELEMANLARI.....	119
4.7 VERİLERİN (İŞLEM SÜRELERİNİN) HAZIRLAMASI.....	127
4.8 MODELİN DOĞRULANMASI VE GERÇEKLENMESİ.....	134
4.9 MODELİN SONUÇLARI.....	135
4.10 SİSTEMİN ELEMENTER MODELİ.....	151
4.10.1 Bir Makinenin Genel Davranışı.....	151
4.10.2 Create Bloğu.....	152
4.10.3 Bir İşlemin Sistem İçinde Sona Erme Durumu.....	153

4.10.4	Attributes Elemanı.....	153
4.10.5	Tallies Elemanı.....	154
4.10.6	Sequences Elemanı.....	155
4.10.7	Queues Elemanı.....	159
4.10.8	Resources Elemanı.....	160
4.10.9	Dstats Elemanı.....	160
4.10.10	Stations Elemanı.....	161
4.10.11	Route Bloğu.....	162
DEĞERLENDİRME VE SONUÇ.....		163
KAYNAKÇA.....		166
EK 1	: Arena 2.2 Simülasyon Programı Tanıtımı.....	178
EK 2A	: Mevcut Durum İçin Makinelerin Dolu, Boş Ve Arıza İle Geçen Zamanları Ve Yüzdeleri.....	184
EK 2B	: Öneri 1 İçin Makinelerin Dolu, Boş Ve Arıza İle Geçen Zamanları Ve Yüzdeleri.....	188
EK 2C	: Öneri 2 İçin Makinelerin Dolu, Boş Ve Arıza İle Geçen Zamanları Ve Yüzdeleri.....	192
EK 3	: Program.....	196

TABLolar LİSTESİ

Tablo 1	Sistem Değişkenlerinin Birbirleri İle Olan İlişkisi.....	19
Tablo 2	Kümülatif Olasılıkların Ve Monte Carlo Sayılarının Belirlenmesi.....	34
Tablo 3	A Planı İçin Rulman Bozulma Süreleri.....	35
Tablo 4	A Planı İçin Maliyetler.....	36
Tablo 5	B Planı İçin Bozulma Süreleri.....	37
Tablo 6	B Planı İçin Bozulma Maliyeti.....	37
Tablo 7	C Planı İçin Maliyetler.....	38
Tablo 8	A, B ve C planları için Karşılaştırmalı Toplam Maliyetler.....	38
Tablo 9	Temel Simülasyon Özelliklerinin Gerçekleştirilmesi.....	88
Tablo 10	Simülasyon Dillerinin Karşılaştırılması.....	88
Tablo 11	Renkli Tv Üretim Miktarları (Adet).....	107
Tablo 12	Mevcut Durum İçin Tamir Ve Arıza İstatistikleri.....	122
Tablo 13	Makinelerin Çalışma Zamanlarının İstatistikleri.....	127
Tablo 13.A	Perçin Analizi.....	128
Tablo 13.B	Jumper Analizi.....	129
Tablo 13.C	Sequencer Analizi.....	130
Tablo 13.D	Axial Analizi.....	131
Tablo 13.E	Radial Analizi.....	132
Tablo 13.F	Smd Analizi.....	133
Tablo 14	Mevcut Durum İçin Kuyrukta Ortalama Bekleme Süreleri.....	138
Tablo 15	Mevcut Durum İçin Kuyruktaki Ortalama Varlık Sayısı.....	139
Tablo 16	Öneri 1 İçin Tamir Ve Arıza İstatistikleri.....	140
Tablo 17	Öneri 1 İçin Kuyrukta Ortalama Bekleme Süreleri.....	141
Tablo 18	Öneri 1 İçin Kuyruktaki Ortalama Varlık Sayısı.....	142
Tablo 19	Öneri 2 İçin Arıza Ve Tamir İstatistikleri.....	145
Tablo 20	Öneri2 İçin Kuyrukta Ortalama Bekleme Süreleri.....	147
Tablo 21	Öneri2 İçin Kuyruktaki Ortalama Varlık Sayısı.....	148
Tablo 22	Öneri 2 (Son) İçin Kuyrukta Ortalama Bekleme Süreleri.....	149
Tablo 23	Öneri 2 (Son) İçin Kuyruktaki Ortalama Varlık Sayısı.....	150

ŞEKİLLER LİSTESİ

Şekil 1	Kesikli Simülasyonda durum değişkeni.....	26
Şekil 2	Sürekli simülasyon durum değişkeni.....	31
Şekil 3	Dikdörtgen Dağılım.....	43
Şekil 4	Üçgen Dağılım.....	44
Şekil 5	Normal Dağılım.....	48
Şekil 6	Gamma Dağılımı.....	49
Şekil 7	Üstsel Dağılım.....	50
Şekil 8	Erlang Dağılımı.....	51
Şekil 9	TV Üretim Akış Şeması.....	112
Şekil 10	Otomatik Dizgi Bölümündeki Makine Yerleşim Planı.....	115
Şekil 11	Otomatik Dizgi Bölümünde İş akışı.....	116
Şekil 12	Simülasyon Akış Diyagramı.....	121
Şekil 13	Arena Modelinin Genel İş Akışı.....	123
Şekil 14	Perçin Ve Jumper Gruplarının Yerleşimi.....	124
Şekil 15	Sequencer Ve Axial Gruplarının Yerleşimi.....	125
Şekil 16	Radial Ve Smd Gruplarının Yerleşimi.....	126
Şekil 17	Simülasyon Çıktılarının Korelasyon Diyagramı.....	136
Şekil 18	Simülasyon Replikasyon Raporu.....	137
Şekil 19	Arena Simülasyon Programının Blok Ve Elemanları.....	151
Şekil 20	Bir Makinenin Genel Davranışı.....	151
Şekil 21	Create Bloğu.....	152
Şekil 22	Bir İşlemin Sistem İçinde Sona Ermesi.....	153
Şekil 23	Attributes Elemanı.....	154
Şekil 24	Tallies Elemanı.....	154
Şekil 25	Sequences Elemanı.....	155
Şekil 26	Perçin Makine Grubu.....	156
Şekil 27	Jumper Makine Grubu.....	157
Şekil 28	Sequencer Makine Grubu.....	157
Şekil 29	Axial Makine Grubu.....	158
Şekil 30	Radial Makine Grubu.....	158

Şekil 31	Smd Makine Grubu.....	159
Şekil 32	Queues Elemanı.....	159
Şekil 33	Resources Elemanı.....	160
Şekil 34	Dstats Elemanı.....	161
Şekil 35	Stations Elemanı.....	161
Şekil 36	Route Elemanı.....	162



ÖNSÖZ

Bilgisayar simülasyonu, yüksek teknoloji bilgisayarların son yıllarda yaygınlaşmasıyla, sistem dizaynı ve analizinde, hızla popüler bir araç olmuştur. Simülasyon, mühendis ve planlamacılara bir sistemin dizaynı ve işletimiyle ilgili, zamanında ve rasyonel kararlar vermeleri için yardımcı olur. Simülasyon, tek başına problemleri çözemez, fakat problemi açıkça tanımlar ve sayısal olarak alternatif çözümleri değerlendirir. Önerilen herhangi bir çözüm için sayısal ölçüm ve analiz yapabilir ve kısa zamanda en iyi alternatif çözümü bulmaya yardımcı olur. Simülasyon tekniği, mevcut sistemi modelleyerek, gerçek sistem üzerinde yapılması yüksek maliyetli yada imkansız görülen alternatif işletme politikalarını, kurulan model üzerinde test etme, performans değerlerini kıyaslama olanağı sunar. Yeni bir sistemi hayata geçirmeden önce, bilgisayarda modelleyerek, sistem ilk çalıştırıldığında karşılaşılabilecek bir çok tuzağı önceden belirlemeye yardımcı olur. Devreye alma aşamasında iyi ürün elde etmek için, aylar, belki de yıllar süren çalışmalar simülasyonla, günlere, hatta saatlere indirgenmiş olur .

İmalatta simülasyon kullanımı konusundaki bu çalışmanın gerçekleşmesinde ve akademik hayatıma yaptıkları katkılarından dolayı, başta danışmanım sayın Prof. Dr. Hülya TÜTEK olmak üzere, bölümümüzdeki değerli hocalarıma, uygulama konusunda gösterdikleri kolaylıklardan dolayı Vestel Elektronik A.Ş. çalışanlarına ve aileme içtenlikle teşekkür ederim. Tezimin, simülasyonla modelleme konusunda çalışan araştırmacılara yararlı olmasını dilerim.

GİRİŞ

Günümüzde imalat örgütleri, küresel pazarda, diğer şirketler gibi ürünlerini satmaya çalışarak iyi bir yer edinmeye çabalamaktadırlar. Müşteri isteklerini kalite ve fiyat yönünden karşılamayan ürünler, küresel pazarda diğer ürünlerle rekabet etme şansını kaybederler. Ürünlerin yapısında görülen değişim sonucunda, uluslararası rekabet içinde yerini korumak ve geliştirmek isteyen bir işletme, üretim sisteminin yapısını değiştirmek ve onu müşteri isteklerine hızla uyum sağlayabilen bir özelliğe kavuşturmak zorunda kalmaktadır. Bu nedenle imalat örgütlerinin piyasadaki değişimleri karşılayacak dinamik bir yapıya gereksinimleri vardır. İmalat örgütlerinin bu dinamik yapısı yöneticilerin belirsizlik altında doğru kararlar almasını gerektirir.

Piyasa şartlarındaki değişmelere etkin olarak uyum sağlayabilmek için, yöneticilerin doğru zamanda, doğru karar destek sistemlerini kullanmaları gerekir. Yeni nesil karar destek sistemlerinden biri de simülasyondur. Uzun yıllardan beri simülasyon tekniği kullanılmasına rağmen, gelişmiş bilgisayarlar ve karmaşık yazılımların üretilmesinden sonra, tekniğin gücünün farkına varılmıştır.

Yeni bir sistem dizayn etmek, yada var olan bir sistemde iyileşmeler sağlamak, basitçe sistem elemanlarını tanımlamak ve performans hedeflerini belirlemekten daha çok şey gerektirir. Sistem elemanlarını bağımsız olarak inceleyerek, sistemin nasıl davranacağına karar vermek yanıltıcı sonuçlar doğurabilir. Sistem performansını artırmak için, sisteme bir bütün olarak bakmak gerekir. Özellikle sistemdeki neden-sonuç ilişkileri ortaya çıkarılmalıdır. Neden-sonuç ilişkileri sistemin dinamik yapısını tanımlayarak, sistemin nasıl işlediğini gösterir. Sistemdeki bütün hareketlerin belirlenmesi ve bunlara sebep olan, koşullar, olaylar ve diğer etkenlerin ortaya çıkarılması, neden-sonuç ilişkilerini tanımlama açısından ilk adımdır.

Simülasyon, sistemdeki neden-sonuç ilişkilerini bilgisayara taşıyarak, değişik koşullar altında gerçek sisteme ait davranışların bilgisayar modelinde izlenmesini sağlayan bir modelleme tekniğidir. Simülasyon tekniği, modelde yer alan varlık hareketlerinin istatistiksel özetini de üretir. Simülasyonun çalıştırılmasından sonra, çıkan sonuçlar

sistem performansı için ölçülebilir değerler verir. Bu açıdan bakıldığında simülasyon bir değerlendirme aracıdır. Kullanıcısına değişik dizayn ve işletim stratejilerinin genel sistem performansı üzerindeki etkisini gösterir.

Yeni bir sistemin dizaynında, sisteme ait denemeler, sadece oluşturulacak bir model üzerinde yapılabilir; çünkü gerçek sistem henüz ortada yoktur. Simülasyon, sistemin kuruluş aşamasında gerekli olan hata bulma ve sistemin performans parametrelerini ölçmek için harcanan zamanı önemli ölçüde azaltır. Var olan bir sistemde, iyileştirme çalışmaları yapmak için simülasyon kullanıldığında, mevcut sistem üzerinde hiç bir değişiklik yapmadan, model üzerinde senaryo analizleri yapılabilir. Simülasyon modelindeki varlıklar ve kaynaklar, sisteme yapılan değişikliklerin sonuçlarını değerlendirme açısından , hem daha fazla esneklik sağlar hem de maliyet açısından gerçek yatırımlara göre çok daha ucuzdur.

Bu tezin ilk bölümünde, simülasyon kavramı üzerinde durulmuş, imalatta simülasyon tekniğinin kullanım alanları irdelenmiş ve bilgisayar simülasyon modeli analiz edilmiştir. Simülasyon modelleri sınıflandırılarak, özellikle kesikli simülasyon modelleri ayrıntılı olarak incelenmiştir.

İkinci bölümde, el ile simülasyon, rastgele sayılar kullanılarak örneklenmiş, rastgelelik testleri ve simülasyon analizi için gerekli istatistiksel dağılımlar ayrıntılı olarak incelenmiştir. Daha sonra, simülasyon analizinin aşamalarına geçilmiş, problemi tanımlama, girdilerin toplanması ve istatistiksel analizi, yazılımı seçme, simülasyon modelini geliştirme, doğrulama, modelin geçerliliğini sınamaya ve çıktıların istatistiksel analizi adımları açıklanmıştır.

Üçüncü bölümde, simülasyon paketleri ve simülatörler sınıflandırılarak, yaygın kullanımı olan diller incelenmiştir. İmalat sistemlerinin simülasyonunda yeni yaklaşımlar ele alınmış, paralel ve dağıtılmış sistemlerin simülasyonu, yapay zeka ile modelleme ayrıntılı olarak irdelenmiş ve nesneye yönelik programlama açıklanmıştır. Vestel Elektronik A.Ş.'de bir montaj bandının simülasyonla modellemesini kapsayan uygulama bölümünde, firma hakkında bilgi verilmesinin ardından, 24 saat sürekli

alıřan, retim bandı Őekillerle tanıtılmıř, sistemi temsil eden simlasyon modeli kurulmuř, toplanan veriler simlasyon modeline uygulanmıř, sistemin yoęunluęundan dolayı, gerek sisteme uygulanamayan zaman ve metod etd nerileri, simlasyon modeli zerinde denenmiř ve sonuta sistemde %10 retim artıřı saęlanmıřtır.



BİRİNCİ BÖLÜM

SİMÜLASYON KAVRAMI

1.1 SİMÜLASYONA GİRİŞ

Dünya da yaşanan güçlü küresel rekabet, tüketim talebinin artması ve gelişen teknolojiler, şirketleri işlerini yaparken bir kez daha düşünmeye zorlamıştır. Günümüzde yüksek kaliteli, düşük fiyatlı, müşterilerin zevk ve tercihlerine yönelik ve beklemeden elde edilebilen ürünlerin talebi gittikçe artmaktadır. Talep artışının sürekliliği, kalite, fiyat, zamanında teslimat, müşteri zevk ve tercihlerine uyum gibi kriterlere bağlı olduğundan, artık şirketler mal ve hizmet üretimlerinde bu kriterlere göre davranmanın önemini kavramışlardır. Böylece, mal ve hizmet üretiminin tasarım ve yönetiminde aşağıdaki faktörlerden ileri gelen bazı değişimler yaşanmaktadır.

- Yeni teknolojilerin çoğalması ve gereksinimlerin hızlı değişimi sistemlerin yaşam eğrilerini kısaltmıştır. Piyasaya yeni ürünlerin girmesi artık geçmişte yaşandığından çok daha erken olmakta ve bu durum da büyük yatırımlarla ürünlerini tasarlayıp, üreten ve piyasada yer edinmeye çalışan şirketleri zor durumda bırakmaktadır.
- Teknolojilerin ilerlemesi ve süreçlerin entegrasyonu, sistemlerin giderek daha karmaşık bir hal almasına yol açmaktadır. Bu karmaşık sistemlerin tasarımı, planlanması, üzerinde yapılacak olan modifikasyonlarla paralel olarak gittikçe zorlaşmakta ve mali portresi de ağırlaşmaktadır.
- Sistemlerin, rekabetin kızışması ve müşteri beklentilerinin artmasından dolayı, farklılaşan gereksinimleri karşılayacak performansla sahip

olmaları gerekmektedir. Bu amaçla etkin olmayan, verimliliği düşük süreçlerin sistemlerden ayrılması ya da iyileştirilmesi kaçınılmazdır¹.

Yukarıda söz edilen sistemlerin karmaşıklığı nedeniyle ve süreçlerin iyileştirilmesi konusunda başvurulan analitik yöntemler her zaman uygulanamamakta ya da iyi sonuçlar vermemektedir. Analitik yöntemlerin uygulanmaması durumunda firmaların kullanmaya yöneldikleri analiz araçlarından biri de Simülasyondur.

Simülasyonun tanımına geçmeden önce, sistem kavramı açıklanmalıdır. Sistem, ortak bir amaca ulaşmak için birlikte hareket eden; insan, makine, hammadde vb. gibi birbiriyle ilişkili elemanlar ya da bileşenler topluluğu olarak tanımlanabilir. Sistem doğrudan çalıştırılarak veya üzerinde deneyler yapılabilecek bir prototipi oluşturularak ya da matematiksel ve mantıksal modeller kurularak incelenebilir. Ancak, sistemin doğrudan kendi üzerinde ya da prototipi üzerinde deneyler yapmak, aşırı pahalı, zaman alıcı, değişik açılardan riskli olabileceği gibi çoğu sistem için hiç de pratik olmayabilir. Sistemin modellenmesinin amacı, gerçekte sistemi kurup çalıştırmadan, onu tasarlamak, araştırmak ve değişik kısıtlar altında nasıl davranacağını tahmin etmektir². Sistemin bir ögesi üzerinde değişiklikler yapmak sistemin tümü üzerinde değişimlere neden olacaktır. Sistemler yaklaşımı bu kavramlardan hareketle doğmuştur. Sistem yaklaşımı, probleme ilişkin tüm öğelerin göz önüne alınıp incelenerek problemin anlaşılması ve belirlenmesine ilişkin bir bakış açısı olarak tanımlanabilir. Sistem yaklaşımında herhangi bir öğeyi tek başına ele almak yerine, onu ait olduğu sistem içerisinde incelemek, sistem içindeki rolünü ve diğer öğelerle olan ilişkilerini dikkate almak gerekir. Böylelikle sistemler yaklaşımı; bir düşünme yolu, bir analiz yöntemi ve sonuç olarak bir yönetim tarzıdır³.

¹ Charles HARRELL, Kerim TUMAY, *Simulation Made Easy: A Manager's Guide*, Institute of Industrial Engineers, Atlanta, 2000, s.2-3

² Francis NEELAMKAVIL, *Computer Simulation and Modelling*, John Wiley & Sons, Great Britain, 1987, s.1

³ Oygur YAMAK, *Üretim Yönetimi*, Alfa Basım Yayın, İstanbul 1999, s. 9

Sistemler yaklaşımının karmaşık sistemlerin analizinde ve tasarımında yararlandığı önemli araçlardan biri de Simülasyondur.

Tanım olarak bir sistemin simülasyonu, sistemi temsil edebilecek bir model oluşturma işlemidir. Bu modelin işleyişi ile gerçek sistemin ve ona bağlı alt sistemlerin davranışları rahatlıkla gözlemlenebilir.

Robert Shannon'a göre Simülasyon; "Karmaşık sistemlerin incelenmesi ve tasarımında kullanılan güçlü bir analiz aracıdır"⁴.

Bir başka tanıma göre Simülasyon, mühendis ve planlamacılara bir sistemin tasarım ve işleyişi için rasyonel ve zaman kazandırıcı önerilerde bulunan en güçlü analiz araçlarından biridir.

Simülasyon, kendi kendine problemi çözmez. Ancak problemi açık olarak tanımlayarak, alternatif çözümleri sayısal olarak değerlendirir. Genellikle bir sistemin farklı durumlarını analiz ederek modelin yapısını oluşturabiliriz⁵. Simülasyon, analiz etme, sistemleri tanımlama fırsatı verir, örgütsel sınırlar çerçevesinde eşgüdümlü çalışmayı destekler ve bunun sonucunda, bilgi ve haberleşmenin gelişmesini sağlar⁶.

Simülasyon, aşağıdaki örneklerdeki gibi farklı disiplinlere uygulanabilmektedir:

Ticari hava taşımacılığının lideri "Boeing Corporation's Commercial Aircraft Group" ürettiği uçak modellerinin tasarım ve gelişiminde, uçuş güvenliği ve emniyet faktörleri ile ilgili aerodinamik mühendislik analizlerinde, hava

⁴ Robert SHANNON, Introduction to Simulation, **Proceedings of the 1992 Winter Simulation Conference**, s. 156

⁵ Constantin V. NEGOITA, Dan RALESCU, **Simulation Knowledge-Based Computing and Fuzzy Statistics**, Van Nostrand Reinhold Co., USA, 1987, s.1-2

⁶ Lars HOIST, Gunner BOLMSJO Simulation Integration in Manufacturing System Development: A study of Japanese Industry, **Industrial Management and Data Systems**, 2001, s. 339-356

olaylarının etkileri ve uçak dayanıklılık testleri vb. konularda uçuş simülasyonlarını uzun yıllardır kullanmaktadır⁷.

“Exxon Corporation”, stoklarını etkin olarak kontrol etmek ve tank kullanımını maksimize edebilmek için, bir petrol rafinerisinde benzin dolm sistemini üzerine bir model kurmuş ve simüle etmiştir. Stok tasarrufundan sağlanan değer yaklaşık 1,4 milyon Dolar’dır.

Malzeme taşıma ekipmanları üreticisi olan “HK SYSTEMS” firması, yeni ve farklı sistemlerin tasarımı için simülasyon çalışmaları ile fabrikalarda malzeme ambarları için gerekli depolama alanını minimize ederek, malzemeleri dikey olarak geliştirdiği raf sistemlerine üzerlerindeki barkodları okuyarak otomatik olarak yerleştiren ASRS sistemlerini üretmiştir⁸.

Meksika körfezinde petrol dağıtım maliyetini azaltmak amacıyla, kullanım oranları, kapasite ve talep değerleri, ulaşım maliyetleri ve ilgili diğer kriterleri kapsayan bir simülasyon modeli kurularak, rafinerilerden petrol sevkiyatı en düşük maliyete indirilmiştir⁹.

New York Dünya Ticaret Merkezinde, müşterilerin bekleme zamanlarını azaltabilmek için simülasyon analizi yardımıyla özel asansör sistemi tasarlanmıştır.

ABD’deki bir Devlet Hastanesi, Nükleer Tıp ve Radyolojik görüntüleme merkezindeki hasta yığılmalarını önlemek, personel ve hasta çizelgelemesi yapmak için simülasyon modeline başvurmuştur¹⁰.

Tusaş Motor Sanayi AŞ. de kalite kontrol işlemlerinde ölçüm hataları ve bağlı maliyetleri en aza indirebilmek amacıyla karmaşık optimizasyon teknikleri yerine

⁷ <http://www.sm.com/overview/whitepapers/arenafamily.htm>, Erişim 14.05.2001

⁸ <http://www.hksystems.com/asrs/index.cfm>, Erişim 19.01.2005

⁹ Anderson, Sweney, Williams, **Management Science**, South Western College Pub., 2000, s. 608

¹⁰ www.mech.bee.qut.edu.au/men170/smsch2.html, Erişim 08.05.2001

simülasyon yöntemi uygulanmış ve mevcut ölçü aleti daha verimli kullanılarak hata maliyetleri azaltılmıştır¹¹.

Yukarıdaki örneklerde ve diğer tüm simülasyon çalışmalarında simülasyon aşağıdaki işlevleri yerine getiren deneysel ve uygulamalı bir yöntemdir :

- a) Sistemin davranışlarını gözler ve tanımlar
- b) Gözlenen davranış için geçerli olan teoriler ve hipotezler kurar.
- c) Bu teorileri, gelecekteki davranışı öngörülecek için kullanır.

1.2 SİMÜLASYONUN AMACI VE ÖZELLİKLERİ

Bir sistem önceden tanımlanmış sürece bağlı olarak gelişirken, sistemin davranışı simülasyon modeli ile incelenir. Bu model, sistemin işleyişi ile ilgili, matematiksel, mantıksal veya sembolik bir takım varsayımlar içerir. Bu sayede, sistemdeki potansiyel değişimlerin, sistem performansı üzerindeki etkileri analiz edilir. Ayrıca, sistem henüz kurulmadan planlama aşamasında iken, önerilen sistemi incelemek amacıyla da simülasyon kullanılır. Buradan hareketle simülasyonun kullanım amaçlarını şöyle sıralayabiliriz¹²:

- a) Simülasyon, karmaşık sistemlerin alt sistemlerini incelemek için,
- b) Örgütsel, çevresel ve işlevsel değişikliklerin sistem üzerindeki etkilerini, girdilerin niceliğinde ve niteliğindeki değişimlerin çıktı üzerindeki etkilerini incelemek için,
- c) Simülasyon, eğitime yönelik bir araç olduğundan analitik çözüm yöntemlerini kuvvetlendirmek için,
- d) Simülasyon, yeni işletme politika ve kararlarının yerine getirilmesinin ne gibi sonuçlar doğurabileceğini incelemek için kullanılır.

¹¹ Atilla İŞLİER, Mümtaz ERDEM, Ölçüm Hataları Açısından, Ekonomik Ölçüm Sayısının Belirlenmesi, *Endüstri Mühendisliği Dergisi*, Kasım 1998, s.15

¹² Reuven Y. RUBENSTEIN, Benjamin MELAMET, *Modern Simulation Modeling*, Wiley Interscience Pub., Newyork, 1998, s.6-10

1.2.1 Simülasyonun Özellikleri

Tüm simülasyon modelleri girdi-çıkı modelleridir. Yani, etkileşimli alt sistemlerine bir girdi verildiğinde; bir çıktı verirler. Bu anlamda simülasyon modelleri, istenen bilgiyi ve sonuçları elde etmek için problemi çözmekten çok gerçek durumu örneklemek için koşullar (run). Bu durumda simülasyon modelleri bir teori değil, problem analizi için bir yöntemdir. Simülasyon modelleme performans değerlendirme için etkin bir şekilde kullanılır¹³.

Çeşitli işletme problemlerinin çok karmaşık yapıda olmaları ve bunların çözümünde analitik yöntemlerin kullanılmasının yanı sıra, modelleme yardımıyla bilgisayarlı simülasyon sistemlerinden yararlanılması giderek yaygınlaşmaktadır.

Simülasyon;

- a) Sadece gerçeği temsil etmekle kalmayıp onu taklit eder.
- b) Deney yapmaya yatkındır.
- c) Daha çok tasvire dayalı bir tekniktir,
 - Optimum aranmaz.
 - Özellikler tanımlanır ya da tahmin edilir.
 - Problemin çözümüne yönelik seçeneklerden en iyiyi belirleme olanağını verir.
- d) Belirli kararların toplam etkisini bulmak veya değişik seçenekleri değerlendirebilmek için defalarca tekrar edilebilir.

Simülasyon "eğer-ise" soruları sorarak, deneme yanılma esasına dayalı deneyler yapan bir karar destek aracıdır. Deneme yanılma yönteminin simülasyonda kullanımı olası konfigürasyonların değerlendirilmesine olanak sağlar¹⁴.

¹³ M. NIKOLAIDOV, D. ANAGNOSTOPOULOS, A Distributed System Simulation Modelling Approach, **Simulation Modelling Practise And Theory**, Vol. 11, 2003

¹⁴ Henri PIERREVAL, Jean Lue PARIS, From Simulation Optimization To Simulation Configuration Of Systems, **Simulation Modelling Practise And Theory**, Vol. 11, 2003

Simülasyon aşağıdaki örnek sorulara benzer sorular için yanıtlar arar¹⁵ :

- Yaşam eğrisinin kısalmasının çıktı miktarı üzerindeki etkisi nedir?
- Koruyucu bakım daha az yapılırsa makine duruşlarında önemli artış olur mu?
- Süreç içindeki yarı mamul stokları %10 artırılsa ne olur?
- Parti üretim miktarı azaltılırsa ne olur?
- İşyeri düzenlemesindeki değişiklik işgücü verimliliğini nasıl etkiler?

Simülasyon, matematiksel anlamda optimumu arama işlemi değildir. Sistemle ilgili akla gelebilecek iyileştirme önerilerinin incelenmesi ve bu alanda karar almaya yardımcı olacak bilgilerin toplanmasıdır¹⁶.

1.2.2 Simülasyon Kullanımının Uygun Olduğu Durumlar

Simülasyon ile çözülmek üzere kurulmuş bir model , analitik yolla çözülmek için kurulmuş bir modelden çoğunlukla farklıdır. Analitik yöntem daha dar kapsamlıdır¹⁷. Bir problemi temsil edecek ve modele eklenecek yeni değişkenler modelin çözülebilirliğini güçleştirir, hatta bazen olursuz çözümler de ortaya çıkabilir. Buna karşın simülasyon ile modelleme, daha esnek bir yapıya sahiptir ve modele değişken ekleme nispi olarak daha kolaydır.

Simülasyon, deneme yanılma yöntemiyle problemin formülasyonu ve çözümüne odaklanır. Simülasyon süreci iterasyonlarla gerçekleştirilir ve problemin ilgili olduğu alana yeni bakış açıları sağlar. İterasyona yönelik prosedürde mevcut çalışılan model, sürekli bir şekilde yeniden tanımlanır. Simülasyon ve modelleme bu nedenle simülasyon süreci çerçevesinde ayrılmaz bir bütündür¹⁸.

¹⁵ Stewart ROBINSON, The Application of Computer Simulation in Manufacturing, **Integrated Manufacturing Systems**, Vol 4 1993, s.18

¹⁶ Murat DİNÇMEN, Benzetim Tekniği ve Sanayideki Uygulamaları, **Sanayide Bilgisayar Kullanımı ve Otomasyon**, 1990, s.169

¹⁷ John BUZACOTT, J. George SHANTIKUMAR, **Stochastic Models of Manufacturing Systems**, Prentice Hall, New Jersey, 1993, s.14-15

¹⁸ Georges HAPCHI, Claire BERCHET, A Model For Manufacturing Systems Simulation With A Control Dimension , **Simulation Modelling Practice And Theory**, Vol. 11, 2003, s. 21-44

Simülasyon, aşağıda belirtilen özelliklerinden dolayı belirli koşullar altında uygulanmalıdır:

- a) Problemin tam bir matematik formülasyonu'nun olmadığı veya matematiksel modelin çözümlenmesini sağlayacak analitik yöntemin henüz geliştirilmemiş olduğu durumlar.
- b) Analitik modellerin kullanılabilir olduğu ancak matematiksel çözümlene yollarının çok karmaşık olması nedeniyle simülasyonun pratik yarar sağlayacağı durumlar.
- c) Analitik çözümlerin bulunduğu, ancak bu çözümlerin, bu işin sorumlusu olan kişinin kantitatif bilgi ve becerisini aştığı durumlar. Bu durumlarda simülasyonun tasarlanması, test edilmesi ve uygulanmasının maliyeti ile dış yardım elde etmenin maliyeti kıyaslanmalıdır.
- d) Deneylerin yürütülmesinde ve olayların mevcut çevre içinde gözlemlenmesinde güçlük çekilen durumlar.

1.3 SİMÜLASYONUN AVANTAJ VE DEZAVANTAJLARI

Simülasyon yaklaşımının uygun koşullarda kullanılması durumunda sağladığı yararlar aşağıda tartışılmaktadır:

1.3.1 Simülasyonun Avantajları

- Simülasyon, analizi yapan kişiyi genel ve geniş düşünmeye zorlar.
- Simülasyon ile sistemlerin gerçek zamanı, oluşturulan simülasyon modeli üzerinde daraltılmış veya genişletilmiş süre içinde incelenebilir.
- Matematik modellerle analitik çözümler bulunduktan sonra, bu çözümlerin doğruluğu simülasyon ile araştırılabilir¹⁹.

¹⁹ Wieslaw SWITEK, Tadeusz MAJEWSKI, Dynamic Modeling and Optimization For Technology Management, *Computers & Industrial Engineering*, Vol.33, 1997, s.11-14

- Yöneticilerin, simülasyon analizi ile işletme problemi hakkında elde ettikleri bilgi, simüle edilen sistemin yapı ve politikalarında diğer zamanlarda elde edilemeyecek biçimde ilerlemeye yol açar. Diğer bir deyişle, model kurmaya alt sistem yöneticilerinin katılmasının sağlayacağı fayda, bir bakıma simülasyonun sağlayacağı faydadan daha fazladır.
- Simülasyon, üretim sisteminin başarısını etkileyen çevre koşullarını göz önünde bulundurur. Sistem başarısının bütün olarak değerlendirilmesi ile başka herhangi bir yöntemle gösterilemeyen değişkenler arası etkileşimlerden hangilerinin sistem başarısını tayin etmede etkin olduğu belirlenmiş olur.
- Simülasyon, analitik yöntemlerden daha kolay uygulanabilir. Özellikle karmaşık sistemlerin tasarım ve işlevsel performansının analizinde matematiksel ve analitik yollarla açıklanamayacak ilişkileri tanımlamak için simülasyon analizi rahatlıkla kullanılabilir²⁰.
- Simülasyon analizi, önerilen politika ve kararların çoğunluğunu test eden ve değerleyen bir deneme çalışması olarak yöneticilere ışık tutar. Simülasyon, ister uzun ister kısa dönem için dinamik üretim sistemlerini analiz etme ve değerlendirme olanağı verir.
- Farklı durumların incelenmesi için istenildiği kadar kullanılabilir.
- Simülasyon, bir işletmede ele alınan sistemin değişkenleri arasındaki etkileşimleri analiz etme ve değerlendirme olanağı verir.
- Simülasyon, değişik koşullar altında sistemin nasıl olacağı hakkında çok az ya da hiçbir veri bulunmadığında yeni durumlar üzerinde deney yapma amacıyla kullanılabilir. Bir simülasyon modeli, temel olarak "eğer-ise" ("what-if") analizlerinin yapılmasını sağlayan bir araç olarak ele alınmalıdır. Kullanıcısına değişik dizayn ve işletim stratejilerinin genel sistem performansı üzerindeki etkisini gösterir²¹.

²⁰ A.ANGLANI, M GRIECO, M. PACELLA, T. TOLIO, Object-Oriented Modeling and Simulation of Flexible Manufacturing Systems: a rule-based procedure , Proceedings of 2002 WSC, s. 209

²¹ <http://www.uytes.com.tr/simulasyon/index.html>, Erişim 12.11.2003

1.3.2 Simülasyonun Dezavantajları

- Bir simülasyon modeli, gerçek durumu yansıtmadığı halde, gerçek durumun modeli gibi algılanabilir.
- Her zaman kolaylıkla bulunmayacak düzeyde yaratıcılık gerektirir.
- Araştırmacılar simülasyon tekniğini öğrendikten sonra, onu analitik yöntemlerin daha uygun olduğu durumlarda da kullanma eğilimindedirler.
- Kurulan bir simülasyon modelinin, bilgisayarda defalarca çalıştırılması gerektiğinde yüksek tabanlı ve pahalı bilgisayar sistemlerine gereksinim vardır.
- Simülasyon modelinin, bilgisayarda kurulması ve geçerli olduğunu ispatlamanın maliyeti yüksektir. Genel olarak her sistem için ayrı bir program yazmayı gerektirir ki bu zaman alıcıdır. Simülasyon için yazılmış paket programlar ve özel amaçlı diller bu sakıncayı bir dereceye kadar ortadan kaldırır.

1.4 İMALATTA SİMÜLASYONUN KULLANIMI

Artan rekabetten dolayı işletmeler pahalı otomasyon yatırımlarına zorlanmakta, bundan dolayı üretim politikalarını yeniden değerlendirmektedirler. Günümüzde endüstriyel problemlerin doğasındaki stokastik yapı ve karmaşıklık pek çok analitik çözümü saf dışı bırakmaktadır. Oysa, problemlerin yapısı değişen teknolojiyle birlikte karmaşılaşmakta ve entegre sistemlerin sayısı hızla artmaktadır²². İmalat işletmeleri yeni tasarım ve sistemlerini uygulamaya koymadan önce dinamik bir analiz aracı olarak simülasyonu tercih etmektedirler. Simülasyon çalışmalarında kullanılan animasyonlar sistem davranışının anlaşılmasını kolaylaştırmaktadır. Yazılım programlarının

²² Erkan BAYRAKTAR, Karmaşık Endüstriyel Sistemlerin Tasarım ve Analizi için Simülasyon, **Makine Magazin**, Ocak 1997, s.55

gelişmesi, bilgisayarlı simülasyon sonuçlarının alınmasında zaman ve maliyeti azaltmıştır²³.

Bir imalat sisteminin tasarımından uygulamasına kadar simülasyon, aşağıdaki temel konularda yöneticiye yardımcı olur:

- Fabrika yerleşimi, tasarım, ekipman kararları ve alternatif işletme politikalarının belirlenmesinde²⁴,
- Üretimin Planlanmasında çizelgeleme aracı olarak aşağıdaki hedefleri gerçekleştirmek amacıyla:
 1. Üretim Olanaklarının en etkin şekilde kullanımı ve kapasite planlamasında,
 2. En az hatalı talep tahminleri yapmada ve müşteri taleplerinin olabildiğince hızlı yanıtlanmasında,
 3. Hammadde, parça ve malzemelerin en uygun zaman ve maliyetle temininde,
 4. Optimum işgücü ve iş dağıtım planlamasında rotalama aracı olarak,
- Gerçek zamanlı ve on-line kontrol sistemi olarak bir veri tabanından bilgileri okuyarak çizelgeleme ve öngörülemede bulunma amacıyla kullanımı. Örneğin, Esnek İmalat sistemlerinin kullanımını artıracak etkin çizelgeleme ve kontrol politikalarının geliştirilmesinde bir çevrim içi (on-line) karar destek aracı olarak, dinamik ve stokastik sistemlerin zaman içerisindeki davranışlarının etkisi, simülasyon analizi ile gerçekçi bir şekilde incelenebilir²⁵.

Simülasyon imalat sistemlerinde bir tasarım aracı olarak kullanılacaksa, aşağıdaki sorulara cevap arar :

²³ Hamid R. PARSEI, William G. SULLIVAN, **Concurrent Engineering**, Chapman & Hall, Londra, 1993, s.120

²⁴ J.A. HARDING, K. POPPLEWELL, Simulation: An Application of Factory Design Process Methodology, **Journal of the Operational Research Society** , April 2000, s.440

²⁵ Semra TUNALI, Benzetim Tekniğinin bir Çevrim İçi Karar Destek Aracı Olarak Esnek İmalat Sistemlerinde Kullanılması, **Endüstri Mühendisliği Dergisi**, Ağustos 1996, s.10

- Bu tasarımın üretim kapasitesi ne olacaktır? Bu düzey, üretim hedeflerini karşılar mı?
- Üretim hattında darboğaz nerede olabilir? Gidermek için ne gibi düzenlemeler yapılabilir?
- Farklı tasarım alternatifleri arasında hangisi en iyisidir? Tezgah tipi ve sayısı, işgören sayısı, malzeme aktarma sistemi, süreç içi stok seviyeleri sistem performansını nasıl etkiler?
- Sistemin güvenilirliği nedir? Arızalar ve koruyucu bakım sistemini nasıl etkiler?

İmalat sistemlerinde simülasyon kullanılması, yerel değişkenlerin; yani alt sistemlerdeki değişkenlerin, tüm sistem davranışını ne şekilde değiştireceğini belirlemeye olanak sağlar. Bunun yanı sıra, imalat sistemlerinin analizinde simülasyon kullanımının başlıca yararları şöyle sıralanır:

- Çıktı miktarında artış,
- Ürün ve malzeme stoklarında azalma,
- Makinelerin ve insanların etkin kullanımı(Beklemelerin azalması),
- Sermaye gereksiniminde azalma,
- İşleme(operation) maliyetinde düşme,
- Zamanında teslimat,
- Simülasyon çalışması için, toplanan verilerin sistemin daha iyi anlaşılmasını sağlamasıyla çalışmaya konu olan alanın dışındaki konularda da iyileştirmelerin sağlanması,

Simülasyonun sıklıkla uygulandığı imalat alanları da şöyle sıralanabilir:

- a) Bekleme Hattı Sorunlarında Simülasyon:** İmalat sırasında iş istasyonlarında işlem görmek üzere bekleyen malzeme ya da yarı mamuller, işlem yapmak üzere bekleyen işgörenler, makineler vb. gibi öğeler vardır. Bu tür sorunlarda amaç, insan ve makinelerin boşa geçen zamanlarının

toplam maliyetini minimize etmektir. Bir işgünü boyunca üretimin sürekliliği ve nicel büyüklüğü göz önüne alındığında, elle yapılan çizelgelemelere kıyasla, bilgisayarlı simülasyonun çok daha hızlı sonuç verdiği, üstelik girdilerdeki değişimleri anında sonuçlara yansıttığı açıktır²⁶.

b) Montaj Hattı Sorunlarında Simülasyon: Genellikle yürüyen bant ya da konveyör ile malzeme ve yarı mamulün hat boyunca hareket ettiği montaj hatlarında, darboğazlardan kaynaklanan dengeleme sorunlarının çözümünde yine simülasyon analizi kullanılabilir. En iyi çözüm, değişik şartlar altında işlerin iş merkezlerine boşta geçen zamanı en aza indirecek şekilde yüklenmesidir. Montaj hattı dengeleme sorunları, genelde bilgisayar kullanılarak çözülür. Bir montaj hattında, malzeme deposundan malzemenin ilk işlem için üretim hattına girişi ile başlayan, giriş sıklığı (gelişler arası zaman) dağılımının belirlendiği ve özel şekilde gösterildiği bilgisayarlı modellemelerde SLAM, SIMAN, GPSS gibi simülasyon dilleri sık sık kullanılmaktadır²⁷.

c) Tamir ve Bakım Sistemlerinin Simülasyonu: Makinelerin bozulması, özellikle çok pahalı yatırımlarla alınan ve tesisteki iş yükünün büyük kısmını üstlenen otomasyon sistemlerinin durması çok ağır maliyetlere katlanmaya, hatta, işletmenin kapanmasına kadar gidecek olumsuzluklara sebep olmaktadır. Üretim sistemlerinin tümü ile aksamadan çalışmasını sürdürmek, bakımı belirli bir plan çerçevesinde yürütmek ve beklenmedik arızaları minimum düzeyde tutmak kısaca, fabrikanın güvenilirlik derecesini artırmak şeklinde tanımlanabilir. Güvenilirliği artırmanın yolu da işletmenin yapısı ve kısıt faktörlerini dikkate alan tamir bakım politikalarını belirlemekten geçer²⁸. Bu nedenle koruyucu ve tamir bakım olarak iki tür bakımdan söz edilebilir. Koruyucu bakım çok iyi bir planlamayla yapılırsa bazı arızalar oluşmadan ilgili parçaların değiştirilmesi suretiyle olası arızalar kısmen önlenir. Ancak

²⁶ C. K. LIN, K.F. LAI, S.L. HUNG, Develop of a Workforce Management System For a Customer Hotline Service, **Computers & Operations Research**, 2000 vol. 27, s.987

²⁷ S.E. MOUSSA, C. MOGRABI, M.S. EID, Simulation The First Operation in an Assembly Line, **Computers & Industrial Engineering**, Vol 37 1999, s.211

²⁸ Bülent KOBU, **Üretim Yönetimi**, Onbirinci Basım, İstanbul 2003, s.315

koruyucu bakım ve tamir bakım maliyetleri arasındaki ilişkiler ters yönlüdür. Bundan dolayı tamir bakım sistemlerinin amacı toplam bakım maliyetini azaltmaktır.

Yapılacak olan onarımın kapsamının belirlenmesinde aşağıdaki alternatiflerden hangisinin seçileceğine karar verme, tipik bir tamir bakım simülasyon analizini gerektirir. Bu seçimde;

- a. Tamir bakım ekibini geniş tutmak,
- b. Koruyucu bakıma ağırlık vermek,
- c. Yedek üretim kapasitesi bulundurmak,
- d. Makinelerin güvenilirlik derecesini artırmak,
- e. İş istasyonları arasında yarı mamul stoğu bulundurmak,
- f. Koruyucu bakım ile tamir bakım arasında optimal dengenin kurulması,

Alternatiflerinden birinin ya da bir kombinasyonun seçiminde, simülasyon analizi en uygun yöntemlerden biri olarak karşımıza çıkar.

d) Üretim Planlama ve Çizelgelemede Simülasyon: Ürün çeşitliliğine göre türü ve miktarı değişen müşteri siparişlerini, zamanında teslim esasına göre yapılan çizelgelenmeler çok daha hızlı ve kapsamlı bir biçimde simülasyon ile analiz edilebilir. Tüketici talebinin geleceğini de öngörülecek şekilde, talep tahminleri de modele katılabilir.

e) Envanter Sorunlarında Simülasyon: Yine üretim planlarına bağlı olarak malzeme tedarik zamanları göz önüne alınıp, çok çeşitli envanter politikalarının sonuçları simülasyon analizi ile kıyaslanabilir.

1.5 SİMÜLASYON MODELİNİN YAPISI

Model, en genel anlamda çok basit olarak aşağıdaki gibi gösterilebilir²⁹:

$E = f(X_i, Y_i)$ Bu ifade;

E: Sistem performansının etkisi

X_i : Denetleyebildiğimiz değişken ve parametreler

Y_i : Denetleyemediğimiz değişken ve parametreler

f: X_i ve Y_i arasındaki ilişki

Simülasyon açısından, hemen hemen tüm modellerin aşağıdaki elemanlardan oluştuğunu söylemek mümkündür:

1. Varlıklar(Entropy): Model içerisinde tanımlanan nesnelere ve bir araya geldiklerinde sistemi oluştururlar. Bağımsız olarak belirlenirler ve bunların ortak performansı sistemin çıktısını oluşturur³⁰.

2. Değişkenler(Variables): Sistemin özellikleridir. Değişik koşullarda ve değişik sistem durumlarında farklı değerler alırlar. Değişkenler dört karşıt kategori içinde sınıflandırılabilir :

- *Bağımsız ve bağımlı*
- *Denetlenebilir ve denetlenemez*
- *İçsel ve dışsal*
- *Girdi ve çıktı*

Örnek olarak, bir üretim sistemi düşünüldüğünde, belirli faktörlere bağlı olarak farklı değerler alan değişkenler bağımlı, diğerleri bağımsızdır. Üretimde uygulanan kuyruk disiplini denetlenebilir bir değişkendir. Sistem üzerinde alternatif değişiklikler yaparak değişkenin değeri değiştirilebilir. Kuyruğa geliş

²⁹ Thomas J. GOGG, Jack R. MOTT, **Improve Quality and Productivity With Simulation**, JMI Consulting Group, 1992, s.95

³⁰ John S. CARSON, Introduction To Simulation, **Proceedings of the 2003 Winter Simulation Conference**, s.8

oranı ise gözlemler sonucu bulunup modelin kurulmasında sabit bir değişken olarak modelde yer alan denetlenemez bir değişken olur. Simülasyon modelinin kurulup, çalıştırılması sonucu, sistem tarafından üretilen kuyruktaki ortalama eleman sayısı ya da kullanım oranı gibi değerler içsel girdileri oluşturur. Bir envanter sistemi ele alındığında, yeniden sipariş noktası denetlenebilir içsel çıktılardır. Dışsal değişken ise sistemin çalışması sonucu oluşan çıktı değerleridir. Dışsal değişkenlerle beraber sistem kaynakları irdelenerek verimlilik artışı sağlanmaya çalışılır³¹.

Tablo 1 söz konusu değişkenlerin ilişkilerini özetlemektedir³².

Tablo 1 Sistem Değişkenlerinin Birbirleri ile olan İlişkisi

Sistem Değişkenleri	Bağımsız	Bağımlı	Denetlenebilir	Denetlenemez	İçsel	Dışsal	Girdi	Çıktı
Bağımsız			X			X	X	
Bağımlı				X	X			X
Denetlenebilir	X				X	X	X	
Denetlenemez		X			X	X	X	X
İçsel		X	X	X				X
Dışsal	X		X	X			X	
Girdi	X		X	X		X		
Çıktı		X		X	X			

3. Parametreler: Sistem analistinin keyfi değerler verdiği değerlerdir. Bunlar analiz boyunca değiştirilemezler. Örneğin $Y = 3x$ gibi bir denklemde 3 parametre, Y ve x değişkenlerdir.

4. İlişkiler: Sistemin varlıkları arasındaki bağıntılardır. Bunlar sistem durumundaki değişimleri denetler. İlişkiler dokuz grupta incelenebilir :

- **Yapısal İlişki:** Varlıkları ve bunların özelliklerini birbirine bağlayan ilişkilerdir.

³¹ Andre THOMAS, Patrick CHARPENTIER, Reducing Simulation Models For Scheduling Manufacturing Facilities, **European Journal Of Operational Research**, Vol. 161, 2005

³² Bayram Ali SU, **Simulation Languages**, D.E.Ü. Endüstri Müh Bl., Yayınlanmamış Ders Notları, 1996

- **İşlevsel İlişki:** Varlık ile varlıkların davranışını belirleyen ilişkilerdir.
- **Sayısal İlişki:** Birbirine yalnızca zaman açısından bağımlı olan sistem olayları arasındaki ilişkilerdir.
- **Mekansal İlişki:** Fiziksel öğelerin, mekan içinde, kendi aralarında belli bir ilişki içinde buldukları statik sistemler buna iyi bir örnektir.
- **Zamansal İlişki:** Modeli oluşturan varlıklar arasında zamana dayalı etkileşimleri kapsar.
- **Neden - Sonuç İlişkisi:** Bu ilişki en büyük doğa yasalarından biridir. Sonlu bir zaman süresi içinde gerçekleşen, tersinmez süreçlerde uygulanan bir yasadır.
- **Enerjinin Korunumu İlişkisi:** Maddenin bir biçimden diğerine geçişte görülen bir ilişki türüdür.
- **Mantıksal İlişki:** Özellikle soyut sistemlerde görülen bir ilişki türüdür. Yani, düşünceler ve kavramlara ilişkin sistemlerde rastlanabilir.
- **Matematiksel İlişki:** Mantıksal ilişkinin özel bir biçimidir.

5. Varsayımlar: Modeli gerçek durumdan soyutlayan kabullerdir. Varsayımlar üzerinde yapılan değişiklikler ile modelin soyutlama derecesi de değişir.

6. Kısıtlar: Kısıtlar, değişkenlerin değerleri veya kaynakların nasıl tahsis edileceği üzerindeki sınırlandırmalardır. Bu kısıtlar, tasarımcı tarafından koyulabileceği gibi, sistemin doğasında da olabilir.

7. Ölçütler: Ölçüt fonksiyonu, sistemin amaçlarını ve bunların nasıl değerlendirileceğini kapsar. Ölçüt değerlendirme standardı olarak da tanımlanabilir. Buna göre ölçüt fonksiyonu iki açıdan büyük önem kazanmaktadır. Birincisi, modelin tasarımı ve işletilmesi üzerinde büyük etkisi vardır. İkincisi, ölçütün yanlış tanımlanmasının yanlış sonuçlar vereceği gerçeğidir.

Tek elemanla hizmet veren bir benzin istasyonunun simülasyonla modellenmesini örnek olarak ele aldığımızda:

Varlıklar: Hizmet almak üzere bekleyen arabalar, kuyruğu görerek akaryakıt almaktan vazgeçen arabalar, herhangi bir anda hizmet almakta olan ve hizmet alıp istasyondan ayrılmakta olan arabalar, pompa görevlisi, farklı akaryakıt pompaları varlıkları oluşturur.

Değişkenler: Ana değişkenler arabaların gelişleri ve hizmet alımlarıdır. Gelen arabaların ortalama kuyrukta bekleme süresi, kuyrukta herhangi bir anda araba sayısı, ortalama hizmet süresi gibi içsel değişkenler ise, sistem tarafından oluşturulmaktadır.

Parametreler: Belirli miktarda ve katlarında akaryakıt alanlar için promosyon olarak verilen mal veya hizmetler parametreleri oluşturur.

İlişkiler: Akaryakıt alan arabaların mekansal, zamansal, mantıksal, sayısal vb. davranışlarının analiz edilerek modele yansıtılmasıdır. Bir pompanın bir arabaya akaryakıt verirken, pompa görevlisinin diğer bir arabaya hizmet vermeye başlaması, bir arabanın akaryakıt aldıktan sonra hava/yağ filtrelerini değiştirmek istemesi, bir başka arabanın yıkama talep etmesi gibi olası durumlar, uygun değişkenlerin eklenmesiyle modele katılmalıdır.

Varsayımlar: İstasyonun 24 saat hizmet vermesi

Kısıtlar: Pompa sayısının mekansal ilişki gereği 4'ü aşmaması, mantıksal ilişki gereği en fazla 2 görevlinin çalıştırılması vb. durumlar simülasyon modelinin yapısını oluşturur.

1.6 SİMÜLASYON MODELİNİN NİTELİKLERİ

Simülasyon, sistemin işletimi ile ilgilidir. Simülasyon, gerçek dünyanın sorunları ile ilgili olduğuna göre, elde edilen sonucun gerçek anlamda, gerçek durumu

yansıttığından emin olunmalıdır³³. Bir simülasyon modeli, parametrelerin ve değişkenlerin uç değerlerinde irdelenmelidir. Eğer anlamsız sonuçlar çıkar ise, modelden kuşku duymak ve gözden geçirmek zorunludur.

Son olarak, türetilen bilgilerin kullanıcılarını dikkate almak gerekir. Karar verici tarafından kullanılmayan veya kullanılabilir olmayan bir model hiçbir açıdan savunulabilir olamaz.

Bu açıklamalar çerçevesinde iyi bir simülasyon modelinden neler beklendiği şu şekilde sıralanabilir³⁴:

- Kullanıcı tarafından kolaylıkla anlaşılmalıdır.
- Amaç ve hedef yönlü olmalıdır.
- Anlamsız sonuçlar vermeyecek şekilde tasarlanmalıdır.
- Kullanıcı tarafından denetimi ve işletilmesi kolay olmalıdır.
- Sistemi tam olarak açıklayabilmelidir.
- Model değişikliği ve güncelleştirilmesi için kolaylıkla uyarlanabilir olmalıdır.
- Evrimsel olmalıdır; yani basit bir şekilde başlayıp giderek karmaşıklaşmalıdır.

1.7 BİLGİSAYARLI SİMÜLASYONUN TEMELLERİ

Bilgisayar simülasyonu, bir sistemdeki süreçlere ilişkin verileri kullanarak, elle yapılması son derece zor olan analizleri, bilgisayar yardımıyla yapan deneysel bir yöntemdir. Karmaşık bir sistemin analizinde kullanılacak etkin, ekonomik ve bazen de mümkün olan tek yöntemdir. Bilgisayar simülasyonu, gerçek sisteme uygulanacak deneylerle kıyaslandığında, düşük maliyet, zaman tasarrufu, artan esneklik, ve azalan riski beraberinde getirir. Bundan dolayı bilgisayar simülasyonunun, imalat sistemlerinde kullanılması hem mühendislerin, hem de akademik araştırmacıların ilgisini çekmiştir³⁵.

³³ B. Scholz REITER, U.U. STAHLMAN, A. NETHE, **Process Modeling**, Springer, Germany, 1999, s. 351-354

³⁴ Men170: SMS, <http://mech.bee.qut.edu.au/men170smsch5.html>, erişim 19.09.2000

³⁵ B. WU, **Manufacturing Systems Design**, Chapman & Hall, Londra, 1992, s.193

Gelişmiş imalat sistemlerinin tasarımı ve işletmeye adaptasyonu, hayli karmaşık tasarım sorunları yaratır. İşletmenin, üretim hacmi ve kalitesini doğrudan artıran bu tür sistemlerin başarıya ulaşması için, yöneticiler büyük mali yatırımlar ve çabalar ortaya koymaktadırlar. Ancak böyle sistemlerin, gerçekten işletmede uygulamaya alınmasından önce, yeni sistemin çalışma performansının, işletmedeki sistem performansına etkisinin incelenmesi ve yeni sistemin işletmede yürürlüğe ne şekilde alınacağına bazı denemelere tabi tutulması gerekecektir. Bilgisayar simülasyonu, bu tür durumlarda geniş tasarım ve kullanım kolaylıkları ile yöneticilerin karar vermesini kolaylaştıran bir araç olarak karşımıza çıkar³⁶.

1.7.1 Bilgisayar Simülasyonunun Karakteristikleri

a) Esneklik

Bilgisayar simülasyonunun en önemli avantajlarından biri esnek olmasıdır. Model bir kez kurulduktan sonra, yeni özellikleri modele tanıtmak oldukça kolay ve maliyetsizdir. Aynı model üzerinde, farklı alternatiflerin sonuçlarını inceleyebilmek amacıyla, sık sık girdi verileri değiştirilebilir.

b) Geçici Davranışların İncelenmesi

Çoğu sistem oldukça karmaşık bir yapıda olup, ancak belirli basitleştirme varsayımları altında analitik modeller yaratılır. Aslında sürekli çalışan sonsuz sistemler genellikle bir geçiş evresinden geçerek sabit durumlarını alır. Durağan sisteme geçmiş bir sistemde, kuyruklar belli uzunluklara gelirler ve parçaların sistemde kalış süreleri dengededir. Geçiş evresinde kuyruklar henüz normal uzunluklarını almamışlardır. Bu nedenle sonsuz sistemlerde ya başlangıç koşulları belirlenerek ya da geçiş devresinin bitişi belirlenip, bu zamana kadar olan çıktılar atılarak simülasyon modeli uygulamaya konur.

³⁶ Glenn R. DRAKE, Jeffrey S. SMITH, Simulation As a Planning and Scheduling Tool for Flexible Manufacturing Ssystems, *Proceedings of the 1995 Winter Simulation Conference*, s. 805-810

c) Animasyon

Bilgisayar simülasyonunun bir diğer avantajı da görselliktir. Üretim sistemlerinin, animasyonlar (hareketli resimler) ve grafiklerle ifade edilmesi simülasyona canlılık katar. Animasyon bilgisayar simülasyonunda daha çok modelin doğruluğunun ve geçerliliğinin araştırılması aşamasında kullanılmaktadır.

İyi bir animasyon aşağıdaki özelliklere sahip olmalıdır³⁷:

- **İnteraktiflik:** Kullanıcı model içinden seçtiği birimi detaylı olarak inceleyebilmelidir.
- **Gerçekçilik:** Animasyonun gerçekçiliği modelin daha canlı olmasını sağlar.
- **Performans:** Animasyonun yeterince hızlı hareket etmesi ile kullanılabilirliği artırılmalıdır.
- **Esneklik:** Kullanıcı animasyonda nelerin gösterileceğine karar verebilmelidir.
- **Kolay kullanım:** Animasyona, bilgisayar programının sunduğu ara yüzler sayesinde kolay bir şekilde grafik eklenebilmelidir.

1.7.2 Simülasyon Modellerinin Sınıflandırılması

Simülasyon modelleri çok değişik biçimde sınıflandırılmaktadır. Bu gruplardan başlıcaları:

- Statik/Dinamik
- Stokastik/Deterministik
- Kesikli/Sürekli simülasyon modelleridir³⁸.

³⁷ Michael PIDD, **Computer Simulation in Management Science**, John Willey & Sons, Newyork 1998, s.114

³⁸ David, KELTON, Randall SADOWSKI, Deborah SADOWSKI, **Simulation With Arena**, McGraw Hill , Newyork 2002, s..9

1.7.2.1 Statik ve Dinamik Simülasyon Modelleri

Statik simülasyon modelleri, zamanın bir fonksiyonu olarak herhangi bir etkileşim niteliği göstermez. Diğer bir deyişle, Statik simülasyonda model değişkenleri arasında ilişkiler belirli bir süre sabit sayılır. İşyeri düzenleme ve montaj hattı düzenleme statik simülasyona örneklerdir.

Dinamik simülasyonda, model değişkenleri arasında zamana bağlı değişimler söz konusudur. Envanter, Kuyruk, Sipariş, Çizelgeleme gibi işletme sorunlarına yönelik modeller dinamik modele örnek gösterilebilir³⁹.

1.7.2.2 Stokastik (Olasılıklı) ve Deterministik (Belirli) Simülasyon Modelleri

Deterministik simülasyon modelleri, sistemde herhangi bir tesadüfi öge bulunmaması durumunda söz konusudur. Eğer girdiler hep aynı değerleri alıyorsa, bunların sonuçları da doğal olarak belirli olacaktır. Gerçek fiziksel sistemde tesadüfilik bulunsa bile, sorunun yapısını ve çözümünü kolaylaştırmak için bu göz önüne alınmaz. Montaj hattı dengeleme, iş sıralama gibi sorunlar deterministik simülasyon modelleridir.

Stokastik (Olasılıklı) simülasyon modellerinde, sistemin sonuçlarını tesadüfi nedenler belirler. Sürecin ortalama değeri ve değişkenlik düzeyi olasılık dağılımı ile belirlenir. Yani, olasılık dağılımı süreci temsil eden modeli oluşturur. Stokastik simülasyonda gerçek sistemin tamamen aynısı ya da onun benzeri olan veri türetme yöntemi kullanılır⁴⁰. Tesadüfi olarak ve tarafsızca örnekleme yapılır. Simülasyon sonucunda gerçek sistemin sonuçlarına yakın değerler elde edilemiyorsa model gözden geçirilmelidir. Bilgisayarla yapılabileceği gibi, eğer elle çözüm yapılmak istenirse Monte Carlo simülasyonu, basit ama etkin sonuçlarıyla öne çıkar.

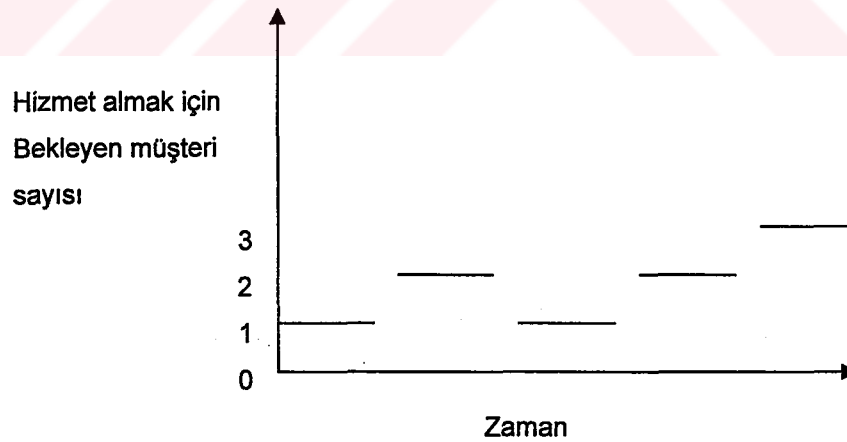
³⁹ Path to success, <http://www.sm.com/overview/whitepapers/pathsuccess.htm>, Erişim 19.09.2000

⁴⁰ Ronald L. RARDIN, **Optimization in Operations Research**, Prentice Hall, 1998, s.11-15

1.7.2.3 Kesikli (Discrete) Simülasyon Modelleri

Kesikli simülasyon, durum değişkenlerinin zaman içerisinde farklı noktalarda değişimini içeren sistemlerin modellenmesidir. Zamanındaki bu noktalar, olayların meydana geldiği anları ifade eder⁴¹. Bir üretim sisteminde olaylara, bir makinenin çalışır durumda iken görevini yapması, yüklü durumdan atıl duruma geçmesi, ya da çalışırken arızalı duruma geçmesi, arızalı iken tamirinin bitişi ile yeniden meşgul statüsünü alması gibi durumlar örnek gösterilebilir⁴². Kesikli simülasyon sistemleri imalat, ulaştırma, iletişim, finans, tıp gibi çeşitli alanlara uygulanmaktadır. Kesikli simülasyon verimliliği en iyilemek için kullanılan bir araçtır⁴³.

Kesikli simülasyona, bir bankanın gişelerinde hizmet almak üzere bekleyen müşterilerin oluşturduğu kuyruk sorunu örnek olarak verilebilir. Bu sistemde, bankadaki müşterilerin sayısı, durum değişkeni olarak ifade edilir ve müşteriye hizmetin tamamlandığı anda, ya da yeni bir müşterinin bankaya gelmesi durumunda, durum değişkeni değerinde değişiklik olur. Şekil 1 zaman içerisinde kesikli noktalarda müşteri sayısındaki değişimi göstermektedir.



Şekil 1 Kesikli Simülasyonda Durum Değişkeni

⁴¹ Russel R. BOSTON, Designing Simulation Experiments, **Proceedings of Winter Simulation Conference**, 2004, s,73-74

⁴² **Fundamentals of Simulation**, <http://euclid.ii.metu.edu.tr/ion561/demo/lectures/charter01/indexhtml>, Erişim 11.11. 2000

⁴³ K.K. CHAN, T.A. SPEDDING, An Integrated Multidimensional Process Improvement Methodology For Manufacturing Systems, **Computers and Industrial Engineering**, Vol. 44, 2003 . s.675-680

Simülasyon modelleri oldukça büyük ve işlenecek veri de oldukça fazla olduğundan, simülasyonun çalıştırılması için bilgisayara ihtiyaç vardır. Buna rağmen küçük modellerin simülasyonunu el ile gerçekleştirmek mümkündür⁴⁴. Kesikli simülasyonda, simülasyonda kullanılan değişkenler daha önceden belirlenmiş zaman aralıklarında gözlemlenir ve aldığı değerler kaydedilir. Ancak bazı durumlarda, simülasyonu sadece belirli zaman aralıklarında gözleme uygun olmaz. Bu durumda, sürekli simülasyondan yararlanmak gerekir.

Kesikli simülasyonda iki temel öge söz konusudur. İlki, bir sonraki olayın ne zaman meydana geleceğini belirten kurallar ve diğeri de bir olay meydana geldiğinde, modelin durumunun değişmesi ile ilgili kurallardır.

1.7.2.3.1 Kesikli Simülasyonda Zaman İlerletme Mekanizması

Simülasyon modellerinde, simüle edilen zamanın değerini gösteren değişkene simülasyon saati denir. Bu saat olay listesinde belirtilen olayların tümünün, kronolojik sıraya bağlı olarak doğru bir biçimde gerçekleştiğini garanti altına alan ve simülasyon saatinin ilerletilmesini sağlayan bir mekanizmadır. Olay listesi gelecek zaman içinde gerçekleştirilecek olayları da çizelgeler. Gelecekteki olayın çizelgelenmesi, faaliyetin başlangıç zamanının, bir istatistiksel dağılımdan elde edilen örnek gerçekleşme zamanının ve olay zamanının, bir arada olay listesinde ifade edilmesidir⁴⁵.

Herhangi bir t zamanında gelecek olay listesi önceden çizelgelenmiş bütün gelecek olayları ve onlara ilişkin olay zamanlarını içerir (t_1, t_2 ..gibi) .

Gelecek olay listesi olay zamanına göre sıralanır. Bunun anlamı ise olayların kronolojik olarak düzenlenmesidir ki;

$$t < t_1 \leq t_2 \leq t_3 \leq \dots \leq t_n$$

⁴⁴ Jerry BANKS, John CARSON, Barry NELSON, David NICOL, **Discrete Event System Simulation**, Prentice Hall, New Jersey, 2000, s.126

⁴⁵ Michael PIDD, **Tools for Thinking (Modelling in Management Science)**, John Wiley Sons, Londra, 1998, s. 254

t zamanı saatin değeridir. Yani simüle edilmiş zamanın mevcut değeridir. t_1 zamanı ile ifade edilen olay anlık olaydır ki bir sonraki olayın meydana geleceğini ifade eder. t simülasyon zaman değeri güncellendikten sonra saat değeri t den t_1 e geçer ve olay gerçekleştikten sonra anlık olay gelecek olay listesinden silinir. Anlık olayın gerçekleşmesi t zamanındaki sistem durumuna bağlı olarak yaratılmış t_1 zamanındaki yeni sistem durumudur. t_1 zamanında yeni gelecek olaylar türetilebilir ya da türetilmeyebilir. Eğer türetilmişlerse uygun pozisyonları gelecek olay listesinde ifade edilir.

t_1 zamanı için yeni sistem durumu güncellendikten sonra, saat değeri yeni anlık olay zamanı kadar artırılır ve olay gerçekleştirilir. Bu süreç, simülasyon sona erene kadar devam eder. Simülasyon dilinin, saati ilerletmesi ve yeni sistem durumunu oluşturması için gerekli faaliyetler dizisine zaman ilerletme mekanizması adı verilir⁴⁶.

Simülasyon saatini ilerletmek için iki yaklaşım öne sürülmüştür. Bunlar; sonraki olay zaman ilerletme yöntemi ve sabit artışlı zaman ilerletme yöntemidir. Birinci yöntem tüm ana simülasyon dillerinde kullanılır ve genel amaçlı diller ile simülasyon yapanlar tarafından tercih edilirler. Bu yöntemde, simülasyon saati sıfırlanır ve ilerideki olayların oluşum zamanları belirlenerek, simülasyon saati ileride ilk olayın saatine ilerletilir ve sistemin durumu güncelleştirilir. Simülasyon saati bu noktadan sonra olacak ilk olayın, oluş saatine kadar ilerletilir ve sistemin durumu tekrar güncelleştirilir. Sistemin ilerleyişi bazı özel durdurma koşulları devreye girene kadar devam eder. Diğer yöntem ise, simülasyon saatinin düzenli artışının tercih edilmesidir.

1.7.2.3.2 Kesikli Simülasyonun Elemanları

Kesikli Simülasyon sistemlerinin paylaştığı ortak elemanlar aşağıdaki gibi özetlenebilir⁴⁷:

⁴⁶ age; s.256

⁴⁷ j. BANKS, J.S. CARSON, B.L. NELSON, D. NICOL, **Discrete Event System Simulation**, Prentice Hall, 2000, s.17

- **Sistem Durumu:** Belirli bir anda, sistemi tanımlamak için gerekli olan değişkenlerin aldığı değer.
- **Simülasyon Saati:** Simüle edilmiş zamanın, mevcut değerini veren değişken.
- **Olay Listesi:** Bir sonraki olayın, oluş zamanını içeren liste.
- **İstatistiksel Sayaçlar:** Sistem performansı hakkında istatistiki bilgileri depolamaya yarayan değişkenler.
- **Başlangıç Yordamı:** Simülasyon modellerinin başlangıç koşullarını hazırlayan alt program.
- **Zamanlama Yordamı:** Olay listesinden, bir sonraki olayı belirleyen ve simülasyon saatini oluş zamanına ilerleten alt program.
- **Olay Yordamı:** Sistemin durumunu, olay olduğu zaman güncelleştiren alt program.
- **Kütüphane Yordamı:** Simülasyon modellerinin bir parçası olarak tanımlanan olasılık dağılımlarından, rastsal sayı gözlemleri oluşturmaya yarayan alt programlar kümesi.
- **Raporlama Yordamı:** İstenilen performans ölçülerinin değerini hesaplayan ve raporlayan alt program.
- **Ana Program:** Sonraki olayın zamanını belirlemek için, zamanlama yordamını çağıran ve sistemin durumunu güncelleştirmek için olay yordamını kullanan alt programdır.

1.7.2.4 Sürekli (Continuous) Simülasyon

Sürekli Simülasyon, zamana bağlı olarak durum değişkenlerinin sürekli değiştiği sistemlerin modellemesini içerir⁴⁸. Bu modellemeler, durum değişkenlerinin oransal değişimi ile zaman arasındaki ilişkiyi veren diferansiyel denklemler gerektirirler.

⁴⁸Hossein ARSHAM, *Modelling & Simulation*, <http://cs.sun.ac.za/lynette/simulation/harsham.html>

Eğer diferansiyel denklemler basit iseler, “sıfır” zamanda durum değişken değerlerinin fonksiyonu olarak, tüm zaman değerlerinde analitik olarak çözülebilirler, fakat çoğu simülasyon modelinde analitik çözüm mümkün değildir. Bu nedenle Runga Kutta entegrasyonu gibi diferansiyel denklemlerin belirli bir aralıkta çözümünü veren nümerik analiz teknikleri kullanılır⁴⁹. Genel olarak, sürekli simülasyon, bir sistemde sürekli değişimler olduğunda kullanılır ve değişkenler diferansiyel denklemler ile değişik değerler alır⁵⁰.

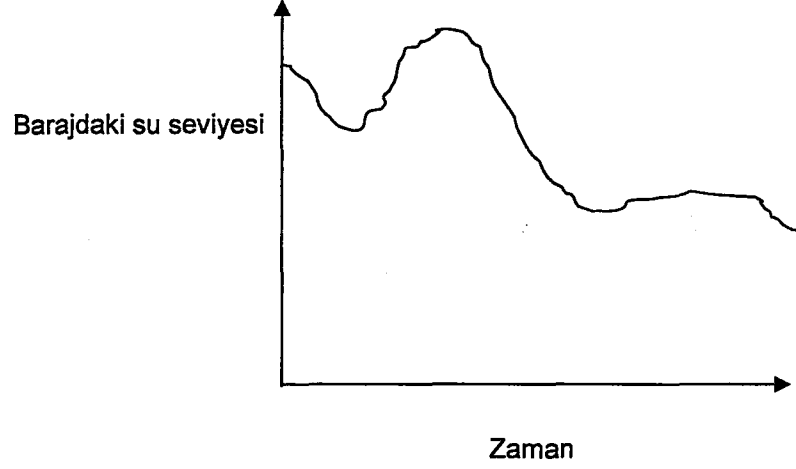
Sürekli simülasyon, durum değişkenlerinin zamana bağlı olarak değiştiği sistemle ilgilenmektedir. Sürekli simülasyonun, bir fuel-oil üretim sistemindeki ortalama tank düzeyini hesaplama gibi önemli uygulanma yerleri bulunmaktadır. Sürekli simülasyon modelleri, zaman ile durum değişkenlerinin değişim oranları için bağıntı kuran diferansiyel denklemleri içermektedir⁵¹. Bu iki simülasyon türüne ek olarak, birleşik simülasyon olarak adlandırılan bir diğer simülasyon türü simülasyonla ilgili birçok yayında görülmektedir. Öte yandan bu durum, sadece kesikli ve sürekli simülasyonun bir karışımı olduğu için, birleşik simülasyonu üçüncü bir tür olarak sınıflandırmaya gerek yoktur.

Bir barajdaki su miktarının zamana bağlı olarak denetimi, sürekli simülasyona örnek gösterilebilir. Yağmur esnasında ve sonrasında barajın içine su birikir. Elektrik elde etmek için su serbest bırakılır. Aynı zamanda buharlaşma ile su seviyesinde azalma olur. Şekil 2, sürekli sistem içerisinde değişen durum değişkenini ifade eder.

⁴⁹ Shepley L. ROSS, **Differential Equations**, Third Ed., John Wiley & Sons, Newyork 1984, s.403

⁵⁰ Dale E. MARTIN, Philip A. WILSEY, Scheduling Optimization on the Simbus Backplane, **Proceedings of The 37. Annual Simulation Symposium**, Virginia, 2004, s. 231-235

⁵¹ M. Law, D.KELTON, **Simulation Modeling and Analysis**, Mc Graw Hill, New York 1991, s.109



Şekil 2 Sürekli Simülasyon Durum Değişkeni

ACSL, CSSL gibi birçok simülasyon dili, sürekli simülasyon modeli kurmak için tasarlanmıştır. SIMAN, SIMSCRIPT II, SLAM II gibi kesikli simülasyon dilleri ise; hem sürekli, hem de kesikli sistemleri temsil etme özelliklerine sahiptir⁵².

⁵² Michael PIDD, An Introduction to Computer Simulation, **Proceedings of the 1994 Winter Simulation conference**

İKİNCİ BÖLÜM

SİMÜLASYON VE MODELLEME

2.1 EL İLE SİMÜLASYON (MONTE CARLO SİMÜLASYONU)

Gerçek hayatta karşılaşılan davranış ve işlemleri genellikle önceden kesin olarak bilmek mümkün olmamaktadır. Monte Carlo yöntemi, bu olasılıklı durumları simülasyona dahil edebilmeyi sağlayan ve tesadüfi sayılar tablosunu kullanan bir yöntemdir⁵³. Simülasyon, başlangıç yordamı aracılığıyla başlatılır ve model varlıkları için ilişkiler tanımlanır. Daha sonra girdi değişkenlerinin tesadüfi olması nedeniyle, sistemdeki değişim için çeşitli nümerik değerler hesap edilir. Bu yeni durum, diğer bir geçiş için başlangıç noktasını oluşturur.

Monte Carlo Simülasyonu, incelenen değişkenlerin olasılık dağılımlarının belirlenmesini, dağılımların kümülatif hale dönüştürülmesini, kümülatif dağılımlardan gruplar oluşturulmasını ve sonra elde edilen verilere rasgele sayıların uygulanmasıyla dağılımdan örnekleme kapsar. Zaman boyunca her bir rasgele değişkenin hareketini tanımlamak için, bir dizi rasgele sayı kullanılır. Monte Carlo Simülasyonu, yöneticilere tesadüfi etkenlerin davranışıyla düzenlenebilecek değişken politikalar veya örgütsel şartların belirlenmesine olanak sağlar.

Monte Carlo Simülasyonu, gerçek bir durumun istatistik modelini kurar, sonra model üzerinde örnek denemeler yapar. Bu teknikte aşağıdaki adımlar izlenir⁵⁴:

- 1- Simülasyon modelinde kullanılan frekans dağılımları, kümülatif olasılık dağılımlarına dönüştürülür. Bu veriler, belirli aralıklara bölünerek, değişkenin alacağı olası değerler için gruplar oluşturulur.

⁵³ Aykut TOP, **Üretim Sistemleri**, Alfa Basım Dağıtım, İstanbul, 1996, s.197

⁵⁴ John A. LAWRENCE, Barry A. PASTERNAK, **A computer Integrated Approach for decision Making**, John Willey Sons, USA, 1998, s. 585-595

- 2- Simülasyonun her replikasyonda kullanılacak spesifik değişken değerini belirlemek için, İlk adımda oluşturulan kümülatif olasılık dağılım grupları, rastgele sayı değerleri için yeniden düzenlenir. Örnekleme için, rasgele sayılar tablosu kullanılarak her replikasyon için alınan rastgele sayıya karşılık gelen grubun değişken değeri belirlenmiş olur.
- 3- Gerekli replikasyon sayısı kadar simülasyon tekrarlanır ve istatistiksel analizler yapılır.

El ile simülasyonda, frekans dağılımının seçimi büyük önem taşır. Dağılımlar geçmiş verilerden veya deneyimlerden elde edilir. Rastgele sayılar tablosundaki değerlerin geçerliliği Chi-kare testi ile denenebilir. Monte Carlo Simülasyonun, bekleme hattı, işyeri yerleşim problemleri, montaj hatları, envanter problemleri, techizat yenileme, tamir bakım planlaması vb. gibi bir çok kullanım alanı vardır. Monte Carlo Simülasyonunu daha iyi açıklayabilmek için, aşağıdaki örneği inceleyelim :

Bir hava yolu şirketi, farklı hizmetlerde kullanılmak üzere yedek parçalar satın almaktadır. Bir dişli kutusu içinde üç tür rulman sorun yaratmaktadır. Geçmişte, bozulan bir rulmanın değiştirilmesi şeklinde politika izlenmiştir. Koruyucu bakım sorumlusu, maliyetlerin düşürülmesine yönelik aşağıdaki üç öneriyi incelemektedir.

- A- Bir rulman bozulduğunda, onu değiştirmek
- B- Bir rulman bozulduğunda, 3'ünü birden değiştirmek
- C- Bir rulman bozulduğunda, onu ve onunla birlikte 1700 saat ve daha fazla kullanılmış olanları da değiştirmek.

Onarımla ilgili veriler bakım elemanı için;

- | | |
|-----------------------|----------|
| 1 Rulmanı değiştirmek | : 5 saat |
| 2 Rulmanı değiştirmek | : 6 saat |
| 3 Rulmanı değiştirmek | : 7 saat |

Bakım elemanının saat ücreti : 9 \$ / saat

Rulman maliyeti :15\$ /adet

Sistemin durmasının maliyeti : 6\$ / saat

200 Rulman için, aşağıdaki geçmiş verilerden alınan rulman ömürlerinden hareketle, kümülatif olasılıkları ve buna bağlı Monte Carlo sayıları Tablo 2'de gösterilmiştir.

Tablo 2 Kümülatif Olasılıkların Ve Monte Carlo Sayılarının Belirlenmesi

Rulman ömrü (saat)	Rulman sayısı	Olasılık	Kümülatif olasılık	Monte Carlo Sayıları
1100	3	0.015	0.015	001-015
1200	10	0.050	0.065	016-065
1300	12	0.060	0.125	066-125
1400	20	0.100	0.225	126-225
1500	27	0.135	0.360	226-360
1600	35	0.175	0.535	361-535
1700	30	0.150	0.685	536-685
1800	25	0.125	0.810	686-810
1900	18	0.090	0.900	811-900
2000	15	0.075	0.975	901-975
2100	4	0.020	0.995	976-995
2200	1	0.005	1.000	996-000
TOPLAM	200			

Kümülatif olasılık dağılımlarının ve Monte Carlo sayı gruplarını oluşturduktan sonra, rulmanların bozulma zamanlarını temsil edecek rastgele sayıların seçimi yapılmalıdır. Bu amaçla aşağıdaki rastgele sayıları kullanarak 15.000 saatlik çalışma için alternatif önerilerin sonuçlarını inceleyelim:

1 Rulman	841, 584, 157, 599, 436, 255, 982, 525, 265, 247, 383, 188, 517, 883, 104
2 Rulman	848, 888, 534, 412, 059, 501, 084, 899, 836, 715, 887, 878, 896, 377, 703

3 Rulman 501, 921, 522, 870, 813, 446, 252, 378, 125, 316, 588, 522, 026, 616, 933

Rastgele sayılar tablosu incelendiğinde 1. rulman için 841 sayısının, 1900 saatlik ömüre (bozulma süresi), 2. rulman için 848 sayısının da yine 1900 saatlik ömüre ve 3. rulman için, 501 rastgele sayısının ise, 1600 saatlik ömüre denk geldiği Tablo 2'den görülmektedir. A planı bozulan rulmanı değiştirmeyi öngördüğünden dolayı, ilk bozulan rulman 1600 saatle 3. olmakta, daha sonra 300 saatin geçmesinin ardından, 1900 saate ulaşıldığında 1. ve 2. rulmanların aynı anda bozulduğu gözlemlenmektedir. Böylece rastgele sayıların tümünün kullanılması ile değişecek rulmanlar ve bozulma süreleri Tablo 3'de görülmektedir.

Tablo 3 A Planı İçin Rulman Bozulma Süreleri

No	1	No	2	No	3
1900		1900		1600	
1700		1900		2000	
1400		1600		1600	
1700		1600		1900	
1600		1200		1900	
1500		1600		1600	
2100		1300		1500	
1600		1900		1600	
1500		1900		1300	
1500		1800		1500	
1600		1900		1700	
1400		1900		1600	
1600		1900		1200	
1900		1600		1700	
1300		1800		2000	

Simülasyonun ilk 1600 saatlik çalışmasıyla bozulan ve değişen 3. rulman, daha sonra ilerleyen 300 saatin ardından da 1900. saatte, 1. ve 2. rulmanların değişimi ve 15000 saatlik çalışmanın sonucu tüm rulman değişimleri, değişimler

sırasında rulman maliyetinin yanı sıra, bakım elemanının ve duruşların maliyetleri Tablo 4'de görülmektedir.

Tablo 4 A Planı İçin Maliyetler

Rulman çalışma Süreleri			Makinenin kümülatif çalışma süresi	Değişen rulman no	Rulman maliyeti	Ücretler	Durma Maliyeti
no 1	no 2	no 3			\$	\$	\$
1600	1600	<u>1600</u>	1600	3	15	45	30
<u>300</u>	<u>300</u>	300	1900	1-2	30	54	36
<u>1700</u>	1700	<u>1700</u>	3600	1-3	30	54	36
200	<u>200</u>	200	3800	2	15	45	30
<u>1200</u>	1200	1200	5000	1	15	45	30
200	200	<u>200</u>	5200	3	15	45	30
200	<u>200</u>	200	5400	2	15	45	30
<u>1300</u>	1300	1300	6700	1	15	45	30
300	<u>300</u>	300	7000	2	15	45	30
100	100	<u>100</u>	7100	3	15	45	30
1100	<u>1100</u>	1100	8200	2	15	45	30
<u>100</u>	100	100	8300	1	15	45	30
700	700	<u>700</u>	9000	3	15	45	30
<u>800</u>	<u>800</u>	800	9800	1-2	30	54	36
800	800	<u>800</u>	10600	3	15	45	30
500	<u>500</u>	500	11100	2	15	45	30
<u>800</u>	800	800	11900	1	15	45	30
200	<u>200</u>	<u>200</u>	12100	3	15	45	30
900	<u>900</u>	900	13000	2	15	45	30
<u>500</u>	500	500	13500	1	15	45	30
200	200	<u>200</u>	13700	3	15	45	30
1200	<u>1200</u>	1200	14900	2	15	45	30
<u>100</u>	100	<u>100</u>	15000	1-3	30	54	36
TOPLAM MALİYET					405	1071	714

B planında ise bir rulmanın bozulmasıyla, her üç rulmanda değiştirileceği için, Tablo 2 de, her sıradaki en küçük değerli sürenin tamamlanması ile rulmanların biri bozulmakta ama, her üçü birlikte değiştirilmektedir.

Bu deęişimlerin zamanları Tablo 5'de, deęişimler sırasında rulman, bakım elemanı ve duruş maliyetleri de Tablo 6'da görölmektedir.

Tablo 5 B Planı İçin Bozulma Süreleri

		Rulman Çalışma Süresi			Makine Kümülatif Çalışma Süresi
No	1	No	2	No	3
				1600	1600
1700					3300
1400					4700
		1600			6300
		1200			7500
1500					9000
		1300			10300
				1600	11900
				1300	13200
				1500	14700
					15000

Tablo 6 B Planı İçin Bozulma Maliyeti

İşçilik	63 \$ (7x9)
Rulman maliyeti	45 \$ (3x15)
Duruş maliyeti	42 \$ (7x6)
Toplam	150 \$

1 bozulma 150 \$ x 10 bozulma = 1500 \$

C Planı ise, bir rulman bozulduğunda bozulan rulmanla birlikte, ömrünün 1700 saatini tamamlayan rulmanlarda deęiştirilmektedir. Tablo 7, bu plan için bozulma zamanlarını ve plan gereęi bozulan rulmanla birlikte 1700 saat

çalışmış rulmanların da değişim zamanlarını, maliyetleri ile birlikte göstermektedir.

Tablo 7 C Planı İçin Maliyetler

Rulman çalışma Süreleri			Makinenin kümülatif çalışma süresi	Değişen rulman no	Rulman maliyeti	Ücretler	Durma Maliyeti
no 1	no 2	no 3			\$	\$	\$
1600	1600	<u>1600</u>	1600	3	15	45	30
<u>300</u>	<u>300</u>	300	1900	1-2	30	54	36
<u>1700</u>	<u>1700</u>	<u>1700</u>	3600	1-2-3	45	63	42
<u>1400</u>	1400	1400	5000	1	15	45	30
200	<u>200</u>	<u>200</u>	5200	2-3	30	54	36
<u>1500</u>	1500	1500	6700	1	15	45	30
100	<u>100</u>	100	6800	2	15	45	30
300	300	<u>300</u>	7100	3	15	45	30
900	<u>900</u>	900	8000	2	15	45	30
<u>300</u>	300	300	8300	1	15	45	30
700	700	<u>700</u>	9000	3	15	45	30
600	<u>600</u>	600	9600	2	15	45	30
<u>200</u>	200	200	9800	1	15	45	30
800	800	<u>800</u>	10600	3	15	45	30
300	<u>300</u>	300	10900	2	15	45	30
<u>1000</u>	1000	1000	11900	1	15	45	30
200	200	<u>200</u>	12100	3	15	45	30
700	<u>700</u>	700	12800	2	15	45	30
<u>700</u>	700	700	13500	1	15	45	30
200	200	<u>200</u>	13700	3	15	45	30
900	<u>900</u>	900	14600	2	15	45	30
<u>400</u>	400	400	15000	1	15	45	30
TOPLAM MALİYET					390	1026	684

Tablo 8 A, B ve C Planları İçin Karşılaştırmalı Toplam Maliyetler

Plan A:	$405 + 1071 + 714 = 2190$ \$
Plan B:	$= 1500$ \$
Plan C:	$390 + 1026 + 684 = 2100$ \$

Tablo 8'de her üç planın toplam maliyetleri incelendiğinde, en iyi stratejinin B planı olduğu görülür. Yani bir rulman bozulduğunda, rulmanların üçünü birden değiştirmek tamir bakım maliyetlerini minimize etmektedir⁵⁵.

2.2 VERİ TOPLAMA VE ANALİZİ

Simülasyonda kullanılacak verilerin üç türü vardır⁵⁶. Birincisi girdi parametrelerinin tahminlenmesinde kullanılmak üzere toplanan verilerdir. Girdi parametreleri, simülasyon boyunca değişmeyen işlem zamanı, bir malzemenin maliyeti vb. gibi sabit değerlerdir.

İkinci tür veriler, modelde kullanılacak tesadüfi değişkenlerin, olasılık dağılımlarının tahminlenmesinde gerekli verilerdir. Bir makinenin bozulma ve onarım zamanları gibi. Olasılık dağılımlarını tahminlemek için iki yaklaşım vardır. Birincisi, basit olarak geçmiş gözlem değerlerini kullanmak ve yapay bir dağılım elde etmek için sıklık diyagramını ya da histogramını çizmektir. Gözlenen verilerden yola çıkılarak oluşturulan histogramlar ana kitlenin şekli hakkında bir fikir edinilmesini sağlar. Genelde histogram oluşturmak için kullanılan veriler, geçmişten ve gözlenen verilerden alınmalıdır.

Bir histogram oluşturmada, verilerin gözlemlendiği ana kitle şeklini tanımlayan sınıf aralığı çok önemli bir seçimdir. Sonra, simülasyonda kullanılacak tesadüfi değişkenlerin spesifik değerlerini belirlemek için yapay dağılımdan örneklemeler yapılır.

İkinci yaklaşım da, toplanan verileri bir teorik dağılıma uydurmaktır. Bir değişkenin pozitif sonsuz değer alabileceği bazı sürekli dağılımlar, uniform, üstel, normal ve beta dağılımlarıdır. Bir değişkenin sonlu bir değer alabileceği kesikli dağılımlar da, Bernoulli, binom, geometrik, ve poisson dağılımlarıdır. Belirli tesadüfi değişkenlere hangi olası teorik dağılımın uygun olduğunu belirlememiz

⁵⁵ Richard TERSINE, **Production Operations Management**, Elsevier, Newyork 1990, s.665-668

⁵⁶ Lawrence LEEMIS, Input Modelling, **Proceedings of Winter Simulation Conference**, 2003, s.14

ve parametre değerlerini tahminlememiz gereklidir. Bu amaçla da iyi uyum testleri (Goodness-of-fit) kullanılır.

Son olarak, gerçek sistemin performansını ölçmek için veriler toplanır. Örneğin, bir makinenin ve tamir elemanının boшта geçen sürelerinin gerçek gözlem değerleri, simülasyon sonuçlarının geçerliliğini sınamak amacıyla kullanılır⁵⁷.

İstatistiksel dağılımların özellikleri ile bunların uyum testleri, simülasyon modelinin kurulması ve gerçek sistemi doğru bir biçimde temsil etmesi açısından son derece önemli olduğundan dolayı, bu dağılımların oluşmasında esas teşkil eden tesadüfi sayıların, gerektiği biçimde türetilmesi ve özellikleri, daha detaylı olarak bundan sonraki başlıklarda ele alınacaktır.

2.3 TESADÜFİ SAYILARIN TÜRETİLMESİ

Simülasyon ile modellemede, tesadüfi olaylar belirleyici bir yere sahiptir. Tesadüfi sayılar, sıfır ile bir değeri arasında tekdüze olarak dağılmış sayılardır. Bunların her birinin seçilme olasılığı diğerine eşittir. Simülasyon çalışmalarında, tesadüfi sayılar ile iki yönden ilgilenilir⁵⁸:

- Tesadüfi sayıların türetilmesi,
- Türetilen sayıların tesadüflüklerinin test edilmesi.

Günümüzde tesadüfi sayı türetimi için, bilgisayardan yararlanılmaktadır. Tesadüfi sayı türeten aygıtların şu özelliklere sahip olması beklenir:

- Türetilen sayılar tekdüze(dikdörtgen) dağılıma uymalıdır.
- Tesadüfi sayı üretici hızlı olmalıdır.
- Tesadüfi sayı üretici program, bilgisayar belleğinde simülasyonun hızını yavaşlatmaması açısından, fazla yer kaplamamalıdır.

⁵⁷ Edwood BUFFA, Rakesh K. SARIN, **Modern Production and Operations Management**, John Wiley Sons, Canada, 1997. s.266-267

⁵⁸ W.J. FABRYCKY, J. H. MIZE, **Simulation Modelling and Simnet**, Prentice Hall, New Jersey, 1998, s. 37-45

- Üretici farklı bir rastgele sayı kümesi türetebilmelidir.

2.4 RASTGELELİK TESTLERİ

Bilgisayarla tesadüfi sayı üretildiği zaman, tesadüfi sayılar dizisinin gerçekten tesadüfi olup olmadığını anlamak için burada kısaca verilecek olan testler uygulanmaktadır.

- **Frekans Testi:** Üretilen sayıların tekdüze dağılıma uygunluğunu anlamak için, Chi kare veya Kolmogorov Smirnov testi uygulanır.
- **Aralık Testi:** Belirli bir basamağın tekrarı arasında görülen basamak sayısı sayılır ve daha sonra ortalama değere göre Chi kare testi uygulanır.
- **Koşum Testi:** Ortalamaya göre alt ve üstte bulunmayı test eder. Bu test gerçek koşumların oluşum sayısını sayma testidir ve bu sayı Chi kare ile ortalama göre karşılaştırılır.
- **Spektral testi:** Fourirer analizine göre, N sayılar kümesinin bağımsızlığı ölçülür. Bu test Knuth testi olarak bilinir.
- **Poker Testi:** Poker oyunu eline eşdeğer bir testtir. Bu test beş veya daha fazla basamak kombinasyonunun, tüm basamakların farklı, bir çift üçlü, full, vb. olmasına göre beklenen oluşuma karşılık test edilir.
- **Oto Korelasyon Testi:** $k = 1, 2, 3, \dots$ değerlerini almak üzere rastgele sayı üretiminde x_n ve x_{n+k} arasındaki korelasyon test edilir⁵⁹.
- **d veya Uzaklık Testi:** Birim karenin köşe noktası olarak üretilen rastgele sayı çifti koordinat olarak düşünülür ve iki nokta arasındaki mesafenin karesi, verilen denklemler kümesinin teorik olasılıklarına karşılık test edilir⁶⁰.
- **Sıralı İstatistik Testi:** Ardışık n sayının, maksimum veya minimum değeri test edilir.

⁵⁹ Daha geniş bilgi için: <http://mathworld.wolfram.com>

⁶⁰ Osman HALAÇ, **İşletmelerde Simülasyon Teknikleri**, İstanbul Matbaası, İstanbul, 1982, s.52

Çok kullanılan tesadüfi sayı üretme algoritmaları, Von Neumann'ın Midsquare metodu, Linear Congreuential metodu, Additive Congreuential metodu ve Quadratic Congruence metodudur⁶¹. Bu metotlar, bağımsız ve tekdüze dağılımlı rastgele sayılar üretmektedir. Bu sebeple amaç, bu değerleri normal ya da üstel dağılımlara yaklaştırmaktır. Bu dağılım yaklaştırma işlemine, direkt yaklaşım denir. Direkt yaklaşımlar, beklenen dağılım ya da tesadüfi değişkeni direkt olarak kullanırlar. Direkt yaklaşımlar, Ters Dönüşüm Metodu, Kompozisyon Metodu ve Helezon Metodu' dur. Eğer direkt yaklaşımlar yetersiz olursa, o zaman dolaylı yaklaşımlar kullanılabilir. En yaygın kullanılan yaklaşımlardan biri Kabul- Ret Metodu' dur⁶².

ARENA SE V2.2' nin kendi rastgele sayı üreticisi bulunmaktadır. ARENA SE V2.2' de kullanılan rast gele sayı üreticisi Linear Congreuential metottur. Bunun on tane rast gele sayı grubu vardır ve her sayı grubu 10000 rastgele sayıdan oluşur. Her sayı grubu birbirinden bağımsızdır. Arena, rastgele sayıları ürettikten sonra, yine test ettiğinden son derece güvenilirdir⁶³.

2.5 İSTATİSTİKSEL DAĞILIMLAR VE UYGUNLUK TESTLERİ

Dağılımlarla ilgili olarak aşağıdaki semboller sıkça kullanılmaktadır:

X :	tesadüfi değişken
$f(x)$:	x'in Olasılık Yoğunluk Fonksiyonu (Sürekli)
$p(x)$:	x'in Olasılık Yoğunluk Fonksiyonu (Kesikli)
a :	Minimum Değer
b :	Maksimum Değer
m :	Mod Değeri
μ :	Ortalama = E[x]

⁶¹ Daha geniş bilgi için: J.BANKS, J.CARSON, B.NELSON, D.NICOL, **Discrete Event System Simulation**, Prentice Hall, 2000

⁶² A. HOUSHYAR, V. NUILA, Required Steps For Succesfull Design And Implementation Of Simulation, **Computers and Industrial Engineering**, Vol. 25, 1993, s.53-56

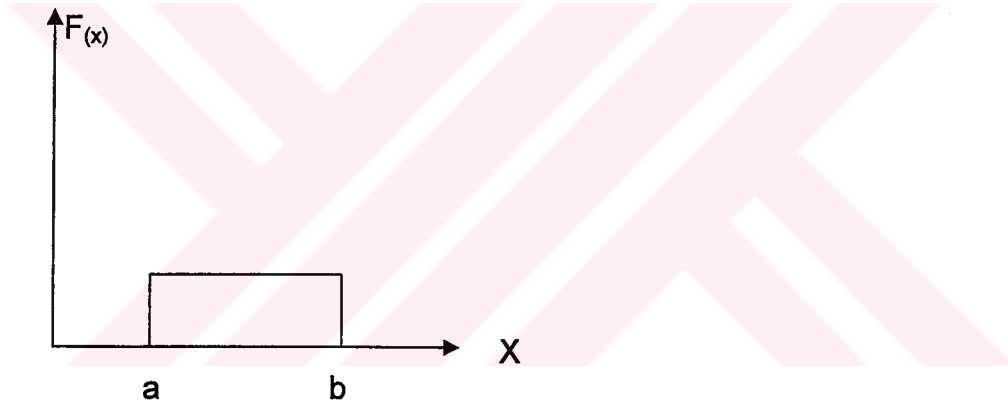
⁶³ ARENA SE 2.0, Tutorial

- σ : Standart Sapma
 σ^2 : Varyans = $E[(x-\mu)^2]$
 α ve β : Yoğunluk fonksiyonunda kullanılan parametreler

Aşağıdaki istatistiksel dağılımlar için daha kapsamlı bilgi <http://mathworld.wolfram.com> kaynağından alınabilir.

2.5.1 Dikdörtgen Dağılım

Dikdörtgen şeklinde olan bu dağılımda, a ve b arasındaki her değer, eşit olasılık değerine sahiptir. Bu dağılım tekdüze(uniform) dağılımın n=2 parametrelili özel halidir.



Şekil 3 Dikdörtgen Dağılım

Dikdörtgen dağılımının, olasılık yoğunluk fonksiyonu, ortalaması ve varyansı aşağıdaki gibidir.

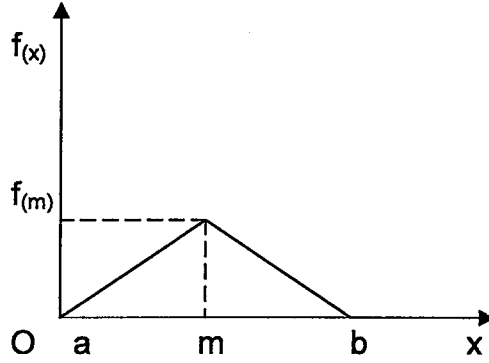
$$f(x) = \frac{1}{b-a} ; a < x < b$$

$$\mu = \frac{a+b}{2}$$

$$\sigma^2 = \frac{(b-a)^2}{12}$$

2.5.2 Üçgen Dağılım

Biri a 'dan m 'ye artan, diğeri m 'den b 'ye azalan iki doğrusal kısma sahiptir.



Şekil 4 Üçgen Dağılım

Üçgen dağılımın olasılık yoğunluk fonksiyonu, ortalaması ve varyansı aşağıdaki gibidir.

$$f(x) = \begin{cases} \frac{2(x-a)}{(m-a)(b-a)}; a < x < m \\ \frac{2(b-x)}{(b-m)(b-a)}; m < x < b \end{cases}$$

$$\mu = \frac{a+m+b}{3}$$

$$\sigma^2 = \frac{a(a-m) + m(b-a) + b(m-b)}{18}$$

2.5.3 Geometrik Dağılım

Geometrik dağılım, binom dağılımına yakın bir dağılımdır. Her ikisi de aynı tür deney üzerine kurulmuşlardır. Yani, bağımsız denemeler sonucunda başarı olasılığı p , başarısızlık olasılığı da $(1-p)$ dir. Geometrik dağılımda, ilk başarısızlık oluştuğu ana kadar ki dağılımla ilgileniriz.

X rastsal değişkeni, ilk başarısızlığa kadar geçen süreyi gösterebilir. X 'in Ortalama dağılım fonksiyonu şöyledir:

$$f(x) = \Pr(X=x)$$

$$f(x) = p^{x-1}(1-p) \quad x=1,2,\dots$$

$$0 \leq p \leq 1$$

$f(x)$, ilk $(x-1)$ denemenin başarılı olması olasılığı ile x inci denemenin başarısızlıkla sonuçlanması olasılığını göstermektedir. İşte bu, geometrik dağılım olarak bilinir⁶⁴.

X 'in beklenen değeri,

$$E(x) = \sum_{x=1}^{\infty} x p^{x-1}(1-p)$$

$$= \frac{1}{1-p} \quad \text{olur.}$$

$$\sigma^2 = \frac{1-p}{p^2}$$

⁶⁴ D. MONTGOMERY, G. RUNGER, *Applied Statistics and Probability For Engineers*, John Wiley & Sons, New York , 2003, s.78

2.5.4 Hipergeometrik Dağılım

Doğru seçim için n seçenek, yanlış seçim için m seçeneğin toplamından oluşan $n+m$ olasılıklı bir kümede, N örneklem sayısı olsun. Eğer seçim doğru ise $x_i = 1$, değil ise $x_i = 0$ olsun. X doğru seçimlerin toplam sayısı olmak üzere,

$$x = \sum_{i=1}^N x_i$$

$$f_{(x)} = \Pr(X=x)$$

$$f_{(x)} = \frac{\binom{n}{x} \binom{m}{N-x}}{\binom{n+m}{N}}$$

Hipergeometrik dağılımdır⁶⁵.

X 'in beklenen değeri:

$$E(x) = \frac{nN}{m+n}$$

$$\sigma^2 = \frac{Nnm(n+m-N)}{(n+m)^2(n+m-1)} \text{ olur.}$$

2.5.5 Bernoulli Dağılımı

Kesikli dağılımların en basitidir. Bu dağılımın temel karakteristiği, rastsal değişkenin yalnız iki farklı değer alabilmesidir. Yani, örnek hacmi, iki noktadan ibarettir. Bernoulli dağılımı, matematiksel olarak aşağıdaki gibi ifade edilir.

$$P_1 = P_{(x_1)} = 1-p$$

$$P_2 = P_{(x_2)} = p$$

Burada x_1 ve x_2 rastsal değişken x 'in iki mümkün değeridir. Bernoulli dağılımın bazı örnekleri , yazı-tura, imal edilen bir mamulün kabulü ya da reddi vb. örnek olarak sayılabilir.

⁶⁵ Hülya TÜTEK, Şevkinaz GÜMÜŞOĞLU, *İşletme İstatistiği*, Barış Kitabevi, İzmir 2000, s.78-79

Bernoulli dağılımının birikimli dağılım fonksiyonu aşağıdaki gibi yazılır.

$$F_{(x)} = \Pr(X \leq x) = \sum_{zi < x} f(z) \quad i=1 \text{ veya } 2$$

$$\mu = p$$

$$\sigma^2 = p(1-p)$$

2.5.6 Binom Dağılımı

Bir deneyin, yalnız iki çıktı verdiğini varsayarak; başarıyı 1, başarısızlığı da 0 ile gösterelim. P, her deneyde başarının olasılığını gösterebilir. Deney benzer koşullar altında n kez tekrarlınsın. Eğer, n adet deneydeki başarı sayısı x ise, bu durumda x'in binom dağılım gösterdiği söylenebilir. Bu aşağıdaki gibi ifade edilir;

$$f_{(x)} = \Pr(X=x) = \binom{n}{x} p^x (1-p)^{n-x} \quad x = 0, 1, \dots, n$$

$$0 < p < 1$$

$$\mu = np$$

$$\sigma^2 = np(1-p) \text{ dir.}$$

2.5.7 Poisson Dağılımı

Poisson dağılımı, kesikli bir dağılımdır. Belli bir zaman aralığında meydana gelen olay sayısı ile ilgili bir dağılımdır. Eğer olaylar arasındaki zaman aralıkları üstsel dağılım sergiliyor ve teker teker meydana geliyorsa, sabit bir zaman aralığındaki olay sayısı Poisson dağılımını sergiler. Binom dağılımı yerine kullanılabilir. Poisson dağılımı aşağıdaki gibi ifade edilir.

$$P_{(x=n)} = \frac{e^{-\lambda} (\lambda)^n}{n!} \quad x = 0, 1, 2, \dots$$

$$\sigma^2 = \lambda t \text{ dir.}$$

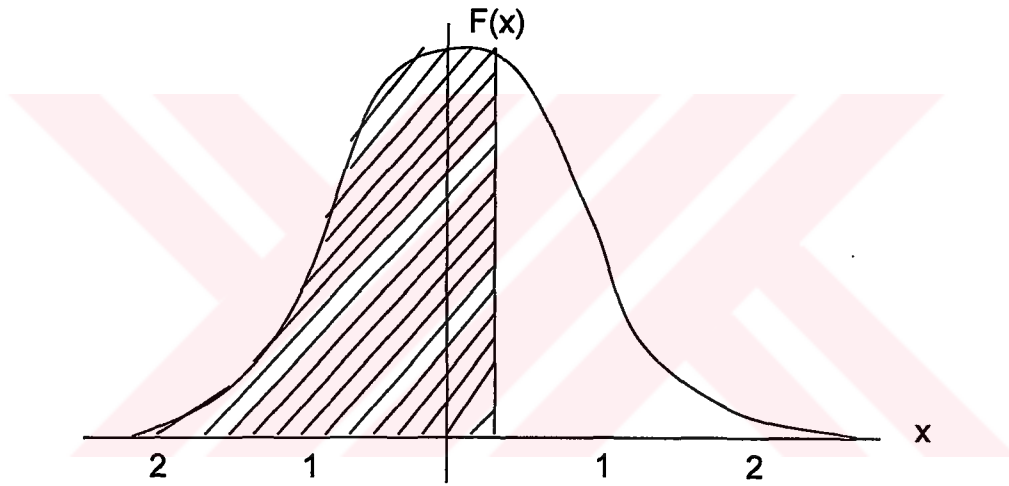
$$E(X) = \lambda$$

2.5.8 Normal Dağılım

İstatistikte en önemli dağılımdır. Merkezi limit teoremine göre, bireysel dağılımları ne olursa olsun, örnek ortalamaları normal dağılım gösterdiklerinden işletmecilikte pek çok uygulaması olan bir dağılımdır. Matematiksel olarak izlenmesi kolaydır.

Sürekli bir dağılım olan Normal dağılım çan eğrisi olarak bilinir ve ortalaması μ ve varyansı σ^2 olan bir Normal Dağılım aşağıdaki şekilde ifade edilir:

$$P(X \leq x) = \frac{x - \mu}{\sigma}$$



Şekil 5 Standart Normal dağılım

Merkezi limit teoremine dayanarak, örneklem sayısı yeterli büyüklüğe ulaştığında,

$$\mu = Np$$

$$\sigma^2 = Npq \quad \text{olur.}$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \quad -\infty < x < \infty \quad \text{olur}^{66}.$$

⁶⁶ Paul NEWBOLD, (4. Basımdan Çeviren Ümit ŞENESEN), *İşletme ve İktisat için İstatistik*, Literatür Yayıncılık, İstanbul 2000, s.211-212

2.5.9 Gamma Dağılımı

Genellikle bekleme zamanlarını modellemekte kullanılan bekleme ve hizmet zamanlarını temsilen α ve θ gibi iki parametrelili bir dağılımdır. Dağılım aşağıdaki şekilde ifade edilebilir:

$$P(x) = \frac{x^{\alpha-1} e^{-x/\theta}}{\Gamma(\alpha)\theta^\alpha}$$

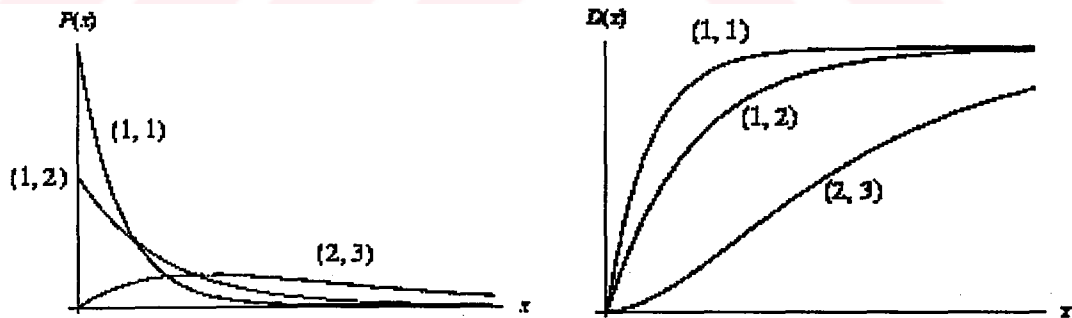
$D(x)$, kuyruk modelindeki bekleme zamanlarını temsil eden ve λ değişim oranına sahip bir Poisson dağılımına bağlı h .nci olaya kadar tanımlanmış bir dağılım fonksiyonu olsun.

$$D(x) = P(X \leq x) = 1 - P(X > x)$$

$$= 1 - \sum_{k=0}^{h-1} \frac{(\lambda x)^k e^{-\lambda x}}{k!}$$

$$= 1 - e^{-\lambda x} \sum_{k=0}^{h-1} \frac{(\lambda x)^k}{k!}$$

$$= 1 - \frac{\Gamma(h, x\lambda)}{\Gamma(h)}$$



Şekil 6 Gamma Dağılımı

2.5.10 Üstsel Dağılım

Herhangi bir Δt zaman aralığında sadece tek bir olayın meydana gelme olasılığı Δt ile orantılı ve bu olayın meydana gelmesi bir önceki olayın oluş zamanından

bağımsızsa, olaylar arasındaki zaman aralıkları üstsel dağılım gösterir. Bu nedenle, eğer bir faaliyet t süresince devam etmiş de olsa, yeni başlamış da olsa, bu faaliyetin gelecek Δt zaman aralığında sona erme olasılığı aynıdır.

Üstsel dağılım, Gamma dağılımından, $\alpha=1$ ve $\beta=1/\lambda$ konulmak suretiyle elde edilir.

$$f(t)=\lambda e^{-\lambda t} ; t=0 \quad \text{ve} \quad \lambda>0$$

$$E(t)=\mu= \frac{1}{\lambda}$$

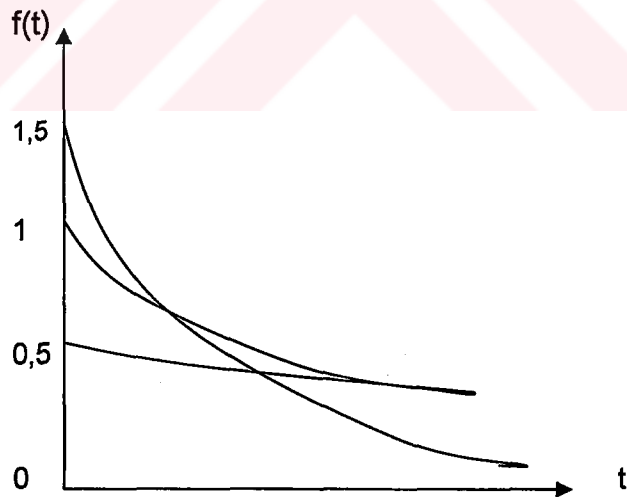
$$\text{Var}(t)=\sigma^2 = \frac{1}{\lambda^2} = \mu^2$$

Üstsel dağılımın, kümülatif dağılım fonksiyonu;

$$F(t) = \Pr(T \leq t)$$

$$F(t) = \int_0^t \lambda e^{-\lambda t} dt$$

$$F(t) = 1 - e^{-\lambda t}$$



Şekil 7 Üstsel Dağılım

Üstsel dağılım ile Poisson dağılımı arasındaki ilişki, bekleme hattı problemlerinde oldukça önemlidir. Bir hizmet birimine ardışık gelişler arasındaki süreler, zaman parametresi λ olan bir üstsel dağılım gösteriyor ise, zaman

birimi başına gelişlerin sayısı, geliş sayısı parametresi λ olan Poisson dağılımına uyar. Yani,

T : Ardışık gelişler arasındaki süre

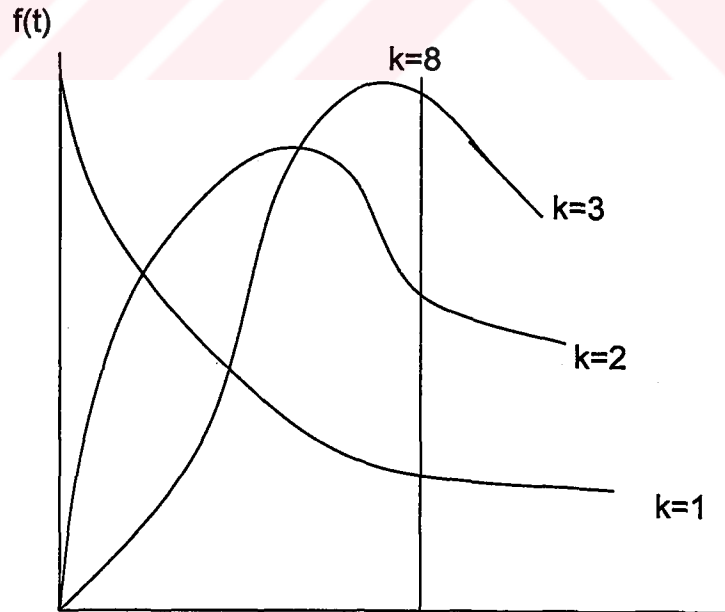
X : Zaman birimi başına gelişlerin sayısı ve

$$F(x) = \lambda e^{-\lambda t} \quad ; t=0 \text{ ve } \lambda > 0 \text{ ise}$$

$$F(x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad ; x=0,1,2,\dots \text{ ve } \lambda > 0$$

2.5.11 Erlang Dağılımı

Birbirinden bağımsız, fakat benzer özellikler gösteren, üstsel dağılmış rastsal değişkenlerin toplamları için kullanılan bir dağılımdır. Gamma dağılımının özel bir şekli olup, daha çok bekleme hattı problemlerinde görülen fazla gelişler ve servis zamanlarının modellenmesinde kullanılır.



Şekil 8 Erlang Dağılımı

Bu dağılımların ortak parametresi $k\mu$, yani $f(t) = k\mu e^{-\mu t}$, ise

$$T = \sum_{i=1}^k T_i \mu$$

Rastsal deęişkeni, ortalaması $\frac{1}{\mu}$ ve varyansı $\frac{1}{k\mu^2}$ olan bir Erlang daęılımını gösterir. Erlang daęılımını, Gamma daęılımında $\alpha=k$ ve $\beta=1/k\mu$ alınarak ařaęıdaki şekilde ifade edilebilir.

$$F(t) = \frac{(k\mu)^k t^{k-1} e^{-k\mu}}{(k-1)!}; t>0, k>0 \text{ ve } \mu >0 \text{ ortalaması, } E(T) = \frac{1}{\mu} \text{ varyansı,}$$

$$\text{Var}(T) = \frac{1}{k\mu^2}$$

2.5.12 Daęılımın Uygunluęunun Testi

Simülasyonla modellemede, gözlenen deęerlerle bunların teorik daęılımlara uygunluęunun test edilmesi gerekir. Bunun için esas olan iki temel test vardır. Chi Kare ve Kolmogorov Smirnov testlerinin her ikisi de üretilmiş olan rastgele sayı kümesinin oluşturduęu örnek daęılım ile teorik dikdörtgen daęılım arasındaki uygunluęun derecesini tespit eder. Aslında bu iki test de, rastgele sayıların oluşturduęu daęılım ile, teorik daęılım arasında dikkate deęer fark olmadığını ifade eden sıfır hipotezine dayanır.

- **Chi Kare Testi**

Chi kare testi, yaygın bir şekilde uygunluk testinde kullanılmaktadır. Bu testin amacı, örnek deęerlerinin gösterdięi daęılımın hipotezde ileri sürülen ana kitle daęılımını destekleyip, desteklemediğini arařtırmaktır. Chi Kare daęılımını bu amaçla uygulayabilmek için, kritik deęerler řu formülle bulunur:

$$\chi^2 = \sum_{i=1}^{\infty} \frac{(O_i - E_i)^2}{E_i}$$

Burada,

O_i = Örnek değişkenin gözlenen frekansı,

E_i = hipotezde ileri sürülen ana kitle dağılımının beklenen frekansıdır.

Bu testin uygulanmasından önce, bir sıfır hipotez belirlenir. Yani, verilerin nasıl bir dağılıma sahip olduğu belirtilir. Daha sonra bu sıfır hipotezde belirtilen olasılık dağılımına göre, beklenen frekanslar belirlenir. Bu beklenen değerler, gözlenen değerlerle (O) karşılaştırılır. Eğer, beklenen değerle, gözlenen değerler arasında uyum varsa, hipotez kabul edilir; yoksa reddedilir⁶⁷.

- **Kolmogorov Smirnov Testi**

Kolmogorov Smirnov testi adını, gözlenen frekans dağılımı ile kuramsal frekans dağılımı arasında önemli bir fark olup olmadığını test etmek için basit, parametrik olmayan bir yöntem geliştiren iki istatistikçiden almıştır. Bu nedenle Kolmogorov Smirnov testi χ^2 testi gibi kuramsal frekans dağılımının iyi uyumunu saptayan diğer bir ölçüdür. Ancak Kolmogorov Smirnov testinin χ^2 testine göre birkaç avantajı daha vardır. Çok güçlü bir test olmasının yanı sıra kullanımı da kolaydır.

Kolmogorov Smirnov istatistiği D_n , özellikle gözlenen frekans dağılımının beklenen frekans dağılımına ne kadar yakın olduğunu, verilerin gruplandırılmasına ihtiyaç olmadan yargılamak için kullanışlı bir ölçüttür. Çünkü D_n ' in olasılık dağılımı, n hacmine bağlıdır. Ancak beklenen frekans dağılımdan bağımsızdır. D_n bir dağılım serbestliği istatistiğidir⁶⁸.

⁶⁷ M. ÇAKICI, A. OĞUZHAN., T. ÖZDİL, **Temel İstatistik**, Özal Matbaası, İstanbul 2000, s.254-255

⁶⁸ Hülya TÜTEK, Şevkinaz GÜMÜŞOĞLU, **İşletme İstatistiği**, Barış Kitabevi, İzmir 2000, s.235

2.6 SİMÜLASYONDA MODELLEME YAKLAŞIMLARI

Bir sistem iki yönden incelenebilir. İlk yaklaşım, gerçek bir sistem üzerinde bir takım deneyler gerçekleştirmektir. Ancak, iki olası nedenden dolayı bu mümkün olmayabilir: İlk neden, sistemin tasarım aşamasında bu tür deneyler gerçek sistemin davranışlarını yansıtmayabilir. İkinci neden de, bu tür deneyler oldukça zaman alıcı ve yüksek maliyetli olabilir. Örneğin imalat sistemleri mevcut üretim kaynaklarına alternatif işlevsel stratejilerin performanslarını kıyaslamak için durdurulamaz.

İkinci yaklaşım gerçek sistemin bir modeli üzerinde deneyler yapmaktır. Modellemenin amacı, sistemin parçaları arasındaki ilişkiyi temsil etmesidir. Daha az zaman ve para gerektirmesi ve aynı zamanda gerçek sistem davranışını bozmaması gibi avantajlar, imalat sistemlerinin analizinde daha geniş ölçüde modelleme yaklaşımının kullanılmasına yol açmıştır. Gerçek sistemin modeli iki tipte olabilir. Bir tanesi, gerçek sistemin fiziksel bir modelidir. Bu tür modeller simgesel olarak nitelendirilebilir ve genellikle eğitim amaçlıdır. İkincisi ise, gerçek bir sistemin matematiksel modelidir. Matematiksel bir model, gerçek sistemdeki mantıksal ve sayısal ilişkileri temsil eder. Geçerli bir matematiksel model kurulduktan sonra, mantıksal ve sayısal değişimlere modelin vereceği yanıtları görmek ve bu değişimleri yönetmek mümkündür⁶⁹.

Model, analitik yöntemler ve simülasyon yardımıyla incelenebilir. Seçim modelin karmaşıklığına bağlıdır. Model çok karmaşık değilse, analitik bir yöntem seçilebilir. Analitik modeller, optimum sonucu vermesine rağmen, sıklıkla basitleştirme varsayımları altında kullanılırlar. Eğer gelişmiş imalat sistemlerinin modellemesi gibi karmaşık bir model söz konusu ise simülasyona ihtiyaç duyulur. Çünkü, gelişmiş imalat sistemlerinin parçaları arasındaki karmaşık etkileşimlerin analitik bir sonuca ulaşmak için, eşitlikler yardımıyla temsil edilmesi son derece güçtür. Simülasyon, optimum sonucu sağlamaz. Başarılı

⁶⁹ Riitta SMEDS, Jens O. RIIS, **Learning Production Management The Effects of Using Simulation Games in Universities and Industry**, Chapman & Hall, Londra, 1998, s. 105-110

bir simülasyon analizi, istatistik ve bilgisayar simülasyon dilleri uzmanının birlikte çalışmasını gerektirir.

Etkili bir simülasyon çalışması yapabilmek için, potansiyel problemleri olan sistem varlıklarının incelenmesi ve analizin buna göre hazırlanması gerekir. İyi bir model gelecek ihtiyaçları da göz önüne alarak, sistemin diğer parçalarını da içine alacak şekilde tasarlanmış olmalıdır⁷⁰. Üzerinde çalışılacak problemin kesin tanımının yapılabilmesi beklenenden zor olabilir. Çalışmanın sonucu ile ilgilenen, sadece modeli oluşturan kişi değildir. Mühendisler, yöneticiler, operatörler ve bir çok çalışanın, oluşturulan çalışmadan değişik beklentileri vardır. Çalışmanın yapısı ve içeriği hakkında, genel bir tanım oluşturmak, bu kişilerden gelecek verilerin ve gerekli desteğin daha kolay elde edilmesini sağlayacaktır. Simülasyon çalışmasının amaçları, genellikle ele alınan problem tarafından belirlenir. Çünkü model kurulduktan sonra, model üzerinde yapılacak çalışmaların problemi çözmesi hedeflenmektedir. Potansiyel sistem iyileştirme imkanlarının ve yöntemlerinin değerlendirilmesinin, çalışma hedeflerinin belirlenmesinde rolü büyüktür.

Sistemin durum değişkenlerine bağlı olarak, kesikli ya da sürekli bir simülasyon modelinin, uygun bir bilgisayar dilinde düzenlenmesi gerekmektedir. Genel amaçlı bilgisayar dilleri (C, PASCAL, FORTRAN; vs.) ve simülasyon dilleri (SIMAN, ARENA SE, SIMFACTORY, GPSS, PROMODEL, vs.) bir sistemi temsil eden modeli kodlamak için kullanılabilir. Kesikli sistemlerle ilgili pek çok işletme problemi olduğundan, bu sistemleri modellemek için özel karakteristiklere sahip, çok sayıda simülasyon dili bulunmaktadır.

Sistemleri simüle etmek için, üç simülasyon yaklaşımı kullanılmaktadır. Bunlar, olayların listesini yapma, faaliyet tarama ve süreç etkileşimidir. Bu yaklaşımların her birinde simülasyon saatini ileri götüren olay kavramı kullanılmaktadır. Bir

⁷⁰ C.D. PEGDEN, R.E. SHANNON, R.P. SADOWSKI, **Introduction to Simulation Using SIMAN**, 1990, s.6-8

sonraki olay süreçleme için seçildiği zaman, modelde kullanılan değişkenler, kurulan modelin mantığına uygun olarak güncellenir.⁷¹

Olayların listesini yapma yaklaşımı, zaman içinde bağımsız olayların birbiri ardına olmasıyla ilgilidir. Bir olay, kurulan simülasyon modeli değişkenleri üzerine etki eden ve simülasyon saatini ileri götüren işlemdir. Modeli yapan kişi sistemde meydana gelebilecek her olayı açıklayarak, her olayın sebep ve sonuçlarını belirterek, olay değişimini yöneten mekanizmaları yaratarak, olayları bir diğerine mantıklı bir şekilde bağlayarak her olaydaki istatistikler ve zamanı güncelleyerek ve ilgili istatistikleri toplayarak bir simülasyon modeli kurmaktadır. Zaman kontrol prosedürü, modelde tanımlanan sürece bağlı olarak en başta olan olayı olay listesinden seçer, simülasyon saatini güncelleştirir ve karşılık gelen olay rutinine neden olur. Olay rutinlerinin içerisinde simülasyon saatinin yanında sistemin analizi için gerekli performans değerleri de bulunur. Bu süreç, daha önce tanımlanan simülasyon saati zamanına ulaştığında ya da süreç boyunca oluşan varlıkların sayısı hedeflenen miktara eşitlendiğinde son olay rutini yerine getirilir. Olayların listesini yapma yaklaşımı her hangi bir genel amaçlı bilgisayar diliyle simule edilen modeli kodlamak için direkt olarak kullanılmaktadır. Oysa simülasyon dillerini kullanma durumunda, olayların listesini yapma yaklaşımı, simülasyon dili mantığının temelini oluşturmaktadır ve yaklaşım, genel amaçlı bilgisayar dilinde yazılmış olan bir olay rutini olarak dolaylı bir şekilde kullanılmaktadır.

Faaliyet tarama yaklaşımı, gerek genel amaçlı programlama dilinde gerekse simülasyon dillerinde bir sonraki olay, olay rutininden seçilerek olaya ait zaman ve durum kaydedilir. Dönem faaliyeti, zaman gerektiren bir sistem durum geçişi olarak açıklanmaktadır. Bazı modellerde, olması beklenen olaylar kesin bir zamana göre programlanamaz. Bir modeldeki her faaliyet rutini, model belli bir duruma sahip olduğu için olması gereken tüm varlık hareketlerini açıklamaktadır. Bu prosedüre, faaliyet tarama denir. Faaliyet tarama yaklaşımında, zaman kontrol mekanizmasını harekete geçirebilmek için

⁷¹ age, s.24

faaliyetler, öncelik sırasına göre incelenmektedir. Daha sonra, mekanizmanın hareketi için gerekli ilk faaliyet rutini yerine getirilir. Araştırma mekanizması öncelik sırasına göre bir daha başlar ve süreç son zamana kadar devam eder.

Süreç etkileşimi yaklaşımı, gerçek hayatı eş zamanlı süreçlerin birbiriyle etkileşimine göre, birbirine rakip veya işbirliği halinde modellenen bir varlıklar bütünü olarak göz önüne almaktadır. Olayların listesini düzenleme yaklaşımındaki gibi, süreç etkileşimi yaklaşımı da, ardışık zamanlı olayları bir simülasyon modeli olarak değerlendirmeyi sağlar. Bu iki simülasyon yaklaşımı çok benzerdir. Bunlar temel olarak dil yapılarında farklılık göstermektedir (Örneğin; SEIZE, RELEASE, QUEUE vs.). Süreç etkileşim durumları, daha anlaşılabilir ve bir simülasyon modelinde yaygın bir şekilde meydana gelen durumları önceden tanımlanmış bloklar şeklinde ifade eder⁷². Süreç etkileşim yaklaşımı seçildiği zaman gerçekte uygulanan simülasyon yaklaşımı olayların listesi yaklaşımıdır. Simülasyon modelini süreç etkileşimine göre düzenlemek genellikle özel amaçlı simülasyon dillerinin kullanımını gerektirmektedir⁷³.

2.7 SİMÜLASYONUN AŞAMALARI

Bir simülasyon çalışması;

- Problemi tanımlama,
- Girdilerin toplanması ve istatistiksel analizi,
- Yazılımı seçme,
- Model geliştirme,
- Modelin doğruluğunu kontrol etme,
- Modelin geçerliliğini test etme,
- Deneylerin Tasarımı,
- Çıktıların istatistiksel analizi,

⁷² M. LAW, D. KELTON, **Simulation Modeling and Analysis**, Mc Graw Hill, New York, 1991, s.237-240

⁷³ N. COLLIS, S.M. WATSON, Introduction to Arena, **Proceedings of the 1993 Winter Simulation Conferences**, s. 45-48

adımlarından oluşmaktadır. Bu adımları başarılı bir şekilde yerine getirmek, modeli kuran kişinin istatistik ve bilgisayarlar konusunda geçmiş deneyimlere sahip olmasını gerektirir. Simülasyon yaklaşımının sadece yazılım seçme ve kodlamayla ilgili olduğu yaygın bir yanlış algılamadır. Eğer yukarıda belirtilen adımlar simülasyon analistleri tarafından takip edilmezse, belirtilen bir problemle ilgili mantıklı çözümler elde etmek imkansızdır⁷⁴.

2.7.1 Problemi Tanımlama

Modeli, gerçek sistemi tam olarak temsil edebilecek şekilde tasarlamak zordur. Bu sebeple problemi tanımlamanın dikkatle yapılması önemlidir. Modelin cevaplayacağı belirli sorulardan oluşan bir liste oluşturmak, bir simülasyon çalışmasına başlamak için iyi bir yoldur⁷⁵.

2.7.2 Girdilerin Toplanması ve İstatistiksel Analizi

Çözülecek problemin tanımını doğru olarak yapıp, kurulacak modelin cevaplayacağı soruları belirledikten sonra, simüle edilecek sistemle ilgili girdi verilerin elde edilmesi ve bunların istatistiksel analizi gerekir. Sistemle ilgili bilgi, sistemi gözlemleyerek, sistemin tasarımından sorumlu kişilerle görüşerek, operatör veya yöneticilerle konuşarak ve simüle edilecek sistemin varsa CAD çizimlerini alarak, yoksa bu tasarımları hazırlayarak elde edilebilir.

Sistemle ilgili belirli bilgileri edindikten sonra, sistem parçalarının ve sistemin işlevsel özelliklerinin listelenmesi gerekir. Sistem parçaları, çoğunlukla makineler, işçiler, malzeme ve yarı mamuller, malzeme taşıma sistemleri, ve robotlardır. İşlevsel özellikler, programlama kuralları, süreçleme mantığını içerir. Bu listeler işlemlerin bir tanımı olarak özetlenebilir. Simülasyon

⁷⁴M. LAW, M.G. MCCOMAS, How to Select simulation software for Manufacturing Applications, **Industrial Engineering**, 1992 July, s. 29-34

⁷⁵ J. BANKS, R.R. GIBSON, Getting Started in Simulation Modeling, **IEE Solutions**, November 1996, s.34-39

modellerinin esasları bu adımda kurulduğu için, işlevsel özelliklerin listelenmesi simülasyon çalışmasındaki en önemli faaliyetlerden biridir.

Modelin kurulmasında, kuyruk disiplini, işçi ve makine kapasiteleri, ara stok değerleri gibi temel verilere ek olarak, simülasyon analisti gelişler arası zaman, işlem zamanı, bozulmalar arasındaki zaman, onarım süresi, konveyör hızı gibi çeşitli girdi verilerine ihtiyaç duyar. Bu girdi verileri deterministik ya da stokastik olabilir. Örneğin; manuel işlemlerin yapılma zamanı ve gelişler arası zaman olasılıklara bağlıdır ve genellikle rastgele sayı üretici cihazların kullanımına ihtiyaç duyarlar. Bu aşamada simülasyon analistinin hangi verilere ihtiyaç duyulduğu, hangi verilerin mevcut olduğu, verilerin uygun olup olmadığı, mevcut verilerin beklenen amaç için geçerli olup olmadığı ve verileri nasıl toplayacağına karar vermesi gerekir. Çoğu simülasyon çalışmasında, hangi verilerin kullanılacağına karar verme çok zor ve zaman harcanılan bir görevdir. Girdi verilerini belirlemede izlenen yöntemler sistemdeki sorunun varlığına bağlıdır⁷⁶.

Eğer sistem mevcut ise, gerçek veriler mevcut olan ya da toplanmış geçmiş verilerden elde edilebilir. Bu durumda, toplanmış geçmiş veriler simülasyon girdi modellerini elde etmek için kullanılabilir. Mevcut veriler toplanırken, iki metot uygulanabilir. Bunlar:

- Doğrudan ampirik dağılımdan örnekleme: Ampirik dağılımdan örnekleme, gelecekteki, işi aynı geçmişteki gibi yapmaktır, yani eğer olaylar geçmişte yer almadıysa bunların gelecekte de meydana gelemeyeceğini varsaymaktadır. Eğer ampirik verilerin kullanılması gerekiyorsa, bunlar, bir kümülatif olasılık dağılımı şeklinde olmalıdır. Gözlenen değerler; (ya da mevcut olan değerleri) artan sırada düzenleyerek, gruplayarak, nispi frekanslarını belirleyerek ve daha sonra kümülatif olasılık dağılımlarını hesaplayarak ampirik bir kümülatif dağılım şekline getirilebilir.

⁷⁶ Alan PRITSKER, Jean O. REILLY, **Simulation with Visual SLAM and AWESIM**, John Wiley sons, Newyork 1995, s.11-12

- Teorik dağılımdan örnek alma: Veriler teorik bir dağılıma uyuyorsa teorik dağılımdan örneklemedir. Eğer teorik veriler kullanılacaksa, üç adımın izlenmesi gerekir:
 - İlk adım histogramın şekline göre, ampirik verileri temsil edebilen bir dağılım seçmektir.
 - İkinci adım, gözlenen verileri kullanarak seçilmiş olan dağılımın parametrelerini tahmin etmektir.
 - Son adım ise, Chi Kare ve Kolmogorov-Smirnov gibi uygun yöntemler kullanarak uygun nispi dağılım eğrisini tespit etmektir. Simülasyon dillerinin çoğunun, bu görevleri otomatik olarak yerine getiren belirli modülleri vardır.

Eğer sistem henüz mevcut değilse, geçmişle ilgili hiçbir veri yoktur ve hiçbir veri toplama şansı da bulunmamaktadır. Bu durumda (örneğin; tasarım aşamasında olan sistemden) veri elde etmek imkansızdır. Bu nedenle, diğer girdi veri kaynakları kullanılmaktadır. Bunlar, benzer sistemlerden operatör tahminleri veya satış elemanlarının verileri olabilir. Hiçbir sistem aynı karakteristikleri gösteremeyeceği için benzer bir sistemden veriler kullanmak riskli olabilir. Örneğin, bir üretim sisteminde, operatör tahminleri geniş bir aralığa sahiptir; zira insanların tahmin yeteneği genelde zayıftır. Satış elemanlarının tahminleri ise, son derece iyimser olabilmektedir⁷⁷.

2.7.3 Yazılımı Seçme

Bir sistemi simüle etmek için, üç alternatif yazılım aracı vardır. Bunlar, genel amaçlı bilgisayar dilleri, simülasyon dilleri ve simülatörlerdir. C, FORTRAN, PASCAL gibi genel bilgisayar dilleri yeterince esnek olmasına rağmen, sık sık iki sistem parçası arasındaki karmaşık etkileşimi simüle etmek için, programlama uzmanına gereksinim duyarlar. Eğer simüle edilecek sistemin karmaşıklığı artarsa, simülasyon modelini düzenlemek için gerekli ön hazırlık

⁷⁷ J. BANKS, Selecting Simulation Software, *Proceedings of the 1991 Winter Simulation Conference*, s. 37-38

zamanı da artar. Simülasyon dilleri ve simülatörler en yaygın kullanılan simülasyon araçlarıdır. Bir simülasyon dili, genel olarak bir bilgisayar paketidir, fakat iş istasyonları ve malzeme taşıma modülleri gibi bir takım imalat tesislerini modellemek için özel karakteristiklere sahip olabilir. GPSS, SIMAN, ARENA SE VE SLAMSYSTEM birkaç iyi bilinen simülasyon dilidir. Bir imalat simülatörü, az bir programlamayla ya da bazı vakalarda hiçbir programlama yapılmaksızın bir sistemi simule eden bir bilgisayar paketidir⁷⁸. Simülatörlerden bazıları ise, PROMODEL, SIMFACTORY, WITNESS'dir⁷⁹. Bir simülasyon aracını seçmek çok zor bir iştir. Bir simülasyon projesinin başarısı, bu seçimden direkt olarak etkilenmektedir.

Simülasyon aracının seçiminden sonra simülasyon modeli, akış şemaları ve algoritma şemaları yoluyla tanımlanan kontrol mantığına dayanılarak geliştirilir.

2.7.4 Simülasyon Modelini Geliştirme

Daha önce de söz edildiği gibi, simülasyon modelini geliştirme, gerçek sistem davranışını bozmadan bir bilgisayar tarafından gerçek sistem parçaları arasındaki ilişkileri yansıtan bir yaklaşım olarak tanımlanmaktadır. Bir simülasyon modeli kurma birkaç adım gerektirmektedir. Bunlar, aşağıdaki gibi özetlenebilir⁸⁰:

- Alternatif politikaların performansını kıyaslamak için, simülasyon analistine yardım eden performans ölçümlerini seçme.
- Seçilen performans ölçümlerini, direkt olarak etkileyen işlevsel karakteristikler ve sistem parçalarını belirtme (işlemlerin tanımı için varsayımlar-limitler listesi yapma).
- İşlemlerin tanımına göre, kontrol mantığı ve varlık akışını temsil eden algoritma şemaları ve akış şemaları çizme.

⁷⁸ J. Banks, R.R. GIBSON, **Simulation Modelling**, February 1997, s.26-31

⁷⁹ M. LAW, M.G. MCCOMAS, How to Select simulation software for Manufacturing Applications, **Industrial Eng.** 1992 July, s. 45-52

⁸⁰ Michael BACKER, Harald SCHALLNER, **Model and Simulation based Flexible Production Management System, Computer Applications in Production and Engineering**, Chapman & Hall, 1997, s.436-442

- Sistem parçaları arasındaki bu matematiksel ve mantıksal ilişkileri düzenlemek için, bir simülasyon aracı seçme (genel amaçlı bilgisayar programlama dilleri, özel amaçlı simülasyon dilleri ya da simülatörler).
- Seçilen simülasyon aracıyla modeli düzenleme.

Bir karar ya da bir tasarım değişikliğinin, bir sistemin davranışı üzerindeki etkisi, ya gözlem yoluyla ölçülebilir ya da simülasyon gibi yöntemler kullanarak tahmin edilebilir. Bu genellikle performans değerlendirme olarak adlandırılır. Simülasyon, bir performans değerlendirme tekniği olduğundan, alternatif karar verme politikalarının performansını kıyaslamakta kullanılacak kriteri seçmek gerekmektedir. Bir imalat ortamındaki tipik performans kriterleri; çıktı oranları, ortalama envanter düzeyleri, müşteri hizmet düzeyleri, ortalama akış süreleri ve makine duruşlarıdır. Simülasyon çalışmalarında çözülecek probleme göre, bu performans kriterlerinin seçilmesi gerekmektedir.

Performans kriterlerinin seçiminden sonra, modelin geliştirileceği varsayımlarla ilgili bir karar verilir. Varsayımlar, uygun sistem sınırlarıyla ilgili karar vermekle ilgilidir. Eğer bu sınırlama çok genişse, simülasyon projesini tamamlamak için ilave zaman ve masrafla yüz yüze gelinecektir. Örneğin, A ve B şeklinde iki alt departmanı olan bir imalat tesisinin olduğunu varsayalım. (Örneğin; parçalar önce A, daha sonra B departmanında işlenmektedir). Eğer mevcut olan problem sadece B departmanını ilgilendiriyorsa, o zaman A departmanını simüle etmeye gerek yoktur. Diğer taraftan eğer sınırlama çok dar ise, bir simülasyon çalışmasıyla cevaplanması gereken sorular tam olarak cevaplanamaz. Bu sebeple, kabul edilen varsayımlar simülasyon modelinin karmaşıklığını etkiler. Varsayımların mantık ve geçerliliği doğru sonuçlara ulaşmak için ispat edilebilir olmalıdır.

2.7.5 Simülasyon Modelini Doğrulama (Verification)

Simülasyon modelini geliştirdikten sonra, modeli doğrulama gerekmektedir. Simülasyon çalışmalarında, doğrulama, bilgisayar programının beklendiği ve

amaçlandığı gibi işlediğini göstermek ve böylece bir simülasyon modelinin doğru bir mantıksal ve matematiksel sunumu temsil etmesini sağlamayı içermektedir. Doğrulama, aynı zamanda hata ayıklama olarak da adlandırılmaktadır. Programlama hatalarının basit ve sade bir şekilde elenmesine ek olarak, doğrulama süreci ayrıca kodlamanın modeli tamamen yansıtıp yansıtmadığını görmeye de yardımcı olmaktadır. Doğrulama hedeflerinden biri de, modellemedeki tüm parçaların hem bağımsız, hem de birlikte çalıştığını göstermek ve doğru verileri doğru zamanda kullanmaktır⁸¹. Bir simülasyon modelini doğrulama için izlenen yol aşağıda verilmektedir:

- Programı modüler bir tarzda geliştirme.
- Simülasyon çıktılarının mantıklılığını kontrol etme.
- Programda hata ayıklama (debugging).
- Modeli kontrol edilebilir değişkenlere göre çalıştırma

Simülasyon modelini, modüler bir tarzda geliştirme, kodlama ve hata ayıklamayı kolaylaştırır. Tüm modelde hata ayıklamak zordur. Alt program ve alt modüllerin bağımsız bir şekilde kontrol edilmesi gerekir. Modüler sistemde gerektiğinde yeni bir modül kolaylıkla eklenebilir. Simülasyon modelini adım adım geliştirmek, program hatalarının oluşumunu azaltır⁸².

Büyük simülasyon modelleri geliştirilirken, bilgisayar programının kontrol mantığını birden fazla kişinin kontrol etmesi önerilmektedir. Programı yazan kişi alışkanlıklarına göre davranabileceğinden dolayı iyi bir eleştirmen olmayabilir. Bazı işletmelerde, bu fikir resmi olarak yerine getirilir ve yapılandırılmış yürüyüş (structured-walk through) olarak adlandırılır. Bu fikir, simülasyon modelini oluşturan bilgisayar programının kontrolünün, daha önceden planlanmış bir süreç çerçevesinde eş zamanlı olarak yapılmasını ifade eder.

⁸¹ R. SHANNON, Introduction to Simulation, **Proceedings of the 1992 Winter Simulation Conference**

⁸² J. KLEIJNEN, Theory and Methodology-Verification and Validation of the Simulation Models, **European Journal of Operational Research**, 1995, Vol. 82, s. 145-162

Çıktı verilerinin olurluluğunu kontrol etmek, mantıksal ya da matematiksel hataları ortaya çıkarmak için iyi bir yoldur. Pilot çalışma, model geliştirildikten sonra yapılabilir. Pilot deneme süresince model çeşitli girdi parametreleriyle çalıştırılır ve daha sonra bunun olurlu olup olmadığını görmek için çıktı kontrol edilir.

Kesikli bir olay simülasyon programında, hata ayıklamak için kullanılabilecek en güçlü tekniklerden biri programın kontrol mantığını izlemektir. İzleme süreci boyunca, simule edilmiş sistemin durumu, olay listesi, durum değişkenliği ve istatistiksel sayaçlar vb., her olayın oluşumundan hemen sonra bir dosyaya kaydedilir. Daha sonra, bu liste hataları ve uygun olmayan durumları göstermesi için kontrol edilebilir. Etkileşimli bir hata ayıklayıcı, bir analizcinin zaman içerisinde seçilmiş bir noktada simülasyonu durdurması ve belli değişkenlerin değerlerini incelemesine olanak sağlar.

Eğer belli bir karakteristik biliniyor ya da bir girdi verisi için hesaplanabiliyorsa, bu, simülasyon analistinin modeldeki hataları analiz etmesine yol gösterebilir. Örneğin; modelin başladığımız şekilde davranıp davranmadığını görmek için deterministik sayılar kullanılabilir.

Doğrulama amacıyla, bazen animasyon kullanılabilir. Öte yandan doğrulama için, animasyon kullanmak çok güvenilir bir yol değildir. Bir modeli doğrulama için animasyon kullanıldığı zaman, bunun amaçlanan sistemi tam bir şekilde temsil etmesi gerekir⁸³.

⁸³ M.ROHRER, Visualization And Its Importance in Manufacturing Simulation, **Industrial Management** (May / June), 1996

2.7.5.1 Doğruluğu Sınama Teknikleri

Bu teknikler, objektif ya da sübjektif olarak kullanılabilirler. Burada istatistiksel test ya da matematiksel prosedürlerin kullanımı söz konusudur⁸⁴.

- *Animasyon* : Modelin işlevsel davranışı, zamana bağlı olarak grafiksel olarak ifade edilir.
- *Diğer modellerle karşılaştırma*: Geçerliliği kabul edilmiş diğer simülasyon modellerinin, çıktı değerleri ile karşılaştırılması yapılır.
- *Dejenerasyon*: Modelin davranışının dejenerasyonu, girdi ve içsel parametrelerin uygun değerlerinin seçimleri ile test edilir. Örneğin geliş oranının servis oranından büyük olduğu durumlarda kuyrukta ortalama bekleyen varlık sayısı artar.
- *Olay geçerliliğinin sınanması*: Gerçek sistem ile simüle edilen sistemde meydana gelen olaylar karşılaştırılarak, benzer olup olmadıklarına bakılır.
- *Aşırı durum testleri*: Model yapısı ve çıktı değerleri, sistemdeki faktörlerin seviyelerindeki kombinasyonlara bağlı olarak, herhangi bir aşırı ya da beklenilmeyen durumda, modeli karşılar nitelikte olmalıdır. Yani çıktı değerlerinin, modelden sonuç alacak nitelikte tasarlanmış olup olmadığını test eder.
- *Sabit değerler*: Sabit değerler (sabitler) çeşitli model girdi parametreleri için kullanılırlar. Bu değerler model sonuçlarının kontrol edilmesine olanak sağlar.
- *Eskiye dayanan verinin geçerliliğinin sınanması*: Eğer eskiye dayanan veri mevcutsa (sisteme ait veri model geliştirme ya da test etme amacıyla toplandıysa), bu verilerin bir kısmı modeli kurmak için kullanılır, geri kalan kısmı ile de modelin geçerliliği test edilir.
- *Operasyonel grafikler*: Çeşitli performans değeri zamana bağlı grafikler yardımıyla ifade edilebilir.

⁸⁴ Jerry BANKS, John S. CARSON, Barry NELSON, David NICOL, **Discrete Event System Simulation**, Prentice Hall, New Jersey 2000, s.367-385

- *Duyarlılık Analizi:* Bu teknikte, girdi değerlerinin değiştirilmesi sonucu modelin davranışı ile çıktı değerleri üzerindeki etkisi incelenir. Gerçek sistemde olduğu gibi, girdi ya da içsel parametre değişikliği simüle edilen sistemde de aynı etkileri göstermelidir. Duyarlılığa sahip parametreler, modelin davranışı ya da çıktı değerleri üzerinde önemli değişimlere neden olurlar.
- *Tahminleyici geçerlilik:* Bu model, sistem davranışını tahminlemeye çalışır ve sistem davranışı ile tahminlenen model arasında aynı değerlere sahip olup olmadıkları konusunda karşılaştırmalar yapar.
- *Takip:* Model içerisindeki belirgin varlıkların davranışı, modelin kurulum mantığının doğru olup olmadığı konusunda karara varmak için izlenir.

Bir simülasyon çalışmasındaki bir sonraki adım ise, modelin geçerliliğini test etmedir. Geçerliliği test etme, simülasyon projelerinde en önemli adımlardan biridir. Bu adım, geliştirilen modelin gerçek sistemi tam ve doğru bir şekilde temsil edip etmediğini görmeye yardımcı olur.

2.7.6 Simülasyon Modelinin Geçerliliğini Test Etme (Validation)

Geçerliliği test etme adımının, yukarıda sözü edilen doğrulama adımıyla karıştırılmaması gerekmektedir. Doğrulama, modelin geçerli olmasını garanti etmez. Simülasyon kodlamasının doğruluğunun tespitinden sonra, programın kontrol mantığı, bunun gerçek sistemi tam olarak temsil etmesini garanti etmesi için, detaylı bir şekilde incelenir. Geçerlilik, tamamlanılan modelin karar vermede kullanılabileceğini ortaya çıkarmak için tanımlanabilir. Geçerlilik ile ilgili bir süreç üç temel soruyla ilişkilidir ⁸⁵:

- Model gerçek sistemi uygun bir şekilde temsil ediyor mu?
- Simülasyon modeliyle üretilen çıktı verileri gerçek sistemin işlem karakteristikleriyle benzer mi?
- Simülasyon modelini kullanan kişi modelin sonuçlarına güveniyor mu?

⁸⁵ age., s.71

Modelin davranışının geçerliliği, gerçek sistemle karşılaştırılabilirliği ile ilgilidir. Geçerlilik, modelin kendi içerisinde tutarlı olup olmadığı ile ilgili bir karar değişkeni değildir. Model, birçok testten geçerken ve model ile gerçeklik arasındaki uygunluğun yeni noktaları bulunduğu anda, modelin kullanılışlığına olan güven aşama aşama artmalıdır. Geçerliliğin testi sürecinde, modeli tasarlama, kurma ve kullanma sürecinde sürekli iyileştirmeler meydana gelir. Eğer model geçerli değilse, o zaman modelden türeyen her hangi bir sonuç güvenilir olmaz.

En önemli geçerliliği test süreçlerinden biri, üç adımdan oluşan bir yaklaşımdır. İlk adım, daha modeli geliştirme aşamasında sistem uzmanlarıyla konuşmalar, sistem gözlemlenmeleri, mevcut teoriler ile kıyaslamak için benzer simülasyon modellerinden elde edilen yeterli miktarda sonuçları ve kendi tecrübeleri de dahil tüm mevcut bilgilerin bir araya getirilmesidir⁸⁶.

İkinci adım ise, modelin varsayımlarını ampirik olarak test etmektedir. Yapılan varsayımların mantıklı ve gerçek sistemi iyi bir şekilde temsil etmesini sağlayan pek çok kantitatif test mevcuttur. Örneğin, eğer gözlemlenmiş olan verilere teorik dağılım uydurulur ve simülasyon modeline girdi olarak kullanılırsa, bu denemenin uygunluğu grafiksel çizimler, Chi Kare ve Kolmogorov Smirnov gibi testlerle test edilebilir. Gözlenen verilerin homojenliği Kruskal-Wallis testi ile test edilebilir. Örneğin; bir gün süresinde gelişler arası zamanlar, farklı teorik dağılımlara uyabilir. Bu durumda gelişler arası zamanlarını temsil etmesi için bir teorik dağılımı kullanmak güvenli sonuçlar vermeyebilir. Geçerliliğin ikinci adımı süresince en faydalı olan araçlarından biri duyarlık analizidir. Bu, simülasyon çıktısı üzerinde en önemli etkiye sahip olan girdi parametresini tespit etmek için kullanılabilir. Bu girdi parametresinin çok dikkatli bir şekilde analiz edilmesi gerekir.

Üç adımlı yaklaşımdaki üçüncü adım ise, simülasyon çıktı verilerinin ne derece temsil özelliğine sahip olduğuna karar vermektir. Simülasyon çıktısının, mevcut

⁸⁶ Law age., 1991, s.307-322

sistemin çıktısını, gerçekten temsil etmesini sağlayan uygun bir çok istatistik testi bulunmaktadır. Diğer yandan, bu kıyaslama görüldüğü kadar kolay değildir. Hemen hemen bütün gerçek sistemler ve simülasyon çıktısı, sabit olmadıkları için (gözlemlerin dağılımları zaman içerisinde değişir), oto korelasyon ve klasik istatistik testleri doğrudan uygulanamaz. Geçerlilik için uygulanan metotlardan en fazla kullanılanları aşağıdaki gibidir:

- *Denetleme Yaklaşımı*: Denetleme yaklaşımı, formel istatistik prosedürlerini kullanmaksızın modelin ve gerçek sistemin bir ya da daha çok istatistiğin karşılaştırmasıyla ilgilidir. Bu istatistikler basit ortalama, varyans vb. olabilir. Bu yaklaşımda hem model hem de gerçek sistemden alınan gözlemlerin sayısı aynı olmalıdır. Bu yaklaşımın dezavantajı formel istatistik prosedürü olmadan elde edilen sadece bir ya da iki istatistiğin, modelin gerçek sistemi iyi bir şekilde temsil ettiği sonucunu çıkarmak için yeterli olmamasıdır.
- *Bağlı Denetleme Yaklaşımı*: Bu yaklaşım, daha mantıklı karşılaştırma kriteri elde etmek için hem gerçek sistem hem de model için aynı geçmiş girdi verilerini kullanmaktadır.
- *Bağımsız Verilere Dayanan Güven Aralığı Yaklaşımı*: Bu yaklaşım, model tarafından üretilen çıktı verilerinin gerçek sistemden toplanan çıktı verilerinden çok farklılık gösterip göstermediğini test eder. Bu yaklaşımda, gerçek sistem ve modeli kıyaslamak için pairet-t testi ve hipotez testi kullanılır. Bu yaklaşım, model ve sistem çıktısını karşılaştırmak için ilk iki yaklaşıma oranla daha güvenilir bir yoldur.

2.7.7 Simülasyon Deneylerinin Tasarımı

Modelin gerçekleşmesinden sonra, deney koşullarının hazırlanması gerekmektedir. Kısacası, istatistiksel metotlar tarafından kullanılacak gerekli verilerin toplanabilmesi ve analiz edilmesi için deneylerin tasarımı, bir deneyi planlamada kullanılan bir metottur. Faktörlerin farklı düzeyleri için, simülasyon modeli çalıştırılır ve sonuçları varyans analizine bağlı olarak değerlendirilir.

Simülasyon deneyleri yapıldığı ve çıktılar alındığı zaman, faktörlerin başlıca etkileri ve etkileşimleri de belirlenebilir. Simülasyon analisti daha sonra istenilen yanıtı ulaşmak için faktörlerin düzeylerini belirleyebilir⁸⁷. Deneylerin tasarımı, bir simülasyon çalışmasının çok önemli bir aşamasıdır.

Deneyel tasarım, simülasyon analistinin, istenilen bilginin en az çaba ortaya çıkarılabilmesi için, hangi konfigürasyonların simüle edileceğine karar vermesine yardım eder. Birden fazla faktör içeren bir modelde deneylerin tasarımı şöyledir:

1- Faktörlerin düzeyleri belirlenmelidir.

2- Söz konusu faktörlerin farklı düzeyleri arasında, anlamlı bir farklılık olup olmadığı tespit edilmelidir.

3- Faktöriyel bir tasarımda, sistem üzerindeki bu faktörlerin gerçek etkileri incelenebilir. Bu tasarımlarda faktör düzeylerinin tüm kombinasyonları da incelenmektedir. Bu, "tam faktöriyel tasarım" olarak adlandırılmaktadır. Eğer faktörlerin sayısı artarsa, tam faktöriyel tasarım bir çok deney ve uzun hesaplama süresine gereksinim duyar. Bu sebeple, bu durumlarda kesirli faktöriyel tasarım kullanılabilir⁸⁸.

4- Varyans analizine bağlı olarak F-testi ile belirlenen düzeyde simülasyon modeline etkisi fazla olan faktörler etkileşimleri ile göz önüne alınır. Etkisi az ya da önemsenmeyecek seviyede olan faktörler ve etkileşimleri de modelden çıkarılır.

2.7.8 Çıktıların İstatistiksel Analizi

Bir simülasyon modelinin çıktılarının analizinde kullanılan yaklaşım, sistemin sonlu olup olmamasına bağlıdır. Sonlu sistemin sabit olan başlangıç koşulları, sistemin her sonlanmasından sonra başa döner ve simülasyonun doğal sonlanmasını ifade eden bir olay oluşur. Örneğin 08 -17:00 zaman aralığı

⁸⁷ M. AKHUN & B. DURMUŞOĞLU, Worker Allocation Problem In Cellular manufacturing : A case study, In **Proceedings of the European Simulation Symposium 1994**

⁸⁸ Houshyar age, 1993. s.56

içerisinde çalışan sonlu sistemler gibi. Değişmeyen başlangıç koşulu bu örnekte, makinenin boşta olma ve bir işgününün bitmesi gibi sonlanma olayıdır. Sonlu olmayan sistemler ne değişmeyen başlangıç koşullarına, ne de doğal sonlanma noktasına sahiptir. Sonlu olmayan sistemlere, hiç kapanmayan hastane ya da sürekli olarak günde 24 saat çalışan bir imalat sistemi örnek olarak verilebilir. Genelde çoğu imalat sistemleri, sonlu olmayan tarzdadırlar.

Çünkü bu sistemler kapanabilirler, ama tekrar aynı başlangıç koşullarına sahip olamazlar. Örneğin, ara stok envanteri bir günden diğerine direkt olarak aktarılır. Sonlu sistemlerde replikasyon sayısı belirlendiği için sonlu sistemi analiz etmek, sonlu olmayana göre daha kolaydır. Sonlu olmayan sistemlerin analizi karmaşıktır ve iyi bir istatistik bilgisini gerektirir.

Bir sonlu sistemde, başlangıç ve bitiş koşulları sistemin doğası tarafından tanımlanır. Başlangıç koşullarına ve replikasyon sayısına müdahale edilemediğinden dolayı, örneklem büyüklüğünün kontrol edilmesinde verilecek tek karar, gerçekleştirilecek simülasyonun replikasyon sayısıdır. Her replikasyonun birbirinden bağımsızlığı, rastgele sayı gruplarının kontrol edilmesi ile sağlanacağından, istatistiksel prosedürleri direkt olarak uygulamak mümkündür. Bu durum, replikasyon içindeki belirli bir gözlemin, önceki replikasyondaki gözlemden etkilenmediği anlamına gelir. Bu nedenle her replikasyondan alınan gözlem değerlerinin ortalama değeri, yığının ortalaması olarak göz önüne alınır ve normal dağıldıkları ifade edilir. Bu durumda, istatistiksel yöntemler kullanılarak performans değerine bağlı güven aralığı oluşturulur. Buna ek olarak, verilen güven seviyesini karşılayabilecek replikasyon sayısının belirlenmesi mümkündür⁸⁹.

Sonlu olmayan bir sistemde, sistemi değişmeyen başlangıç koşullarına döndürecek bir olay yoktur. Buna bağlı olarak, gerek başlangıç koşulları, gerekse simülasyon çalışma uzunluğunun seçiminde bir temel yoktur. Bu sistemlerde önemli olan, sistemin kararlı halini korumaktır. Sistemin kararlı hali

⁸⁹ Charles HARREL, Tımay KERİM, *Simulation Made Easy: A Manager's Guide*, Institute of Industrial Engineers, USA, 2000, s. 116-122

çok uzun bir zaman periyodunu kapsar. Başlangıç koşullarından bağımsız olarak, genelde değişmeyen bir dağılıma sahiptir.

Sistem davranışının kararsız durumdan, kararlı hale geçtiğini gösteren belirgin bir nokta yoktur. Sistemin performansını, kararlı durum evresinde tahmin etmek istediğimizden dolayı, kararsız hal esnasında kayıt edilen gözlemler sonuçlarımızı saptıracaktır. Bu başlangıç kararsızlık evresi “warm-up period “ (kararlı hale geçiş dönemi) olarak isimlendirilir⁹⁰.

Sistemin kararlı hale geldiğini ifade eden bir çok istatistiksel yöntem vardır. Bunlardan bazıları, Gafarian yöntemi, Conway yöntemi ve Fishman yöntemleridir. Bunlara ek olarak, Grafik diyagramları bu noktanın belirlenmesinde yardımcı olur. Diğer bir yöntem ise başlangıçtaki sapma değerini azaltmak amacıyla, simülasyonu daha uzun bir süre çalıştırmak ve kararlı hale geçiş döneminin sonlandığı zamanı gözlemektir.

2.8 VARYANS İNDİRGEME TEKNİKLERİ (VİT)

Simülasyon çalışmalarında, istatistiksel analizin altında yatan temel düşünce, sistem performansının tahmin edilmesidir. Bunun sonucunda, doğru istatistik yöntemlerin kullanılması çok önemlidir. Eğer simüle edilecek sistem oldukça karmaşık ve simülasyon periyodu çok uzunsa, çalışma çok uzun zaman alabilir. Bunun ötesinde eğer, tasarım noktası sayısı oldukça fazla ise, gerçekleştirilecek simülasyon için günler gerekebilir. Büyük replikasyon sayılarını azaltabilmek ve daha gerçekçi sonuçlar alabilmek için, varyans indirgeme teknikleri olarak ifade edilen verimli istatistiksel sonuçlar sağlayan teknikler kullanılır. Burada verimli istatistiksel sonuç ifadesi, rastgele çıktı değişkeninin varyansının azaltılmasıdır. VİT, model için gerekli olan replikasyon sayısını azaltarak, bilgisayarda harcanacak zamanı azaltmasına rağmen, genelde bu tekniklerin kullanımı analizi karmaşıklaştırır. VİT deneyim ve beceri gerektirir. Bunun ötesinde VİT'in

⁹⁰ Susan M. SANCHEZ, *ABC of Output Analysis*, Proceedings of 2001 Winter Simulation Conference, s,30-35

gerçekten varyansı azaltacağı konusunda genel bir öngörü olmayabilir. Bazı durumlarda VİT'in kullanımı bilinen diğer analizden daha zayıf sonuçlar üretebilir. Buna rağmen çoğu durumda bu prosedürler, tahminleyicinin varyansını indirgemede etkindir. VİT'in en önemlileri⁹¹:

- 1) Dolaylı tahmin,
- 2) Kontrol değişkeni,
- 3) Ortak rastgele sayılar.

Bunların arasında en popüler ve kullanışlı olan VİT, ortak rastgele sayılar yöntemidir.

Dolaylı Tahmin: Bazı durumlarda, analiz edilecek performans değeri analitik metotlarla hesaplanabilir. Bu analitik formüller, performans değerinin dolaylı olarak tahmininde kullanılabilir. Dolaylı tahmin, bu değer için direkt tahmininden daha küçük varyansa sahiptir. Dolaylı tahminlerin kullanımı, modellenen sisteme ve sistem hakkındaki bilgiye bağlıdır. Örneğin, çoğu simülasyonda, model kuyruğa bağlı performansı ölçer. Kuyruk teorisi ve onun analitik ilişkileri, performansın dolaylı tahmininde kullanılır. Çoğu durumda, dolaylı ölçüm daha dar bir güven aralığı sağlar ki, varyansı %30-40 arası indirger. Dolaylı ölçümlerin esas dezavantajı, çeşitli durumlar için bilinen bir ilişkinin bulunmasının zorluğudur. Buna karşın bu metodun avantajı, basitliği ve aynı simülasyon çalışmasına bağlı direkt ve dolaylı ölçümleri sağlayabilme yeteneğidir.

Kontrol Değişkeni Yöntemi: Kontrol değişkeni yöntemi, sistem hakkındaki bilgiye dayanır. Bu durumda ilgilenilen rastgele değişken ile, kontrol rastgele değişkeni arasında kesin bir matematiksel ilişkinin bilinmesine gerek yoktur. Değerlerin ilişkili olduğunu bilmek yeterlidir. Kontrol rastgele değişkeni gelişler arası zaman gibi, sistem hakkında bilgiyi temsil eder. Örneğin, bir replikasyondan diğerine, gelişler arası gözlenen değerler beklenen değerden

⁹¹ Sheldon M. ROSS, *A Course in Simulation*, Mc Millan Pub. Co., USA, 1990, s.112-120

küçük ya da büyük olabilir. Gözlemlenen gelişler arası süre beklenen değerden fazla ise, daha az sıklıkta varlıkların gelmesi beklenir ve çıktı değerinin de beklenenden düşük seviyede olması tahmin edilir. Bunun gibi, daha fazla varlık geldiğinde, çıktı değerinin de beklenenden daha yüksek seviyede olması beklenir. Bunun sonucunda, rastgele değişken C, (örneğin gelişler arası süre) çıktı değişkenini (X) kısmi olarak kontrol eder.

Bu yöntem şu şekilde formüle edilebilir:

$$X_{ci} = x_i - a(c_i - E(c))$$

$$a = \text{Cov}(X,C) / s^2(C)$$

- X_{ci} : İlgili rastgele değişkenin düzeltilmiş gözlem değerleri
 x_i : İlgili rastgele değişkenin gözlem değerleri
 c_i : Kontrol rastgele değişkenin gözlem değerleri
 $E(C)$: Kontrol rastgele değişkenin beklenen değeri
 a : Ağırlıklandırma faktörü

Düzeltilmiş rastgele değişken X_c , x ve C arasında korelasyon bulunduğu ve ağırlık faktörünün kullanılmasından dolayı, direkt olarak gözlemlenmiş rastgele değişken X'den daima daha az varyansa sahiptir.

Bu yöntemin kullanılması oldukça kolaydır ve dolaylı tahmin yöntemindeki gibi, kontrol değişkenleri olsa da olmasa da varyansın hesaplanmasında aynı replikasyon kümesi kullanılır.

Ortak Rastgele Sayılar Yöntemi: Çoğu simülasyon dillerinin, bu yöntemi desteklemesinden dolayı sıklıkla kullanılır. Bu yöntem iki ya da daha fazla alternatif tasarımın, birbiriyle karşılaştırılmasında kullanılır.

Ortak rastgele sayılar yönteminde temel düşünce, deneysel koşulların iniş çıkışlarından ziyade, sistem özelliklerinin farklılıklarına dayalı performans değişiklikleridir. Simülasyonda bu deneysel koşullar, simüle edilmiş zaman doğrultusunda model oluşturmak için kullanılan türetilmiş rastgele değişkenlerdir.

Sonuç olarak, simülasyon modelleri, problem çözme ve karar verme problemlerinde artan bir şekilde kullanılmaktadır. Bu modellerin geliştiricileri ve kullanıcıları, model sonuçlarından ortaya çıkan bilgiden yararlanan karar vericiler ile bu tür modellere dayanan kararlardan etkilenen insanlardır. Modelin doğrulanması, sistem ile bilgisayarda sayısal hale dönüştürülmüş öngörülen model değerleri arasında, kabul edilebilir bir doğruluk (kararlılık) bulunmasıdır. Modelin geçerliliği ise sıklıkla öngörülen modelin bilgisayar programlaması ve bu programın çalışması durumunda olumlu sonuçlar almak olarak ifade edilir.

Modeli doğrulamak hem zaman isteyen, hem de maliyetli bir iştir. Bunun yerine, modelin doğru olabileceği yeterli bir güven aralığı belirlemek ve bu değerleri karşılayabilecek test ve değerlendirmeler yapmak daha uygundur.

ÜÇÜNCÜ BÖLÜM

SİMÜLASYON DİLLERİ VE KULLANIM ALANLARI

3.1 SİMÜLASYON DİLLERİNİN SINIFLANDIRILMASI

Bu bölümde simülasyon dillerinin iki ana sınıfta toplandığı ifade edilmiştir:

1. Simülasyon yazılımları,
2. Simülatörler.

3.1.1 Simülasyon Yazılımları

Simülasyon yazılımı, genelde paket program şeklinde simülasyon analistine sunulur. Belirli uygulamalar için özel niteliklere sahiptir. Örneğin SIMAN ve SLAM2, konveyörler ve AGV ler(Otomatik olarak yönlendirilen araçlar) için imalat modüllerine sahiptir. Simülasyon yazılımında modelin gelişimi, kullanılan simülasyon dilinin modelleme elemanları aracılığı ile gerçekleştirilir. Genellikle bu modelleme elemanları; örneğin bloklar, QUEUE (kuyruk), DELAY (işlem, hizmet süresi), RELEASE (kuyruktan çıkma), GENERATE (yaratma) gibi temel makro kelimeleri içerir. Bu kelimeler, sistem içerisinde hareket eden varlıkların faaliyetlerini açıklayabilmek için kullanılır⁹².

Bir varlık, bloğa girdiğinde; simülasyon yazılımının belirlediği ilgili rutin işlemleri gerçekleştirir. Simülasyon yazılımlarının en önemli avantajı, herhangi bir sistemi modelleme yeteneğidir. Diğer avantajları ise şöyledir:

- Simülasyon yazılımları, doğrulama, geçerliliği sınama, girdi ve çıktı analizi gibi çoğu simülasyon çalışmasını yapabilme yeteneğine sahiptir.
- Makro kelimeler içeren doğal bir dil oluşturarak kolay öğrenimi sağlar.
- Simülasyon yazılımında oluşturulan simülasyon modelleri, genelde kolay bir şekilde değiştirilebilir.

⁹² Allan CARRIE, *Simulation of Manufacturing Systems*, John Wiley Sons, Londra, 1998, s.156-162

- Çoğu simülasyon yazılımı, dinamik depolama paylaşımı sağlayarak hata tespitini kolaylaştırır.

Simülasyon yazılımlarının istenen niteliklerinden bazıları, aşağıda özetlenebilir:

- **Girdi esnekliği:** Simülasyon yazılımının tasarımı esnek olarak oluşturulmalıdır ki, girdinin tanımı için harcanan çaba minimum düzeyde olsun.
- **Yazım kolaylığı:** Simülasyon yazılımında kullanılan yazım şekli, kullanıcı dostu olmalı ve grafik tabanlı bir yapıya sahip olmalıdır. Bu sayede model gelişimi esnasında hatalar azaltılmış olur.
- **Yapısal Modülerite:** Simülasyon yazılımları modüler bir anlayışla tasarlanmalıdır. Böyle bir modelleme ortamında modelin her bir varlığı diğerlerine dokunmadan değiştirilebilir.
- **Standart istatistik türetimi:** Simülasyon yazılımı, girdi ve çıktı analizini gerçekleştirebilecek istatistiksel türetime ihtiyaç duyar.
- **Animasyon:** Yüzeysel analiz yapabilmek için animasyon uygundur. Örneğin, bir sistemde darboğaz noktasını belirleyebilmek için animasyon kullanılabilir. Buna rağmen animasyon bir analiz aracı değildir ve istatistiksel metot yerine geçmez⁹³.
- **İnteraktif model denetimi:** Simülasyon yazılımları, simülasyon modelinin oluşturulmasında birçok seçeneğe sahiptir. Örneğin, adım adım modelin oluşturulması, modelin daha iyi anlaşılmasını ve modelin hatalarının ayıklanmasını sağlar.
- **Maliyet:** Modellenecek sistem için gerekli esneklik ve spesifikasyonlara sahip olan Simülasyon yazılımları arasında, maliyet fayda analizi yapıldığında, fiyat avantajı aranılan nitelikler arasındadır.

⁹³ Roger Mc HANEY, **Computer Simulation: A Practical Perspective**, Academic Press Inc., California, 1991, s.27-29

Simülasyon yazılımlarının seçimi çalışmanın başarısını etkilemektedir. Bu nedenle çalışmanın amacının dikkatli bir şekilde belirlenmesi ve amacı gerçekleştirecek ihtiyaçların özenle ortaya çıkarılması gerekmektedir.

3.1.1.1 Simülasyon yazılımı seçimi

Simülasyon yazılımının seçiminde dikkate alınacak birçok özellik vardır. Bu özelliklerin en önemlileri aşağıda özetlenmiştir⁹⁴:

1. Kullanım kolaylığı gibi tek bir konu üzerinde yoğunlaşmamak gerekir. Bunun yanı sıra, doğruluk ve detay izleyebilme kapasitesi, öğrenilme kolaylığı ele alınmalıdır.
2. Hız önemlidir. Hız model gelişim süresini etkiler. Modeldeki hataları ayıklayabilmek için modelin sık sık çalıştırılması gerekir ki buda oldukça zaman alır.
3. Yazılımlardaki tanıtım bölümleri (demonstrasyon) göz önüne alınmamalıdır. Bunlar örnek problemi çözmek için geliştirilmiştir.
4. Satıcıya problemin çözümünde, yazılımın küçük bir versiyonunun olup olmadığı sorulmalıdır.
5. Yazılımın sunduğu önceden belirlenmiş spesifik varlıkları her model içerisinde kullanmaktan kaçınılmalıdır. Çünkü bu varlıklar modele uygun olmayabilir.
6. Yazılım paketinin 3. nesil programlama dillerini desteklemesine dikkat edilmelidir.
7. Grafik bazlı yazılımlarla programlama bazlı yazılımların avantaj ve dezavantajları iyi analiz edilmelidir. Problemin hangi tip yazılıma uygun olacağına karar verilmelidir⁹⁵.

⁹⁴ Jack KLEIJNEN, William Von GROENENDAL, *Simulation, A Statistical Perspective*, John Wiley Sons, England, 1992, s. 120-128

⁹⁵ James HARRINGTON, Kerim TUMAY, *Simulation Modelling Methods (To Reduce Risks and Increase Performance)*, Mc Graw Hill, USA ,2000, s. 65-75

3.1.1.2 Simülasyon Paketleri

Son zamanlarda kişisel bilgisayarlarda çalışan simülasyon paketleri Microsoft Windows 95/98, Windows XP/Server 2003, Linux işletim sistemlerini kullanmaktadır. Bu paketlerin az bir kısmı da UNIX iş istasyonlarında ve Apple makinelerinde çalışır. Özellikleri bakımından farklılıklar gösterse de, genelde ortak yönleri oldukça fazladır⁹⁶.

Ortak özellikleri arasında grafik kullanıcı arabirimi, animasyon ve sistem performansının değerlendirilmesi için otomatik çıktı verisi toplama sayılabilir. Sanal olarak tüm paketlerde, simülasyon esnasında simülasyon sonuçları grafik ya da tablo şeklinde ifade edilebilir. Yine grafik ya da tablo olarak değişik senaryolardan elde edilen çıktı değerleri karşılaştırılabilir. Performans değerleri ve karşılaştırmalarında çoğu simülasyon paketi, güven aralıklarından oluşan bir istatistiksel analiz gerçekleştirebilir.

3.1.1.2.1 ARENA Simülasyon Paketi

İş, Standart ve Profesyonel sürümleri olarak Systems Modeling Corporation tarafından sunulmuştur. ARENA programı kesikli ve sürekli sistemlerinin simülasyonunda kullanılır. ARENA ürün ailesine son olarak OPTQUEST adında simülasyon optimizasyon paketi eklenmiştir⁹⁷.

ARENA Standart Sürümü: Daha detaylı kesikli ve sürekli sistemler için tasarlanmıştır. İlk olarak 1993 yılında piyasaya sunulmuş ve modelin gelişiminde tam bir grafik desteği sağlayabilmek için nesne tabanlı tasarım kullanmıştır. Sistemin mantığını ve fiziksel bileşenlerini ifade edebilmek için modül olarak adlandırılan grafiksel nesnelere oluşturulmuştur. Modüller, veri içeren simgelerden oluşan bir diyalog penceresidir. Bu simgeler varlık akışını

⁹⁶ <http://www.fortunecity.com/simulation>, Erişim 10.10.2003

⁹⁷ Michael DREVNA, Cynthia J. KASALES, Introduction to Arena, **Proceedings of the 1994 Winter Simulation Conference**, s. 431-435

ifade edebilmek için birbirleriyle bağlanmıştır. Modüller, şablon olarak ifade edilen koleksiyonlarda organize edilir⁹⁸.

ARENA standart sürümü, uygulamaların tüm türlerinin modellenmesini ve genel amaçlı niteliklerini sağlayan temel ARENA şablonudur. Standart niteliklere ek olarak; ARENA şablonu imalat ve malzeme taşıma sistemlerinin özel amaçlarını karşılayabilecek modülleri de içerir. ARENA standart sürümü, örneğin ilaç ve kimyasal üretim gibi kesikli ve sürekli sistemlerin bir araya geldiği modelleri de oluşturmak için kullanılır..

ARENA İş Sürümü: İş süreçleri ile diğer sistemlerin, yüksek seviyede analizi ve modellenmesine odaklanmıştır. Süreç dinamiklerinin, hiyerarşik düzenini akış şeması şeklinde ifade eder ve sistem bilgisini veri tabloları şeklinde saklı tutar. Program içerisine entegre edilmiş faaliyet odaklı maliyetleme ve Visio şema çizme programı içermektedir.

ARENA Profesyonel Sürümü: ARENA standart sürümüne ek olarak terminoloji, süreç mantığı, veri, performans değerlendirme ve animasyon da içerir. ARENA'nın kalbi, SIMAN simülasyon dilidir. Simülasyon modellerinin animasyonunda, ARENA'nın temel modelleme elemanları; kuyruk analizi, kaynakların kullanım oranları ve varlık akışıdır. ARENA'nın girdi veri analiz modülü, uygun dağılım ve parametre seçimini otomatikleştirir.

3.1.1.2.2 AUTOMOD

AUTOMOD, simülasyon ürününün esas ilgi alanı imalat ve malzeme taşıma sistemleridir⁹⁹. AUTOMOD, çoğu genel malzeme taşıma sistemlerine uygun, hazır şablonlara sahiptir. Malzeme taşıma sistemleri şu bileşenleri içerir: Araç sistemleri, konveyörler, robotlar ve köprü kreynerlerdir. Malzeme taşıma şablonlarına ek olarak AUTOMOD tam destek veren bir simülasyon diline sahiptir. Üç boyutlu animasyon özelliği ile gerçek zaman içinde herhangi bir

⁹⁸ **Arena Variables Guide**, Systems Modeling Corporation, 1994

⁹⁹ www.simul8.com/products/automod.htm Erişim 12.09.2004

açından görüntü alınabilir. Bir AUTOMOD modeli bir veya daha fazla sistemden oluşabilir. Bu sistem, akış ve kontrol mantığının tanımlandığı bir süreç sistemi ya da malzeme aktarma şablonlarının biri, yani hareket sistemi olabilir. Her model, bir süreç sistemine ve çoklu sayıda hareket sistemine sahip olabilir. Süreçler gerek imalat malzemeleri gerekse denetim mesajlarının kontrol edilmesi için gelişmiş mantığı içerir.

3.1.1.2.3 QUEST

Denep Robotics firması tarafından, iş hücresi simülatörü olarak piyasaya sunulmuştur. Daha sonra imalat odaklı simülasyon paketine dönüştürülmüştür¹⁰⁰. Igrip ve Ergosim adlı alt süreçlerden oluşur. QUEST modelleri üç boyutlu bilgisayar destekli tasarım geometrisine dayalıdır. QUEST grafik kullanıcı arabirimi ile çalışan simülasyon analistleri, konveyör ve robotlar için oluşturulmuş malzeme akış modüllerinden meydana gelen simülasyon ortamını yaratır.

3.1.1.2.4 EXTEND

Yeni blokların yaratılmasında blok diyagram yaklaşımı kullanır. Proses odaklı olmasına rağmen hem sürekli hem de kesikli modellere uygulanabilir. Blok diyagramlarına simgesel animasyon kolaylığı sağlar. Her blok bir simgeye sahiptir. Ve içinde kod, parametreler, kullanıcı arabirimi ve yardım dosyalarını içerir. Son kullanıcılar blokları yer değiştirerek ve birbirine bağlayarak model oluşturabilirler. Ve blokların içine parametre değerlerini bir pencere vasıtasıyla girebilirler. Blok grupları bir alt model temsil edecek şekilde hiyerarşik olarak kümelenir. Bu sayede yüksek seviyeli blok diyagramı bir çok detayı gizleyerek son kullanıcıya gösterilir.

¹⁰⁰ www.surrey.ac.uk/MME/QUEST/ Erişim : 24.08.2004

Yeni blokların yaratılması için kullanılan EXTEND; aynı zamanda MOD L adlı C tabanlı bir simülasyon diliyle derlenebilir. MOD L, mesaj iletişimi ve özel kullanıcı arabirimi desteği ile EXTEND'e destek sağlar.

3.1.1.2.5 MICROSAIN

Görevlerden oluşan ağın tanımlanmasında, akış diyagramları yaratılarak model geliştirilir. Animasyon simgesel ve akış şeması sunumuna dayanır. Menüler ve sayılaştırılmış simgeler doğrultusunda, sıralanmış kuyruk ve durumsal görevlere sahiptir. MICROSAIN imalat, sağlık, perakende ve askeri uygulamalarda kullanılır. Ergonomi ve insan performans modellemeyi de destekler. OPTQUEST ile optimizasyon yeteneğine sahiptir.

3.1.1.2.6 PROMODEL

İmalat sistemlerinin modellenmesinde kullanılan, simülasyon ve animasyon aracıdır. PROMODEL şirketi tarafından yaratılan bu simülasyon paketinin, çeşitli versiyonları bulunmaktadır. Örneğin; sağlık sektörü için MEDMODEL, hizmet sistemleri için SERVICEMODEL, iş süreçlerinde akış şeması bazlı PROCESSMODEL üretilmiştir. PROMODEL üç boyutlu perspektife sahiptir.

PROMODEL, imalat odaklı modelleme elemanlarına ve kurala dayalı karar verme mantığına sahiptir. Bazı sistemler, PROMODEL'in yüksek dereceli parametre eklenmiş modelleme elemanlarının seçimi ile modellenebilir.

PROMODEL' da modelleme elemanları, parçalar ya da varlıklar, konumlar, kaynaklar, yol ağları, rotalama ve süreç mantığı ile gelişlerdir. Parçalar, bir konumdan diğerine rotalama ve süreç mantığını izleyerek gider. Kaynaklar, insanları temsil etmekte kullanılır. Araçlar ise, konumlar arasındaki taşınan parçaları ifade eder. Kaynaklar daha önce tanımlanmış hız, ivme, yükleme ve boşaltma zamanlarına bağlı olarak hareket eder. Rotalama ve süreç elemanı,

kullanıcının tanımladığı prosedürün mantığına bağlı olarak, simülasyon programında kullanılır.

3.1.1.2.7 Taylor Enterprise Dynamics (TAYLOR ED)

Taylor ED imalat , depolama, hizmet ve veri akış süreçlerini içeren malzeme taşıma sistemlerinin modellenmesinde kullanılır. Aynı zaman da, akış süreçlerinin, gerçek zamanın ortaya çıkarılmasında kullanılır. Hem iki hem de üç boyutlu animasyonu destekler.

Taylor ED atomun yapısına dayandırılmıştır. Atomlar, bu simülasyon paketinin akıllı nesnelere olarak model geliştirme kaynaklarını oluşturur.

Bir atom dört boyutu olan (X,Y,Z ve zaman) bir nesnedir. Her atom konum, hız, rotasyon ve zamana bağlı dinamik davranış özelliklerine sahiptir. Atomlar diğer atomların davranışlarından etkilenebilir ve diğer atomları içerebilirler. Bunlar yaratılıp yok edilebilir. Bu özelliklerden yararlanılarak atomun dört boyutu varlıklarla ilişkilendirilir. Varlıkların sahip olduğu üç boyuta zaman boyutu da eklenerek model oluşturulur.

3.1.1.2.8 WITNESS

WITNESS, imalat makinelerine odaklı bir programdır. Makinelerin spesifikasyonlarına bağlı olarak modellemeyi kolaylaştırır. Kesikli parça modellemesinde birçok eleman içerir. Aynı zamanda akışkan bazlı süreçlerin oluşturduğu sürekli sistemlerin modellenmesinde kullanılır¹⁰¹.

WITNESS modelleri, şablon elemanlarına dayanır. Elemanlar tasarımcının eleman modülüyle bir araya getirilerek uyumluluk sağlanır. Makine elemanı tek,

¹⁰¹ www.witness-for-simulation.com/simulation/witnessoptimization.htm Erişim: 05.01.2005

yığın üretim montaj, çoklu istasyon şeklinde olabilir. Bu simülasyon paketi, iki boyutlu yerleşim animasyonu içerir¹⁰².

3.1.1.2.9 GPSS

GPSS (General Purpose Simulation Systems), süreç tabanlı simülasyon dilidir. 1961 yılında IBM şirketi tarafından geliştirilmiştir. Bir çok simülasyon modelinde, kuyruk teorisine bağlı olarak oldukça popüler bir hale gelmiştir. Sonraki yıllarda, daha ileri versiyonları olan GPSS/H ve GPSS/PC versiyonları geliştirilmiştir.

GPSS/H 1977 yılında VOLVERINE yazılım tarafından geliştirilmiştir. GPSS/H derlenmiş bir dildir ve ilk GPSS versiyonları ile kıyaslandığında 5 kat daha hızlıdır. Bu versiyonda, gerçek sistem zamanının tespiti diğer yazılım dosyalarını okuma ve yazabilme yeteneği, yönetilebilen çıktı raporları düzenleme, gelişmiş kontrol komutları ve matematiksel fonksiyonlar ile olasılık dağılımlarından, rastgele sayı değerleri üretebilme niteliklerine sahiptir. Bu niteliklerden dolayı başka bir programlama diline ihtiyaç duymaz.

GPSS/H altmıştan fazla standart komuttan oluşur. Bu komutlar, bloklarla şekilsel olarak ifade edilir. GPSS modelini oluşturabilmek için, standart blokların yer aldığı bir diyagram oluşturulur. Blok diyagramı oluşturulduktan sonra, bu komutlar GPSS yazılımına aktarılır. Sistemdeki varlıklar, GPSS içinde geçişler olarak nitelendirilir ve bu geçişlerin değerlerine parametre adı verilir.

GPSS/PC: Bu versiyonda hata ayıklama, girdi hata kontrol ve blok diyagramındaki faaliyetlerin, grafiksel gösterimi mevcuttur. GPSS/PC bir derleyici olmadığından dolayı, modelde yapılan değişiklikler programın yeniden derlenmesi beklenmeden anında görülebilir.

¹⁰² M.R. Rotab KJHAN, S.C. HARLOCK, Computer Simulation of Production systems for Woven Fabric Manufacture, **Computers & Industrial Engineering**, Vol. 37, 1999, s. 747-749

3.1.1.2.10 SIMAN (Simulation Anaysis)

SIMAN, süreç odaklı modelleme, olay odaklı modelleme ya da her ikisinin de kombinasyonu olarak, model geliştiren bir simülasyon dilidir. Tipik bir uygulamada, çoğu simülasyon modeli, süreç oryantasyonu kullanılarak geliştirilir. Süreç yaklaşımında, imkansız ya da uygun olmayan geliştirilmiş karar verme mantığı, olay rutinlerinde kodlanabilir ve buna süreç modeli denir. SIMAN Systems Modeling şirketi tarafından, 1987 yılında Dennis PEGDEN tarafından geliştirilmiştir. SIMAN, mikrobilgisayarlar için uygun olan ilk simülasyon dili olduğundan ve ayrıca imalat elemanları için özel nitelikler taşıdığından dolayı çabuk kabul görmüştür¹⁰³.

SIMAN, süreç simülasyon modeli iki belirgin kısımdan oluşur. Model yapısı ve deneysel yapı. Bu yapılar ayrı dosyalarda tutulur. Model yapısında, modelin yapı taşları, birbiriyle dinamik etkileşim halinde bulunan varlık ve kaynakların mantıksal ilişkisini tanımlar. Her blok, onu temsil eden şekilsel gösterime sahiptir. Bu şekiller doğrusal olarak yukarıdan aşağıya blok diyagramı şeklinde bir araya getirilir.

Deneysel yapıda modelin yapı taşları, belirli parametre değerlerini tanımlayabilmek, kaynak türü ve miktarını belirlemek ve istenen çıktı istatistiğini gerçekleştirmek için kullanılan elemanlardır.

SIMAN çıktı işlemcisi, aynı ya da farklı sistem konfigürasyonlarından oluşan simülasyon çıktı değerlerinin hipotez testleri ve güven aralıklarının belirlenmesine olanak sağlar. Buna ek olarak, istatistiksel grafik sunumlarının üretilmesini destekler.

SIMAN, tüm bilgisayar sınıfları için uygundur. Mikrobilgisayar sürümlerinde süreç odaklı blok diyagramını, eş zamanlı grafiksel işlemci kullanarak, blokları oluşturmak mümkündür. Elementler olarak isimlendirilen benzer bir program ise,

¹⁰³ Bernard P. ZEIGLER, *Theory of Modeling and Simulation*, Academic Press, New York, 2000, s. 78-79

deneysel yapıyı geliřtirmede kullanılır. Bu yetenek model geliřim sürecinin doęruluk ve hızını arttırır. Model geliřtirmek için kullanılan temel bloklar, varlıklar, kuyruklar ve kaynaklardır.

3.1.1.2.11 SIMSCRIPT

SIMSCRIPT, süreç ya da olay odaklı bir simülasyon dilidir. 1962 yılında Herry Markowich tarafından geliřtirilmiřtir. Kesikli olay , sürekli olay veya her ikisini de içeren bir programlama dilidir. Genel süreç yaklařımı, geliřmiř veri yapısı ve güçlü kontrol komutlarından dolayı SIMSCRIPT, büyük ve karmařık modellerin simüle edilmesinde kullanılır. Mikro bilgisayar ve iř istasyonlarının sürümlerinde "SIMGRAPHICS" olarak adlandırılan animasyon ve grafik paketi bulunmaktadır. Bu paket, dinamik ve statik sunum grafiklerinde, histogram çizimlerinde ve deęiřkenlerin zaman grafiklerinde kullanılır¹⁰⁴.

SIMSCRIPT'in süreç bölümünün temel modelleme elemanları, süreç varlıkları, kaynaklar ve kuyruklardır. Bir süreç varlığı, ilgili süreç içersinde ilerler ve çeřitli niteliksel deęerler alır. SIMSCRIPT'te, bir simülasyon modeli oluřturmak için, analist bir ana program yazmalı ve her sürece karřılık gelen bir süreç rutini belirlemelidir.

3.1.1.2.12 SLAM (Simulation Language for Alternative Modeling)

SLAM, gerek süreç gerekse olay odaklı modellemede kullanılan bir dildir. Süreç modelini oluřturmada, bařlangıç sistem için gerekli grafiksel aę diyagramının analist tarafından belirlenmesi gerekir. Aę diyagramı, dallar ve uçlar olarak tanımlanan standart semboller kümesinin bir araya getirilmesiyle oluřturulur. Bu diyagramda, uçlar varlıkların yaradılıřına, dallar ise zamanın geçiřine karřılık gelir. Sistemin birleřmiř aę modeli, daha sonra bu modele denk gelen SLAM II program komutlarına dönüřtürülür.

¹⁰⁴M.Law Averell, Chris LARMEY **Introduction to Simulation Using Simscript II.5**, 1992, CA, s.24-28

SLAM II, deęişik türlerde bilgisayar platformu ve istenilen animasyon yeteneğine baęlı olarak mevcuttur. Temel SLAM II dili animasyon içermez, ancak SLAMSYSTEM olarak isimlendirilen, mikrobilgisayar sürümü MS Windows ile entegre edilmiştir.

3.1.2 Simülatörler

Bir imalat simülatörü, en temel yapısında programlamaya ihtiyaç duymadan, önceden belirlenmiş bir imalat sisteminin sınıfında, sistemin simüle edilmesine olanak saęlayan bir yazılım paketidir. Örneęin, imalat, bilgisayar ve haberleşme sistemlerinin belirli türlerinde mevcut simülatörler kullanılmaktadır. Simülatörün, herhangi bir programlamaya ihtiyaç duyulmaksızın menü ve grafikleri aracılığıyla kullanımı mümkündür.

Simülatör için tipik modelleme elemanları; parçalar, makineler, kuyruklar, taşıyıcılar ve konveyörlerdir. Simülatörün en büyük avantajı, programı geliştirebilmek için, gereken zamanın bir simülasyon diline göre daha az olmasıdır. Bu özellik, birçok iş ortamında zaman kısıtlı olduğundan büyük önem arz eder. Bir dięer avantajı, bu yazılım paketini kullanan personelin simülatöre kolay uyum saęlamasıdır. Programlama deneyimi olmayan ya da simülasyondan seyrek olarak yararlanan personel simülatörleri tercih etmelidir.

Simülatörlerin en büyük dezavantajı, standart özelliklerine baęlı olarak kısıtlı modelleme yetenekleridir. Bu durumu aşabilmek için, yani karmaşık karar verme mantığını oluşturan modellemeyi gerçekleştirebilmek için, programlamaya benzeyen komutlar eklemek gerekir. Simülatörlere örnek olarak SIMFACTORY, STARCELL, WITNESS, XCELL+, VE PROMODEL verilebilir.

3.3 SİMÜLASYON DİLLERİNİN KARŞILAŞTIRILMASI

Çoğu simülasyon dili, oldukça benzer temel modelleme yapıtaşlarına sahiptir. Tablo 9'da görüldüğü gibi GPSS (H ya da PC) , SIMAN, SIMSCRIPT, SLAM II / SLAMSYSTEM dilleri; yeni varlıkları yaratabilmek ve gerekli istatistik değerlerini elde etmek için, programlama komutlarını kullanır.

Çoğu simülasyon dili, kuyruk modeli odaklıdır ve GPSS ile SIMAN ve SLAM II dillerinin süreç kısımları, bu tür problemlerin çözümü için gerekli yapıtaşlarına sahiptir. SIMAN ve SLAM II dilleri, aynı zamanda birçok temel olay çizelgeleme yaklaşımını destekler.

SIMSCRIPT, temel simülasyon dilleri arasında en genel süreç yaklaşımına sahiptir. Bunun sonucunda, sanal olarak herhangi bir sistem olay çizelgeleme yaklaşımını kullanmaksızın modellenenabilir. Buna rağmen genel yapısından dolayı GPSS, SIMAN ve SLAM II de kullanılan standart kuyruk modeli problemlerinin kurulması için diğer diller de kullanılan kodlama yapısından fazlasını içerir. Daha karmaşık simülasyon modellerinde, SIMSCRIPT (kuyruksuz modellerde) tercih edilir. Çünkü, SIMSCRIPT geliştirilmiş kontrol komutları ve veri yapılarıyla zenginleştirilmiş, genel bir programlama dilidir.

Tablo 9, Temel simülasyon dillerinin özelliklerinin gerçekleştirilmesini gösterirken, Tablo 10'da simülasyon dillerinin özelliklerinin karşılaştırılması sunulmuştur.

Tablo 9 Temel Simülasyon Özelliklerinin Gerçekleştirilmesi

	DİLLER			
	GPSS (H/PC)	SIMAN	SIMSCRIPT	SLAM II/SLAMSYSTEM
Yeni varlıkların yaratılması	Generate	Create	Activate	Create
Kaynağın kullanımı ve serbest bırakılması	Seize/Release	Seize/Release	ReQUEST/Relinquish	Await/Free
Zamanın ilerlemesi	Advance	Delay	Work, Wait	Activity
Kesikli zaman istatistiği	Queue/depart, Tabulate	Tally	Tally	Colct
Sürekli zaman istatistiği	Queue/depart, Enter/Leave, Tabulate	Dstat	Accumulate	Timst

Tablo 10 Simülasyon dillerinin Karşılaştırılması

Özellik	DİLLER					
	GPSS/h	GPSS/pc	SIMAN	SIMSCRIPT	SLAM II	SLAMsystem
Olay(O) ya da Süreç (S) odaklı	S	S	O, S	O, S	O, S	O, S
Bilgisayar sınıflarına uyumluluğu	A,B,C	A	A,B,C	A,B,C	A,B,C	A
Animasyon	A	A	A,B ^d	A,B	B,C	A
Grafiksel Model Girdisi	Hayır	Hayır	A	Hayır	E	Evet
Kesikli ve Sürekli (Bütünleşik) Sim.	Hayır	Evet	Evet	Evet	Evet	Evet
Rastgele Sayı Kümelerinin sayısı	Genelde Limitsiz	Genelde Limitsiz	10 ⁹	10 ⁹	10 ⁹	10

A: Mikrobilgisayar

B: İş İstasyonu

C: Mini bilgisayar/MainFrame

D: Sadece Sunum

E: SLAM II kullanarak

O: Olay

S: Süreç

3.4. İMALAT SİSTEMLERİNİN SİMÜLASYONUNDA GELİŞEN YAKLAŞIMLAR

Günümüzde bilgisayar simülasyonu, araştırmacılar için imalat hatlarının ve diğer üretim sistemlerinin çözümünde kullanılan yaygın ve etkili bir yardımcı araç haline gelmiştir. Bilgisayar destekli tasarım, her zaman üretim sektörünün vazgeçilmez yardımcılarında biri olmuştur. Günümüzde mühendisler ayrı ayrı tasarlanan parçaları, üretim aşamasından çok önce bilgisayar ekranında, sanal gerçeklik programları aracılığıyla üç boyutlu olarak görme şansına sahiptirler¹⁰⁵.

3.4.1. Paralel ve Dağıtılmış Sistemlerin (Parallel and Distributed Systems) Simülasyonu

Geçen yıllar içinde bilgisayar teknolojisi, bilgisayar ya da işlemcilerin her birinin birbirine bağlandığı, paralel ya da dağıtılmış işlem yapan çevre birimlerine dönüşmesine yol açmıştır¹⁰⁶. Örneğin, göreceli olarak ucuz olan mini bilgisayarlar ve hatta mikro bilgisayarlar birbirlerine ağ ile bağlanabilirler. Ya da daha büyük bir bilgisayarda, birden çok işlemcinin bir araya gelerek birbirleriyle iletişim kurduğu bir yapıya ev sahipliği yaptığı görülür. Böyle bir çevrede, aynı zaman içinde ya da paralel olarak işlem yapan birbirinden bağımsız işlemcilerin yaptığı farklı görevleri, kendi aralarında paylaşmak mümkündür. Bunun sonucunda, görevi tamamlayabilmek için toplam zaman azaltılmış olur. Bunu gerçekleştirebilmek, görevin yanında yazılım ve donanıma da bağlıdır.

Simülasyon saati ve olay listesi, hangi olayların birbirini takip edeceğini belirtir. Simülasyon saati, olay zamanına bağlı olarak gelişir ve bilgisayar olay mantığına göre işlem yapar. Bu işlem sonucunda durum değişkenleri kuyruk ve olay listeleri güncellenir. Rastgele sayılar ve değişkenler türetilir. İstatistik

¹⁰⁵ Paulo, BRANDIMARTE, Agostina, VILLA, *Modelling Manufacturing Systems*, Springer Berlin, 1999, s. 116-124

¹⁰⁶ Helen, D. Karatza, *Modeling And Simulation Of Distributed Systems And Networks, Simulation Modelling, Practice and Theory*, 2004, Volume 12, s.183-185

değerleri toplanır. Bu mantık önceden belirlenen sistem zamanına kadar devam eder, ve işlemler sıralı bir şekilde bir tek makinede gerçekleşir. Dağıtılmış ve paralel işlemlerin simülasyonu, veri tabanı tasarımında uygulanmıştır.

Simülasyonun farklı işlemciler üzerine dağıtılmasının bir yolu dinamik simülasyondur. Dinamik simülasyon ile; farklı işler, ayrı işlemcilere dağıtılır. Direkt işler aynı işlemci üzerinde atanır. Bu işler ana simülasyon programını oluşturur. Ana simülasyon programı çalışmasına devam ederken, aynı zamanda endirekt işler farklı işlemcilere atanarak, simülasyon süresinin kısalmasına yardımcı olur. Destek fonksiyonları rastgele sayı üretimi, rastgele değişken üretimi, liste ve kuyrukların güncellenmesi ve istatistiksel veri toplanmasından oluşur. Simülasyonun, mantıksal gerçekleşimi hala sıralıdır. Örneğin, modelin kuyruk listesi güncellendiğinde, ana simülasyon programı gerekli mesajı, ilgili işlemciye iletir. Bu mesajlar, yapılması gereken işlemleri fonksiyon halinde ifade eder. Aynı zaman içinde, ana simülasyon programı; program mantığına dayalı olarak ilerlemeye devam eder.

Simülasyonun, farklı işlemciler üzerine dağıtılması için, bir diğer yol ise, modeli çeşitli alt modellere parçalamaktır. Örneğin, bir imalat tesisi genelde kuyruk istasyonlarının ağı olarak, her biri farklı bir aktiviteyi temsil edecek şekilde birbirine bağlanmıştır. Her bir alt model farklı işlemcilere atanarak, modelin parçalarını simüle eder. Alt modellerin birbirleri arasındaki mantıksal ilişkilere dayanarak, işlemciler birbirleriyle haberleşmek zorundadır. İmalat örneğinde bu durum şöyle açıklanır: Bir iş parçası bir kuyruk istasyonunu terk edip diğerine geçtiğinde, simülasyon faaliyeti de, bir işlemciden ilgili diğer işlemciye geçer. Faaliyetlerin, zaman sırasına göre dizilmesine dikkat edilmelidir. Modelin tüm faaliyetlerinin doğru bir şekilde ifade edilebilmesi için, farklı işlemciler üzerindeki alt modellerin işlemlerinin senkronize olması gerekir¹⁰⁷.

¹⁰⁷ Andrew S. TANENBAUM, Maarten van STEEN **Distributed Systems: Principles and Paradigms**, Prentice-Hall, New York, 2002, s. 68,86

Dağıtılmış simülasyonun en büyük avantajı, genel bir simülasyon saatine ve tam bir olay listesine ihtiyaç duymamasıdır. Geleneksel simülasyon modelinde, programı çalıştırabilmek için olay listesinin işlenmesi oldukça uzun zaman alır. Bu durumda, olay listesini devre dışı bırakmak iyi bir fikirdir. Sistemde simülasyon saati ve olay listesinin yerine geçen kavram, işlemciler arasındaki mesaj geçiştir. Her geçiş belirli bir zaman aralığını üzerinde taşır. Bunun dezavantajı, simülasyonun çıkmaza girmesine sebep olabilmesidir. Bir işlemci işlem yapabilmek için bir öncekinin mesajını beklemek zorundadır. Bu ise simülasyonun durmasına sebep olur ki bu çıkmazların tespiti ve yok edilmesi gerekmektedir.

3.4.2 Yapay Zeka İle Modelleme Ve Simülasyon

Yapay Zeka terimi, resmen ilk defa 1950'lerin ortasında kullanılmıştır. Yapay Zeka, başka türlü otomatikleştirilemeyecek zeka süreçlerini modellemek ve makineleştirmek için yöntemler sağlamaya gayret eder. Yapay Zeka alanında çalışmak; geleneksel bilgisayar yaklaşımlarına ilaveten; çoğunlukla sembolik, algoritmik olmayan problem çözme metotları ve bilgisayarların, sayılarla hesap yapmaktan ziyade, kavramlarla muhakeme yapmak için kullanımına dayanır.

Yapay Zeka, bir bilgisayar bilimi alanıdır. Temelleri, psikoloji, dilbilim, veri işleme, karar destek sistemleri ve modelleme gibi çeşitli diğer disiplinlere yakından bağlıdır ve tüm bu disiplinleri kaynak olarak kullanır¹⁰⁸.

Yapay zeka, insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak da tanımlanabilir. Yani programlanmış bir bilgisayarın düşünme girişimidir. Daha geniş bir tanıma göre ise, yapay zeka, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılmış bilgisayarlardır.

¹⁰⁸ Paul A. LUKER, Bernord SCHMIDT, **Knowledge Based Simulation**, Springer-Verland, Newyork 1991, s. 2-9

Yapay zeka konusundaki ilk çalışma, McCulloch ve Pitts tarafından yapılmıştır. Bu araştırmacıların önerdiği, yapay sinir hücrelerini kullanan hesaplama modeli, önermeler mantığı, fizyoloji ve Turing'in hesaplama kuramına dayanıyordu. Herhangi bir fonksiyonun sinir hücrelerinden oluşan ağlarla hesaplanabileceğini ve mantıksal “ve” ve “veya” işlemlerinin gerçekleştirilebileceğini gösterdiler. Bu ağ yapılarının uygun şekilde tanımlanmaları halinde öğrenme becerisi kazanabileceğini de ileri sürdüler. Hebb, sinir hücreleri arasındaki bağlantıların şiddetlerini değiştirmek için basit bir kural önerince, öğrenebilen yapay sinir ağlarını gerçekleştirmek de mümkün hale gelmiştir¹⁰⁹.

“1950’lerde Shannon ve Turing bilgisayarlar için satranç programları yazıyorlardı. İlk yapay sinir ağı temelli bilgisayar SNARC, MIT’de Minsky ve Edmonds tarafından 1951’de yapıldı. Çalışmalarını Princeton Üniversitesi’nde sürdüren Mc Carthy, Minsky, Shannon ve Rochester’le birlikte 1956 yılında, Dartmouth’da iki aylık bir toplantı düzenledi. Toplantının en önemli özelliği Mc Carthy tarafından önerilen “Yapay Zeka” adının konmasıdır. İlk kuram ispatlayan programlardan Logic Theorist (Mantık Kuramcısı), burada Newell ve Simon tarafından tanıtılmıştır.

Daha sonra Newell ve Simon, “insan gibi düşünme” yaklaşımına göre üretilmiş ilk program olan General Problem Solver (Genel Sorun Çözücü)’ü geliştirmişlerdir. Simon, daha sonra fiziksel simge varsayımını ortaya atmış ve bu kuram, insandan bağımsız zeki sistemler yapma çalışmalarıyla uğraşanların hareket noktasını oluşturmuştur.

Bundan sonraki yıllarda, mantık temelli çalışmalar egemen olmuş ve programların performanslarını sergilemek için bir takım yapay sorunlar ve dünyalar kullanılmıştır. Daha sonraları bu sorunlar gerçek yaşamı hiçbir şekilde temsil etmeyen “yapay dünyalar” olmakla suçlanmış ve yapay

¹⁰⁹Stuart J. RUSSEL, Peter NORVIG, **Artificial Intelligence: A Modern Approach**, Prentice Hall, New York, 2002

zekanın yalnızca bu alanlarda başarılı olabileceği ve gerçek yaşamdaki sorunların çözümünde kullanılamayacağı ileri sürülmüştür.

Geliştirilen programların, gerçek sorunlarla karşılaştığında çok kötü bir performans göstermesinin ardındaki temel neden, bu programların yalnızca yapay bir şekilde çalışıp konu ile ilgili bilgileri kullanmamasıydı. Weizenbaum tarafından geliştirilen bu dönemin en ünlü programlarından Eliza, karşısındaki ile sohbet edebiliyor gibi görünmesine karşın, yalnızca karşısındaki insanın cümleleri üzerinde bazı işlemler yapıyordu. İlk makine çevirisi çalışmaları sırasında benzeri yaklaşımlar kullanılıp çok gülünç çevirilerle karşılaşıncaya bu çalışmaların desteklenmesi durdurulmuştur.

Her sorunu çözecek genel amaçlı program yerine, belirli bir uzmanlık alanındaki bilgiyle donatılmış programlar kullanma fikri, yapay zeka alanında yeniden bir canlanmaya yol açtı. Kısa sürede "Uzman Sistemler" adı verilen bir metodoloji gelişti. Fakat burada çok sık rastlanan tipik bir durum, bir otomobilin tamiri için önerilerde bulunan uzman sistem programının otomobilin ne işe yaradığından haberi olmamasıydı.

İnsanların iletişimde kullandıkları Türkçe, İngilizce gibi doğal dilleri anlayan bilgisayarlar konusundaki çalışmalar bu sıralarda hızlanmaya başladı. Doğal dil anlayan programların, dünya hakkında genel bilgiye sahip olması ve bu bilgiyi kullanabilmek için, genel bir metodolojisi olması gerektiği belirtilmekteydi.

Bütün bu çalışmaların sonunda, yapay zeka araştırmacıları iki gruba ayrıldılar. Bir grup, insan gibi düşünen sistemler yapmak için çalışırken,

diğer grup ise rasyonel karar verebilen sistemler üretmeyi amaçlamaktaydı. Aşağıda bu yaklaşımlar kısaca incelenecektir¹¹⁰.

3.4.2.1 İnsan Gibi Düşünen Sistemler

İnsan gibi düşünen bir program üretmek için, insanların nasıl düşündüğünü saptamak gerekir. Bu da psikolojik deneylerle yapılabilir. Yeterli sayıda deney yapıldıktan sonra elde edilen bilgilerle bir kuram oluşturulabilir. Daha sonra bu kurama dayanarak bilgisayar programı üretilebilir. Eğer programın giriş/çıkış ve zamanlama davranışı insanlardaki ile aynı ise programın düzeneklerinden bazılarının insan beyninde de mevcut olabileceği söylenebilir.

İnsan gibi düşünen sistemler üretmek, bilişsel bilimin araştırma alanına girmektedir. Bu çalışmalarda asıl amaç, genellikle insanın düşünme süreçlerini çözümlenmede bilgisayar modellerini bir araç olarak kullanmaktır.

3.4.2.2 İnsan Gibi Davranan Sistemler

Yapay zeka araştırmacılarının baştan beri ulaşmak istediği ideal, insan gibi davranan sistemler üretmektir. Turing zeki davranışı, bir sorgulayıcıyı kandıracak kadar bütün bilişsel görevlerde insan düzeyinde performans göstermek olarak tanımlamıştır. Bunu ölçmek için de Turing testi olarak bilinen bir test önermiştir. Turing testinde denek, sorgulayıcıyla bir terminal aracılığıyla haberleşir. Eğer sorgulayıcı, denegin insan mı yoksa bir bilgisayar mı olduğunu anlayamazsa denek Turing testini geçmiş sayılır.

3.4.2.3 Rasyonel Düşünen Sistemler

Bu sistemlerin temelinde, mantık yer alır. Burada amaç, çözülmesi istenen sorunu mantıksal bir gösterimle betimledikten sonra, çıkarım kurallarını

¹¹⁰ E. ALTUNTAŞ, T. ÇELİK; <http://members.tripod.com/~Bagem/bagem/yz3.html>, Erişim 20.01.2005

kullanarak çözümünü bulmaktır. Yapay zekada çok önemli bir yer tutan mantıkçı gelenek, zeki sistemler üretmek için bu çeşit programlar üretmeyi amaçlamaktadır.

Bu yaklaşımı kullanarak gerçek sorunları çözmeye çalışınca iki önemli engel karşımıza çıkmaktadır. Mantık, formel bir dil kullanır. Gündelik yaşamdan kaynaklanan, çoğu kez de belirsizlik içeren bilgileri mantığın işleyebileceği bu dille göstermek, hiç de kolay değildir. Bir başka güçlük de, daha karmaşık sorunları çözerken kullanılması gereken bilgisayar kaynaklarının üstel olarak artmasıdır.

3.4.2.4 Rasyonel Davranan Sistemler

Bu yaklaşımda yapay zeka, rasyonel davranışların incelenmesi ve oluşturulması olarak tanımlanmaktadır. Rasyonel davranabilmek için, gerekli koşullardan biri de doğru çıkarımlar yapabilmek ve bu çıkarımların sonuçlarına göre harekete geçmektir. Ancak, yalnızca doğru çıkarım yapabilmek yeterli değildir. Çünkü bazı durumlarda doğruluğu ispatlanmış bir çözüm olmadığı halde, yine de bir şey yapmak gerekebilir. Bunun yanında, çıkarımdan kaynaklanmayan bazı rasyonel davranışlar da vardır. Örneğin, sıcak bir şeye dokunduğunda insanın elini çekmesi bir refleks harekettir ve uzun düşünce süreçlerine girmeden yapılır¹¹¹.

3.4.2.5 Yapay Zeka Türleri

Yapay zeka konusundaki araştırmalar şu gruplar altında toplanabilir.

- Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler
- Doğal Diller (Bilgisayar ile Doğrudan İletişim)

¹¹¹ Vasif V. Nabiyev, **Yapay Zeka, Problemler – Yöntemler – Algoritmalar**, Seçkin Yayıncılık, İstanbul, 2004, s. 38-59

- Beşeri Algılama Yeteneklerinin Simülasyonu (Görme, Konuşma, İşitme, Koklama vs.)
- Robotikler (Rutin, Kirli ve Tehlikeli İşler İçin Kullanılan Robotikler)
- Sinirsel Ağlar
- Bulanık Mantık
- Sanal Gerçeklik

3.4.2.5.1 Bilgi Tabanlı Yapay Zeka ve Uzman Sistemler

Bilgi tabanlı yapay zeka sistemi, belli bir uygulama alanına (bilgisayar onarımı gibi) ilişkin pratik çözüm veya yordamlama bilgilerinden (sezgi, yargı ve çıkarımlar) oluşmuş bir bilgi tabanına dayalı olarak çalışır. İnsanların kendilerine ait bilgi tabanı sistemindeki, eğer - o zaman (if-then) kurallarını kullanarak belirli sorunları çözme kabiliyeti, bu yapay zeka türüne ilham kaynağı olmuştur. Bilgi tabanlı sistemlerin en gelişmiş örneği uzman sistemlerdir. Belli bir soruna ilişkin uzmanlık bilgileri, bir uzman sistemin bilgi tabanına yerleştirildikten sonra, kullanıcıların bu bilgiden yararlanmak amacıyla, uzman sistemle kurduğu iletişim, bir uzman şahısla kurulan iletişimin bir benzeri olacaktır. Sorun çözülene kadar kullanıcı ile bilgisayar tabanlı uzman sistem arasında, karşılıklı soru-cevap türünde bir iletişim oluşur.

3.4.2.5.2 Doğal Diller

Doğal diller, son kullanıcının doğal dili ile (İngilizce gibi) bilgisayarla iletişim kurmasını sağlayan yazılımlar için, kullanılan isimdir. Doğal dil yazılımlarındaki nihai amaç, geleneksel program dillerinde kullanılan komutlara olan gereksinimi ortadan kaldırmaktır. Fakat halen uygulamada gelinen nokta tatmin edici düzeyde değildir. Şu anda piyasada kullanılan doğal dillerin çoğu, kullanıcının bir uzman sistem ya da veri tabanı ile iletişimini sağlamaktan öte bir fonksiyon görememektedir. Yine de, bilgi işleminin sınırlı olduğu bazı alanlarda, doğal dil uygulamasının oldukça başarılı olduğu gözlenmektedir. Örneğin, insan kaynakları ve satın alma ile ilgili araştırma ve rapor hazırlama faaliyetlerinde

dođal dil uygulaması oldukça gelişmiş düzeyde olup kullanıcı normal İngilizce konuşur gibi bilgisayar ile iletişim kurabilmektedir.

3.4.2.5.3 İnsan Algılama Yeteneklerinin Benzetimi

Bu yapay zeka türü, insani yeteneklerin benzetimi ile ilgili olup bilgisayar sistemlerini görme, işitme, konuşma ve hissetme (dokunma) yetenekleri ile donatma çabasındadır. Bu yapay zeka yeteneklerini, bugünün teknolojisini kullanarak belirli ölçüde gerçekleştirmek mümkün gözükmemektedir.

İnsan algılama yeteneklerine sahip bilgisayarlar, tıpkı insanlar gibi çevre ile iletişim kurma becerisine sahip olabilmektedir. Aşağıda buna bazı örnekler verilmektedir.

- **Konuşma ; Ses Cevap Üniteleri**

Şayet uluslararası veya şehirlerarası bir telefon numarasını aramışsanız, "Aradığınız numara kullanımda değildir" veya tam otomatik bir arabaya binmişseniz "Emniyet kemerinizi takınız" gibi sözel mesajlara şahit olmuşsunuzdur. Bu mesajlar konuşma makinelerinin ses cevap ünitelerinden gelen mesajlardır. İki ayrı ses cevap ünitesi türü vardır; birinci tür bir insan sesinin kaydını kullanırken diğeri bir konuşma synthesizer'ından yararlanır.

Birinci tür, kullanıcı tarafından kasete önceden kaydedilmiş kelime, cümle, müzik, alarm gibi kayıtlardan çıktığı seçer. Bu ses cevap ünitelerinde, sesin gerçek analog boyutları digital verilere çevirerek bir hafıza yongasına sürekli kullanılacak biçimde yüklenir. Çıktı alınırken ise, seçilen ses tekrar analog hale dönüştürülür. Bu tür yongalar, belirli kullanım alanları için seri üretim teknolojisiyle üretilmektedir. örneğin mikrofönlar, yangın alarm cihazları, asansörler, alarm saatleri, otomobil uyarı sistemleri, video oyunları gibi cihaz ve arabalarda kullanılan ses cevap üniteleri bu gruba girer.

Konuşma Synthesizer'ları ise, ham veriyi elektronik olarak üretilmiş konuşmalara dönüştürür. Bunun için de, bu cihazlar konuşmayı oluşturan temel ses birimlerine benzer sesleri bir arada kullanmaya çalışır. Bir konuşma Synthesizer'ı en az 64 temel sesi üretebilme kapasitesine sahiptir. Günümüz teknolojisi ile bunu sınırlı sayıda cümle için yapabilmek söz konusu ise de bu teknolojinin kullanım alanı gittikçe gelişmektedir. Örneğin, bugün okuyucu bir kitabı tarayarak ham verileri elde etmekte ve daha sonra konuşma synthesizer'ı bu ham verileri görme özürü insanların istifadesine sunmak için konuşma haline getirmektedir. Diğer bir uygulama ise, konuşma özürü çocuklar için geliştirilmiş sistemdir. Bu cihaz vasıtası ile bu çocuklar çevreleri ile konuşma imkanı elde edebilmektedir. Bu uygulama alanlarının daha da gelişeceği açıktır. Beklenildiğinin aksine, bu tür cihazlar nispeten ucuz olup, bu açıdan yakın gelecekte kişisel bilgisayarlarda da kullanım alanı bulacağı tahmin edilebilir.

- **İşitme, Ses Simülasyonu**

Bilgisayarların çok doğal olan sesleri yanlış algılaması olağan bir şeydir. Bununla birlikte, ses tanımanın birtakım uygulamaları da mevcuttur. örneğin, satış elemanlarının telefonla bilgisayarı arayıp, müşteri ve sipariş numaraları ile sipariş miktarlarını bilgisayara girmesi bugünün teknolojisi ile mümkündür. Ses tanımada, veri tabanının yaratılması sürecine eğitim denir. Çoğu ses tanıma sistemleri konuşmacıya bağımlıdır, yani, ancak belli konuşmacıların sesi sistem tarafından tanınabilmektedir. Bu nedenle, sistemi kullanan her kişi için ayrı bir kelime veri tabanının yaratılması gerekmektedir¹¹².

- **Görme, Görsel Sistemler Simülasyonu**

Görme, simülasyonu en zor olan insan algılama becerisidir. Örneğin, bir bilgisayarın bir insan gibi bir nesneyi görmesi ve onu yorumlaması mümkün değildir. Bilgisayara görüş kazandırmak için bir kameradan yararlanır. Kamera aracılığıyla veri tabanı yaratmak için gerekli olan girdiler elde edilir. Bir görsel sistem, kamera desteğiyle yorumlanması istenen nesnenin standart halini dijital hale getirir ve bu dijital hale getirilmiş nesnelerin görüntüsü veri tabanına yüklenir. Daha sonra dijital sistem çalışırken, kamera görüntüyü dijital çeviriciye

¹¹² <http://tip.erciyes.edu.tr/anabil/kbb/EGITIM/DERS/>, Erişim 10.01.2004

gönderir. Dijital hale getirilmiş bu görüntü bilgisayarın veri tabanındaki önceden kaydedilmiş dijital görüntüler ile karşılaştırılır. Bu kıyaslama neticesinde sistem nesneyi tanır.

3.4.2.5.4 Robotikler

Robotikler, bilgisayarlar ile endüstriyel robotların uyumlu bir bütünleşmesidir. Endüstriyel robotlara, bilgisayarlar yardımıyla herhangi bir rutin hareketin nasıl yapılacağını öğretmek mümkündür. Örneğin; araba boyama, vida sıkma, malzeme taşıma ve hatta kusurlu parçaları tespit etme gibi daha karmaşık davranışları yapan robotikleri günümüzde görmek mümkündür.

Yapay zekanın, en büyük ticari başarıyı elde ettiği alan robotik alanıdır. Genel inancın aksine, robotikler bilim-kurgu filmlerinde görülen robotlardan gerek görünüm gerekse işlev açısından oldukça farklıdır. Endüstriyel robotların, günümüzde en fazla kullanılanı ise bir bilgisayar tarafından kontrol edilen bir mekanik koldur. Manipülatör olarak da adlandırılan bu kol bir insan kolunun yapabileceği çoğu hareketi yapabilme becerisine sahiptir¹¹³.

Robotlara işin nasıl yapılacağı bilgisayar tarafından öğretilir. Bir bilgisayar programı ile robotları kontrol etmek mümkündür. Bu program robota hareketin zamanı, yönü, mesafesi gibi konularda komut veren bir programdır. Bir kere programlandıktan sonra, robotların hareketlerini kontrol etmeye fazla ihtiyaç yoktur. O artık işini büyük bir titizlikle herhangi bir şey talep etmeden (yeme, içme gibi) yapmaya devam edecektir.

Robotlar konusunda görülen bir diğer gelişme ise, robotlara bazı beşeri algılama becerisini yerleştirmektir. Daha önce açıklanan robotlar beşeri algılama becerilerine sahip olmadıklarından dolayı, ancak tekdüze işleri yapma becerisine sahiptirler ve bu yüzden de bu tür robotlara “seç ve yerleştir” robotları denmektedir. Eğer bu robotlara görme, işitme, konuşma gibi beşeri algılama becerileri kazandırılırsa, bu robotların insan gibi davranması ve böylece bu

¹¹³ A. DOĞAN, K.ALP, YAPAY ZEKA, Kariyer Yayıncılık, İstanbul, 2002, s.18-34

robotlara akıllı denmesi de mümkün olabilecektir. Bugünkü teknoloji ile bir robotu görsel bir alt sistemle teçhiz edip, robotun belli standarttaki bir nesneden farklı nesnelere ayırt etmesi sağlanabilir. Doğal olarak, görsel sistem teknolojisindeki gelişmeler devam ettiği müddetçe, robotların tıpkı bir insan gibi işyerinde dolaşması da mümkün olabilecektir.

3.4.2.5.5 Sinirsel Ağlar

Sinirsel ağlar, çeşitli yollarla birbirine bağlı birimlerden oluşmuş topluluklardır. Her birim iyice basitleştirilmiş bir nöronun niteliklerini taşır. Nöron ağları, sinir sisteminin parçalarında olup biteni taklit etmekte, işe yarar ticari cihazlar yapmakta ve beynin işleyişine ilişkin genel kuramları denemekte kullanılır. Sinirsel ağ içindeki birimler, her birinin belli işlevi olan katmanlar şeklinde örgütlenmiştir ve bu yapıya “yapay sinir ağı mimarisi” denir¹¹⁴.

Günümüzde sinirsel ağ uygulamaları, ya geleneksel bilgisayarlar üzerinde yazılım simülasyonları kullanılarak, veya özel donanım içeren bilgisayarlar kullanılarak gerçekleştirilmektedir. Kredi risk değerlemesinden imza kontrolü, mevduat tahmini ve imalat kalite kontrolüne kadar uzanan uygulamalar yazılım paketlerinden faydalanılarak yapılmaktadır.

3.4.2.5.6 Bulanık Mantık

Bulanık mantık (Fuzzy Logic) kavramı, ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A.Zadeh’in bu konu üzerinde ilk makalelerinin yayımlanmasıyla ortaya çıkmıştır¹¹⁵. Bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematiksel düzen olarak

¹¹⁴ Dany HAJJAR, Siman ABOURIZK, Kevin MATHER, Integrating Neural Networks With Special Purpose Simulation, **Proceedings of the 1998 Winter Simulation Conference**

¹¹⁵ Zekai ŞEN, **Mühendisler İçin Bulanık (Fuzzy) Mantık ile Modelleme Prensipleri**, Su Vakfı Yayınları, İstanbul, 2004, s.6

tanımlanabilir. İnsanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışması gereklidir.

Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek için veriler tam olmalıdır. Bulanık mantık, kişiyi bu zorunluluktan kurtarır ve daha niteliksel bir tanımlama olanağı sağlar. Bir kişi için 38,5 yaşında demektense sadece orta yaşlı demek bir çok uygulama için yeterli bir veridir. Böylece dikkate değer ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir.

Bulanık mantığın uygulama alanları çok geniştir. Sağladığı en büyük fayda ise “insana özgü tecrübe ile öğrenme” olayının kolayca modellenebilmesi ve belirsiz kavramların bile matematiksel olarak ifade edilebilmesine olanak tanınmasıdır. Bu nedenle lineer olmayan sistemlere yaklaşım yapabilmek için özellikle uygundur.

3.4.2.5.7 Sanal Gerçeklik

Sanal gerçeklik, bilgisayar ortamında oluşturulan bir gerçekliktir ve “cyberspace” olarak da bilinir. Yapay zekanın bu alanında doğal gerçekliğe uygun, insan/bilgisayar arabirimlerinin kullanıldığı bir ortam oluşturulur. Sanal gerçeklik, gözlük ve stereo kulaklıktan oluşan başlık seti, vücut hareketlerini algılayan özel bir giysi veya eldivenden oluşan, çok algılayıcı giriş-çıkış cihazlarına bağlı olarak oluşturulmaktadır. Böylelikle üç boyutlu sanal dünyayı görebilir ve dokunabilirsiniz. Sanal gerçeklik, sizin bilgisayar benzetimli nesnelere ve varlıklar ile etkileşim içine girebilmenize olanak sağlamaktadır¹¹⁶.

¹¹⁶ E. ALTUNTAŞ, T. ÇELİK; <http://members.tripod.com/~Bagem/bagem/yz4.html>, Erişim 20.01.2005

Sanal gerçeklik uygulamaları geniş bir alana yayılmıştır. Bilgisayar destekli tasarımda (CAD), tıbbi teşhis ve tedavide, fiziksel ve biyoloji bilimlerindeki bilimsel deneyimlerde, pilot ve astronotların eğitimi için uçuş simülatörlerinde ve eğlence olarak üç boyutlu video oyunlarında kullanılmaktadır. CAD en geniş şekliyle endüstriyel sanal gerçeklik uygulamalarında kullanılmaktadır. Mimarlar ve tasarımcılar, ürünlerin ve yapıların üç boyutlu modelleri üzerinde test ve tasarım işlemleri yapmakta kullanırlar. Bu teknoloji ayrıca ecza ve biyoteknoloji firmaları tarafından yeni ilaçların bilgisayarda tanımlanmış davranışlarını geliştirmek ve gözlemek için kullanılmaktadır. Ayrıca doktorlar hasta vücudunun sanal bir modelinin oluşturulup sorgulanmasında faydalanmaktadır¹¹⁷.

3.4.3 Nesneye Yönelik Programlama

Nesneye yönelik programlama yazılımının avantajı, kesikli olay simülasyonu ile ilişkilidir. Kesikli olay simülasyonunun varlıkları nesneye yönelik programlamanın nesnelere olarak ifade edilir. Her varlık iki tür bilgiye sahiptir: Bunlar durumsal ve davranışsal bilgidir. Nesnenin durumsal verisi varlığın nitelikleri olarak ifade edilir ve varlığın yeteneklerini mevcut durumunu ya da parametrelerini temsil eder. Nesnenin davranışsal verisi ise, belirli koşullar altında varlığın nasıl bir davranış gösterdiği hakkında bilgi verir¹¹⁸.

Nesneye yönelik teknoloji, sistemin modelini oluşturan varlıklar arasındaki etkileşimi kararlı hale getirir. Bunun yanında kullanıcının sistemin bütünlüğünü tehlikeye atmadan sistemi temsil eden modelin sadece belirlenmiş kısımlarında değişimler yaparak incelenmesine olanak sağlar¹¹⁹.

¹¹⁷ Hank GRANT, Chuen-KI LAI, Simulation Modeling With Artificial Reality Technology (Smart): An Integration Of Virtual Reality And Simulation Modeling, **Proceedings of the 1998 Winter Simulation Conference**

¹¹⁸ N.R. NIELSEN, Application of Artificial Intelligence Techniques to Simulation, **Advances in Simulation**, Vol.4 In P.A. Fishwick & R. Modjeski (Eds)

¹¹⁹ Paul A. FISCHWICK, **Simulation Model Design and Execution**, Prentice Hall, Newjersey, 1995, s.17-19

DÖRDÜNCÜ BÖLÜM

UYGULAMA :VESTEL ELEKTRONİK A.Ş. DE BİR ÜRETİM BANDI SİMÜLASYONU

Vestel Elektronik AŞ Televizyon üretim tesisinde, bazı ürünlerin şaselerinin üretiminde darboğazla karşılaşmıştır. Makina sayısı arttırılmadan bu sorun çözülmeye çalışılmaktadır. İlk etapta, tamir bakımdaki gecikmeler giderilerek üretimde artış sağlanması düşünülmüştür. Diğer bir alternatif ise metod etüdü çalışması ile hat besleme hızının arttırılmasıdır. Üretim tesisi, sürekli üretim yapmak zorunda olduğundan bu alternatiflerin tesiste fiziksel olarak denenmesi mümkün değildir. Bunun üzerine gerçek sistemin bilgisayarda simülasyonu gerçekleştirilerek sonuç alınmaya çalışılmıştır.

Bu tezde, sürekli çalışan fiziksel sistem yerine, yeni geliştirilen simülasyon dillerinden biri olan, görsel üstünlüğünün yanı sıra, SIMAN programlama mantığı esasına dayalı, ARENA SE VII kullanılarak gerçekleştirilen sistemin modeli üzerinde, öneriler denenerek çıktı miktarındaki artışlar araştırılmıştır.

Çalışmanın hedefi, çıktıyı artırabilmek için simülasyon ile darboğaz işlemi belirleyip, kuyruk sürelerini minimize etmek ve toplam sistem süresini azaltmaktır. Bunun yanı sıra incelenen üretim bandındaki arıza ve tamir istatistiklerini modele yansıtarak yeni tamir bakım planı oluşturulmasına yardımcı olmaktır. Bu amaçla, TV üretim tesisindeki TV montaj bandının otomatik dizgi bölümünde yoğun talebe göre aralıksız üretilen TV modellerinden birinde kullanılan AK19 kodlu şase üretimi analiz edilmiştir. AK19'un otomatik dizgi bölümünde üretim kapasitesini arttırabilmek için öncelikle metod etüdü yaparak önerilerde bulunulmuş, ikinci aşama olarak üretim bandı modellemesini yapabilmek için detaylı gözlem ve analizlere geçilerek çıkarılan önerilerin kurulan model üzerinde denenmesine geçilmiştir.

4.1 ŞİRKETİN TANITIMI

VESTEL Elektronik Sanayi ve Ticaret A.Ş. bir ZORLU Holding kuruluşudur. Faaliyet alanı Monitör ve TV üretimi olan Fabrika 94.000 m² si kapalı olmak üzere toplam 172.000 m² alan üzerine kurulmuştur. Ürünlerinin %75'ni Almanya, Fransa ve İngiltere başta olmak üzere dış pazarlara satmakta, %25'ni de iç piyasaya vermektedir.

Türkiye'nin tekstilden elektroniğe, enerjiden finans sektörüne kadar uzanan çeşitli alanlarda faaliyet gösteren, bünyesinde 46 şirketi ve 16000 çalışanı barındıran, yıllık dış ticaret hacmi 1,5 milyar dolara ulaşan Zorlu Holding'in temelleri 1950'li yılların başında Mehmet Zorlu tarafından atılmıştır. 1980'li yıllara dek tekstil sektöründe büyümesini sürdüren Zorlu grubu gittikçe diğer sektörlere de girmiş, asıl büyümesi ve bilinirliğinin artması ise son 10 yılda özellikle Vestel Elektronik'i şirketleri arasına katmasıyla gerçekleşmiştir. Holding 1996 ve 1997 yıllarında yatırımlarına devam ederek enerji ve finans sektöründe de şirketlerinin sayısını artırmıştır.

ZORLU HOLDİNG'e bağlı şirketler:

Tekstil Sektörü:

- Korteks Tekstil A.Ş.
- Zorluteks Tekstil Tic. Ve San. A.Ş.
- Zorlu Linen Pazarlama ve Dağıtım A.Ş.
- Zorlu Nakış A.Ş.
- Zorlu Polimer Sentetik İplik A.Ş.
- Zorlu Linen Dokuma Baskı ve Konfeksiyon A.Ş.
- Zorlu Ev Tekstil Ürünleri A.Ş.
- Zorlu Tekstil Ürünleri ve Ticaret A.Ş.
- Bel Air San. A.Ş.

- Zorlu Dokuma ve Ticaret A.Ş.
- Zorlu Gmbh

Elektronik Sektörü:

- Vestel Elektronik Sanayi ve Ticaret A.Ş.
- Vestel Bilgi Teknolojileri A.Ş.
- Vestel Beyaz Eşya A.Ş.
- Vestel Elektronik İletişim ve Bilgi A.Ş.
- Vestel USA
- Vestel Iberia
- Vestel France
- VESEG Gmbh

Finans Sektörü:

- Denizbank A.Ş.
- Deniz Yatırım A.Ş.
- Deniz Factoring
- Deniz Leasing

Enerji Sektörü:

Zorlu Enerji Elektrik Güç Üretimi Otomasyon A.Ş.

Diğerleri:

- Zorlu Dış Ticaret A.Ş.
- Zorlu Pazarlama ve Dağıtım A.Ş.
- Zorlu Sigorta A.Ş.
- Zorlu Havacılık A.Ş.
- Zorlu Turizm A.Ş.

Vestel Elektronik Sanayi ve Ticaret A.Ş. 1984 yılında Kıbrıslı bir işadamı olan Asil Nadir'in "Polypec International Co." şirketi tarafından kurulmuş, 1994 de Zorlu Holding bünyesine katılmıştır. Şirket kuruluşundan itibaren Ar-Ge çalışmalarına büyük önem vermiş, yatırımlarının önemli bir bölümünü yeni projelere ayırmış, sonuçta Amerika'da Dünya Elektronik devlerinin bulunduğu "Slikon Vadisi"ne girme başarısını göstermiştir. Vestel A.Ş. üretiminde kendi tasarladığı ürünlerinin müşterilerin istek çeşitliliğine hızlı yanıt verebilecek şekilde esnekliği sağlayabilmiştir. Bu başarısı ile zaman zaman iç piyasada liderliği yakalamış, teknolojik yenilikleri dünya ile aynı zamanda izleyerek çok kısa sürede tasarladığı projeleri üretime ve ihracata hazır hale getirerek, başta Avrupalı müşterilerin taleplerini karşılayacak dizayn ve üretim becerisini elde etmiştir. Bunun sonucu olarak da kalkıştığı ihracat atağında Avrupa ağırlıklı olmak üzere Amerika'dan Japonya'ya uzanan bir coğrafya da Vestel ya da değişik markalar altında ürettiği uluslararası kalite belgeli ürünleri ile dünyanın sayılı elektronik şirketleri arasında yerini almıştır.

Başarısının temelinde müşteri isteklerine hem hızlı tasarım hem de hızlı kapasite artışı ile yanıt verme becerisi olan Vestel Elektronik A.Ş. kalite düzeyini de git gide artırarak iki kez darbe gördüğü ekonomik krizlerde dahi ihracat gücü sayesinde pazar payını genişleterek ayakta durmayı başarmıştır.

Manisa organize sanayi bölgesinin bir başka bölgesinde 2003 yılında faaliyete geçen High End tesislerinde, farklı teknolojiye sahip plazma ve projeksiyon Televizyonlar üretilmektedir. Uygulamanın yapıldığı Vestel Elektronik A.Ş.'de yalnızca renkli TV üretmekte olup, yıllara göre TV üretim kapasiteleri Tablo 11' de görülmektedir.

İnsan Kaynakları: Şirketin 2650 çalışanını temelde aşağıdaki şekilde sınıflandırabiliriz.

- Beyaz Yakalı Personel: 400 çalışan
- Mavi yakalı Personel: Direkt İşgücü 1500, Endirekt İşgücü 750 çalışan şeklindedir.

Tablo 11 Renkli TV Üretim Miktarları (Adet)

YILLAR	ÜRETİM
1994	500.000
1995	800.000
1996	1.000.000
1997	2.000.000
1998	3.100.000
1999	3.900.000
2000	5.100.000
2001	5.000.000
2002	6.500.000
2003	8.000.000

Mavi yakalı personeli üretim bölümlerine göre ayırırsak ;

- Son Montaj : 668 çalışan
- Otomatik Dizgi : 97 çalışan
- Manuel Dizgi : 556 çalışan
- Plastik Enjeksiyon : 260 çalışan
- Strafor Üretim : 33 çalışan
- Boyahane+Serigrafi : 154 çalışan

İşletmede 8 saatlik 3 vardiyalı çalışma sistemi uygulanmaktadır. Vestel İnsan Kaynakları ve İdari İşler Bölümleri diğer bölüm yöneticileri ile koordinasyon içinde çalışarak, işletme içinde çalışan her kademedeki personelin görev ve sorumluluklarını en iyi şekilde yerine getirebilmesi için gerekli eğitim, teknik, mesleki bilgi ve yeteneklerinin geliştirilmesi çağdaş kalite kültürü ve kaliteyi sağlayıcı bilgi ve tekniklerin öğrenilmesi, çevre bilincinin oluşturulması ve geliştirilmesi, yönetici olanlar için yöneticilik yeteneklerinin olgunlaştırılması amacıyla yurt içi ve yurt dışı eğitim imkanları araştırır, eğitim planları hazırlar ve uygular. Ayrıca firma içinde özel prosesler söz konusu olduğunda, ihtiyaca göre belirli esaslar çerçevesinde kalifiye işgücü sağlar.

Muhasebe ve Finansman : Geçmiş yıllarda gerçekleşen satışlardan hareketle üretim planlama'dan gelen talep projeksiyonları doğrultusunda, yıl içindeki ekonomik şartlar, öngörülen durum ve şartlar, faiz oranları, döviz kurları gibi birçok kritere dayalı yıllık bütçeyi hazırlar. Türkiye'nin içinde bulunduğu siyasi ve ekonomik yapıdan dolayı bütçe son yıllarda haziran ayında revize edilmektedir. Bir sonraki yılın bütçesi ağustos ayı sonunda hazırlanmaya başlanır. Faiz oranları, döviz kurlarının gelişimi ve benzeri ekonomik göstergeler için İstanbul merkezde bulunan Mali Koordinatörlük ve grubun finans sektöründeki ayağı Denizbank'tan da yararlanılmaktadır.

Yatırım konusunda göz önüne alınan ana kriter üretim planıdır. Üretim planı; geçmişe ait veriler, alınan siparişler ve geleceğe yönelik tahminler ışığında hazırlanır. Bu veriler doğrultusunda bir sonraki bütçe döneminde ne kadar ürün üretilmesi gerektiği konusunda hedefler belirlenir. Bu hedeflere dayalı verilecek yatırım kararı iki şekilde ortaya çıkabilir:

1. Mevcut olan kapasiteyi arttıracak yeni yatırımlar yapmak
2. Mevcut kapasiteyi koruyup sistemin daha etkin kullanımını sağlayacak yatırımlar yapmak.

Yatırım kararı verildikten sonra bu karar Yönetim Kuruluna sunulur. Yapılan oylama sonucu eğer yatırım yapılabilir kararı çıkarsa gerekli teknik ve mali işlemlere geçilir. Eğer söz konusu yatırım Devlet teşviği çerçevesinde ise gerekli hukuki prosedürlerle Devlete yatırım hakkında bir dosya hazırlanır.

Pazarlama ve Üretim Planlama: İstanbul'daki Vestel Pazarlama A.Ş. tarafından yurt içi, Vestel Dış Ticaret A.Ş. tarafından da yurtdışı pazarlar için geçmiş verilerin analizi, cari dönem siparişleri ve mevsimsel etkiler de göz önüne alınarak talep tahminleri yapılır. Talep tahminlerinde sezonluk etkiler çok fazladır. Örneğin Temmuz-Ekim, Şubat-Mart aylarında satışlar nispi olarak artmaktadır. Ayrıca dünyadaki, Avrupa Şampiyonası, Dünya Kupası gibi organizasyonlar direkt olarak satışları olumlu yönde etkilemektedir. Vestel Pazarlama A.Ş. ürünlerin yurt içi piyasaya satışından sorumludur. Bayileri

aracılığıyla bölgeler bazında siparişler alır. Gereken zamanda reklam kampanyaları öngörür. Vestel Dış Ticaret ise yurt dışında pazar bulmakta ve dış satış tahminleri yapmaktadır. Vestel markalı ve fason olarak değişik marka adları altında üretilen ürünler için siparişler alır. Üretim planlamaya bildirir. Üretimin ardından ihracat prosedürlerini yerine getirir. Vestel Dış Ticaret A.Ş.'nin yurt dışında herhangi bir başka marka yaklaşımı yoktur. Bu durumun en büyük avantajı, reklam giderlerinin olmayışıdır. Firma kendi adını taşımayan ürünlerin reklamını yapmaya ihtiyaç duymamaktadır.

Üretim Planlama bölümünde toplanan siparişlerin üretiminin ne zaman ve nasıl yapılacağı, gerekli hammadde ve komponentlerin nitelik ve nicelik olarak nasıl temin edileceği, stok planlamasının nasıl olması gerektiği, üretimin hangi banda ne kadar yükleneceği, gereken personel sayıları gibi parametrelere yanıtlar veren programlar hazırlanır.

İşletmede aktif olarak çalışan 9 adet son montaj bandı vardır. Bunlardan 4 tanesi büyük ekran TV'ler (29", 32", 33"), 3 tanesi hem büyük hem de küçük ekran TV ler (25", 28", 29", 20", 21"), 2 tanesi de 14" TV üretiminde kullanılmaktadır. Bu bantlardan en hızlı işleyeni en küçük ekranlı TV'lerin üretildiği 14" bantları , buna karşın en yavaş çalışanı da büyük ekran TV'lerin üretildiği bantlar olmaktadır. Aylık üretim hedefinden hareketle günlük üretim planları düzenlenir. Bu planlarda normal çalışma temposu esas alınarak kapasite tespit edilir. Eğer kapasite talebi karşılayamıyor ise o zaman fazla mesai alternatifi göz önüne alınır. Yine kapasite yetmiyorsa vardiya artırımı kararı verilir. Eklenen vardiyalı çalışma sonucunda da sipariş karşılanamıyor ise siparişin süresi uzatılmaya çalışılır.

Kalite Kontrol : Vestel Elektronik A.Ş.'nin kalite politikası tanıtım kitabında şöyle özetlenmektedir:

"Vestel ailesi olarak temel ve kalıcı amacımız, organizasyon kültürümüzü Toplam Kalite Yönetimine uygun hale getirerek; rekabet gücümüzü ve karlılığımızı arttırmaktır.

Buna baęlı olarak ; müşteri ve pazar ihtiyacına duyarlı, üstün kalite ve değerde ürün ve hizmeti, rakiplerimize göre daha düşük maliyetle, zamanında ve sürekli olarak sağlamayı hedefliyoruz.

Bütün bunları zamanı ve yetenekleri en verimli şekilde değerlendirerek gerçekleştirebileceğimiz bilincinden yola çıkarak, temel amacımız;

En iyi insanları şirketimize çekerek, yaratıcılığımızla farklılık oluşturmak, çalışanlarımızın sürekli gelişmelerine olanak sağlayarak verimliliklerini arttırmak, işbirliği dayanışma ve güvene dayalı bir çalışma ortamı yaratmak Vestel'in her koşulda ve her zamanda devamlılığını sağlamak için seçtiğimiz yoldur."

İşletme ISO 9001 kalite belgesine sahiptir. Kalite politikasından da anlaşılacağı gibi üretimin ilk aşaması olan hammadde ve komponentlerin temininden başlamak üzere, TSE ve ISO 9001 standartlarına göre üretimin her aşamasında sürekli kontroller yapılır. İşletmede bilgi ve mal akışı söz konusu standartların belirlediği çerçeve doğrultusunda kesintisiz ve tüm personel tarafından paylaşımlı olarak gerçekleştirilmektedir. Şirket içi haberleşme, bilgi akışı ve toplantı duyuruları bilgisayar ağı üzerinden yapılmaktadır.

Bakım Onarım: Sistem üzerinde tanımlanmış dört tip bakım vardır:

1. Planlı Bakım : Haftalık, aylık, üç aylık, altı aylık ve yıllık periyotlarla makinelere bakım yapılmasıdır. Her yılın aralık ayında tüm makineler için periyodik bakım planı hazırlanır.
2. Arıza Üzerine Bakım: Bir makine arızalandığında operatör postabaşına bildirir. Postabaşı bilgisayar üzerinden ilgili birimleri bilgilendirir. Arızanın giderilmesinden sonra, harcanan zaman ve malzeme ile ne yapıldığı arıza bildirim formuna yazılır.
3. İstek Üzerine Bakım: İş istek formu doldurularak bakım bölümüne gönderilir. Değerlendirilen formlara göre gerekli bakım yapılır.
4. Operatör Bakımı: Her çalışan çalıştığı yerin temizliğinden sorumludur. Operatörler tarafından yapılabilecek bakım, yağlama gibi basit ayarlar üretime geçmeden yapılır.

Satış Sonrası Hizmetler: ISO 9001 Kalite belgesi çerçevesinde satış sonrası hizmet faaliyetleri için önceden belirlenen yükümlülüklerin oluşması durumunda yurt içi servis prosedürlerine uygun hareket edilir.

Vestel satış sonrasında;

Müşteri Hizmetleri ve Servis Müdürlüğü sorumluluğunda ;

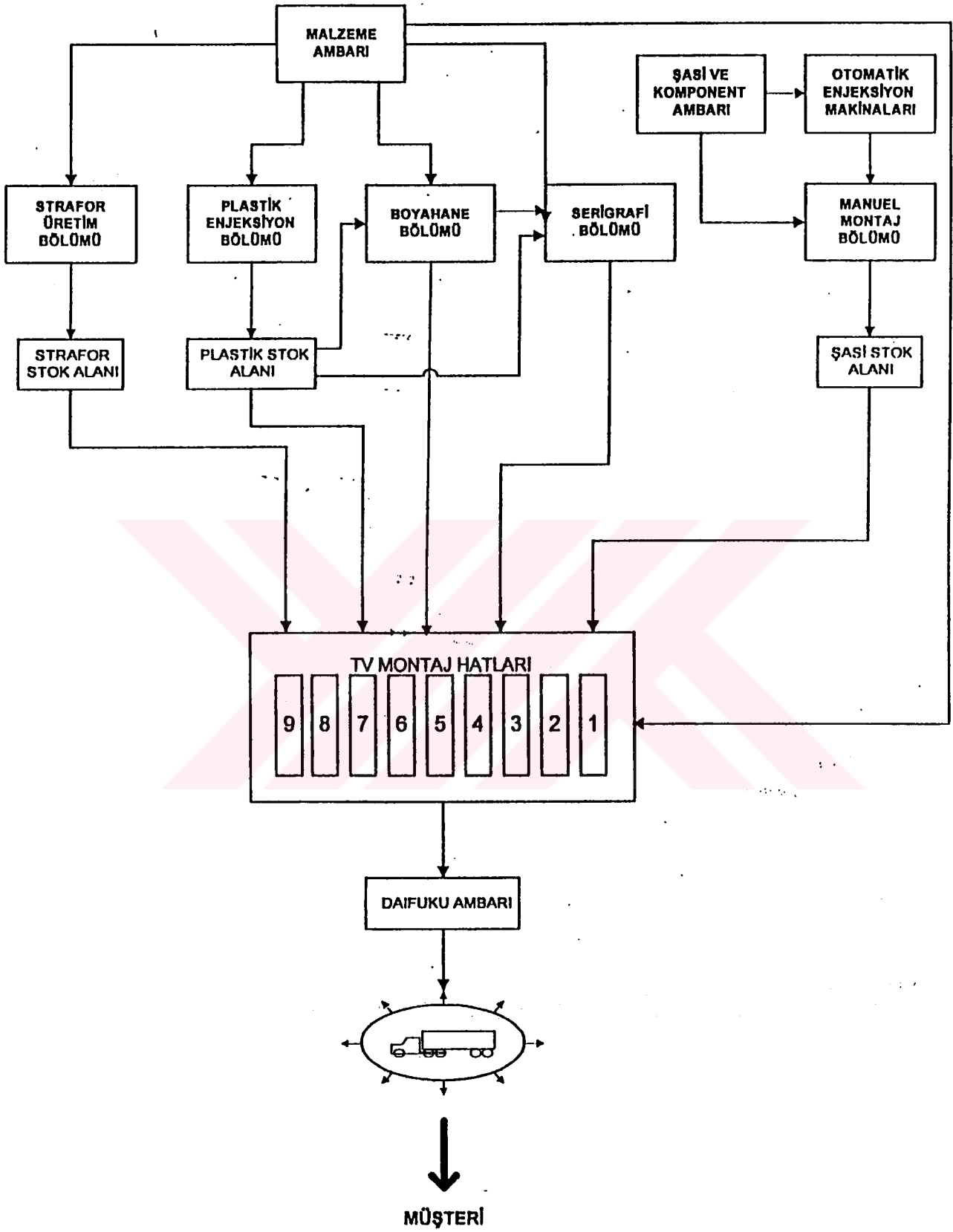
- Yurt içi servis ve yurt dışı müşterilerin istediği yedek parçaların gönderilmesi,
- Yurt içi servislerin eğitimi ve bilgilendirilmesi,
- Yurt içi müşteri şikayetlerinin değerlendirilmesi, çözümü ve geri bildirim gerçekleştirilmesi,
- Müşteri Hizmetleri ve Servis Müdürü gerekli durumlarda Müşteri Teknik Destek Biriminden destek ister.

Yurt dışı ve yurt içi müşteri şikayetlerinin teknik olarak değerlendirilmesi, çözümü ve geri bildirim Müşteri Teknik Destek Biriminin sorumluluğundadır. Bu birim müşterilerden gelen talepleri veri tabanına kayıt eder, gerektiğinde satış ve pazarlama uzmanları ile periyodik toplantılar düzenleyebilir.

4.2. ÜRETİM HATTININ TANITIMI

Vestel Elektronik A.Ş. de TV üretim akış şeması Şekil 9'daki gibidir. TV üretimi mamulü oluşturan farklı nitelikteki bölümlerde başlar ve son montaj aşamasında yarı mamul halindeki bölümler bir araya getirilir. Plastik Enjeksiyon bölümünde TV kasası için Plastik Enjeksiyon makineleri kullanılarak plastik hammaddesi işlenir ve üretim planları doğrultusunda istenen nitelik ve boyutlarda TV kasaları enjeksiyonla elde edilir, Serigrafi bölümü bu kasalara sıcak/soğuk baskı ile logo basımından, garanti belgesi, etiketler ve kullanım kılavuzuna kadar bir çok malzemeyi hazırlar. Boyahane bölümünde yapılan boyama işlemi ile kasa kısmı son montaja hazır hale gelir.

TV MONTAJ ÜRETİM AKIŞI



Şekil 9 TV Üretim Akış Şeması

TV üretiminin belkemiğini elektronik şase üretimi oluşturmaktadır. Müşteri beklentileri doğrultusunda Ar-Ge bölümündeki çok sayıda elektronik, bilgisayar ve makine mühendislerinin koordineli çalışmaları sonucunda TV'nin beyni kabul edilen, TV'nin tüm fonksiyonel özelliklerini bünyesinde barındıran şase planları, hangi modellerin hangi elemanlardan oluşacağı büyük bir titizlikle hazırlanır. Bir araya gelerek elektronik devreleri oluşturan chip, entegre, transistör, kondansatör, diyot, tristör ve dirençlerden oluşan bu elemanlara genel olarak komponent denmektedir. Kullanılan komponentlerin büyük kısmı yurt dışından ithal edilmektedir. Gelen komponentlerin giriş kalite kontrol onayını almasıyla şase üretimi başlar ve bunlar otomatik dizgi bölümünde şaselere çakılma işlemine geçer. Otomatik dizgi bölümünde söz konusu şaselerin önemli sayıdaki parçası çok hızlı ve hatasız bir biçimde tam otomatik olarak çakılır. Ardından manuel dizgi ile otomatik dizgi makinelerine giremeyen elemanlar elle ve bilgisayar kontrollü cihazlarla yerlerine yerleştirilir. Tüm otomatik ve manuel dizgi işlemleri boyunca özel bilgisayarlı cihazlarla insan gözünün takip edemeyeceği elektronik akışlar için sürekli kontrol işlemleri yapılmaktadır.

Böylece son montaj hattına gelindiğinde bir yanda kasalar hazır beklerken, diğer yanda dizimi tamamlanmış şaseler hazırdır. Ara stok alanında da ürünün niteliğine uygun TV tüpleri giriş kalite kontrol işlemlerinin ardından montajı beklemektedir. Bu aşamada TV kasalarına TV tüpleri şaseler ve gerekli diğer malzemeler monte edilerek, kasa kapatılır. Strafor üretim tesisinde, hammadde gerekli işlemlerden geçerek strafor elde edilir. Straforlar üzerine montaj bandının sonunda TV'ler gerekli dokümanlarla birlikte yerleştirilir, uzaktan kumanda cihazı da eklenerek ambalajlanır ve depoya alınır.

4.3 PROBLEMİN TANITIMI

Bu çalışma genel üretim hattının Otomatik Dizgi bölümünde gerçekleştirilmiştir. Otomatik Dizgi bölümü ürünlerdeki şaseler üzerinde bulunan ve ürünün çalışmasını sağlayan temel elektronik elemanlar olan binlerce küçük

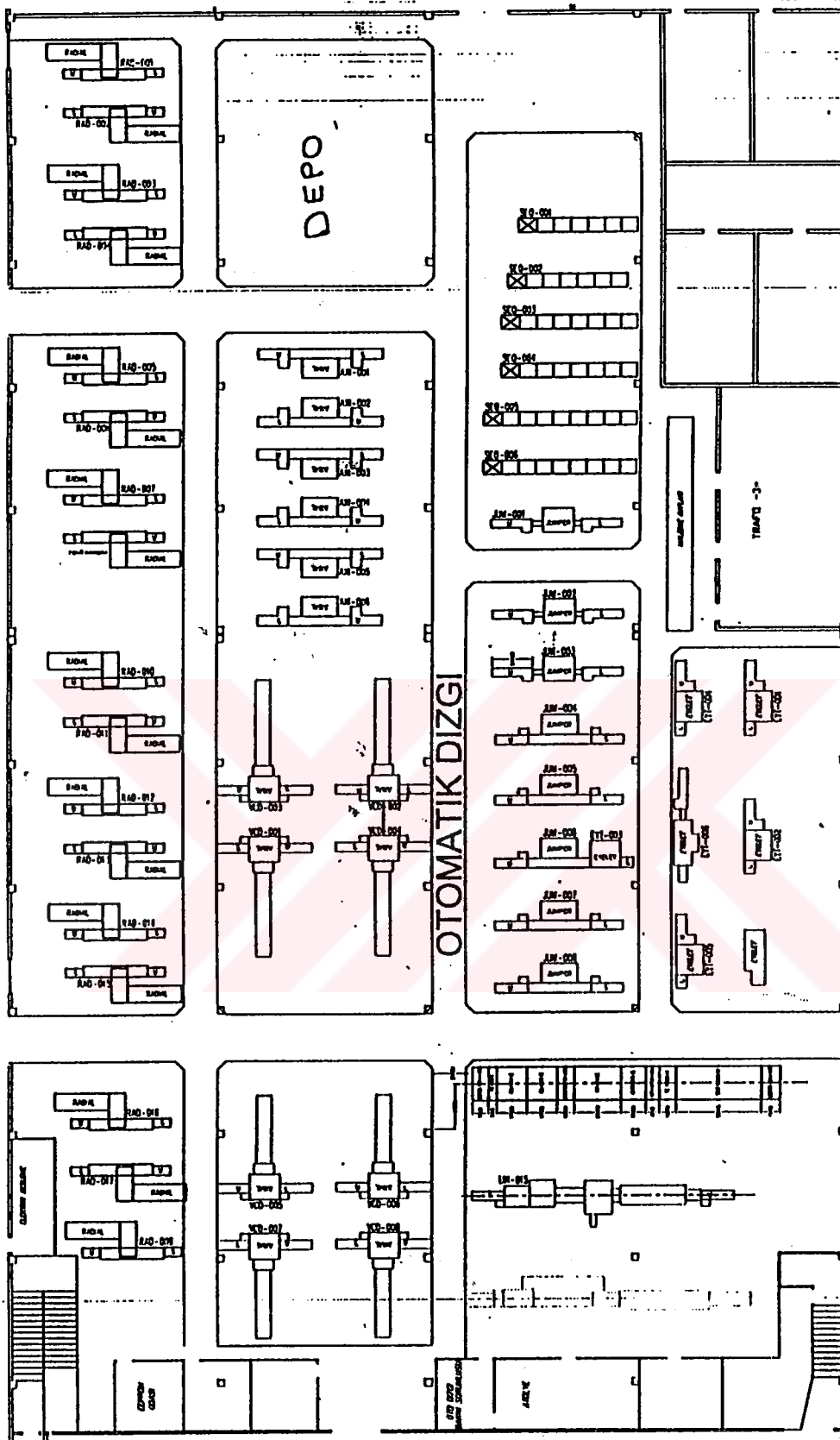
komponentleri şaselere çakan (monte eden) makinelerden oluşur. Otomatik dizgi bölümündeki makine yerleşim planı Şekil 10'da ve iş akışı da Şekil 11'de verilmiştir.

Fabrikada sipariş miktarı yüksek olan bazı ürünlerin şaselerinin üretiminde darboğaz bulunmaktadır. Yönetim, yeni makine yatırımı yapmaksızın ya da her ürün için üretim planlarında belirtilen, işe atanan makine sayısını değiştirmeden çıktı miktarını arttırmayı istemektedir. Dolayısıyla buradaki en önemli sorun yeni makine almadan, metod mühendisliği çalışmaları ile çıktı miktarının artırılmasıdır.

Bu çalışmanın temel amacı, simülasyon analizi yardımıyla metod mühendisliği'nin önerdiği yeni durumların, çıktı miktarında meydana getireceği değişikliğin öngörülenmesidir. Bu çalışmada, ürün çeşitleri arasında sipariş miktarı yüksek olan TV modellerinden birine ait AK19 kodlu şase üretimi analiz edilmiştir.

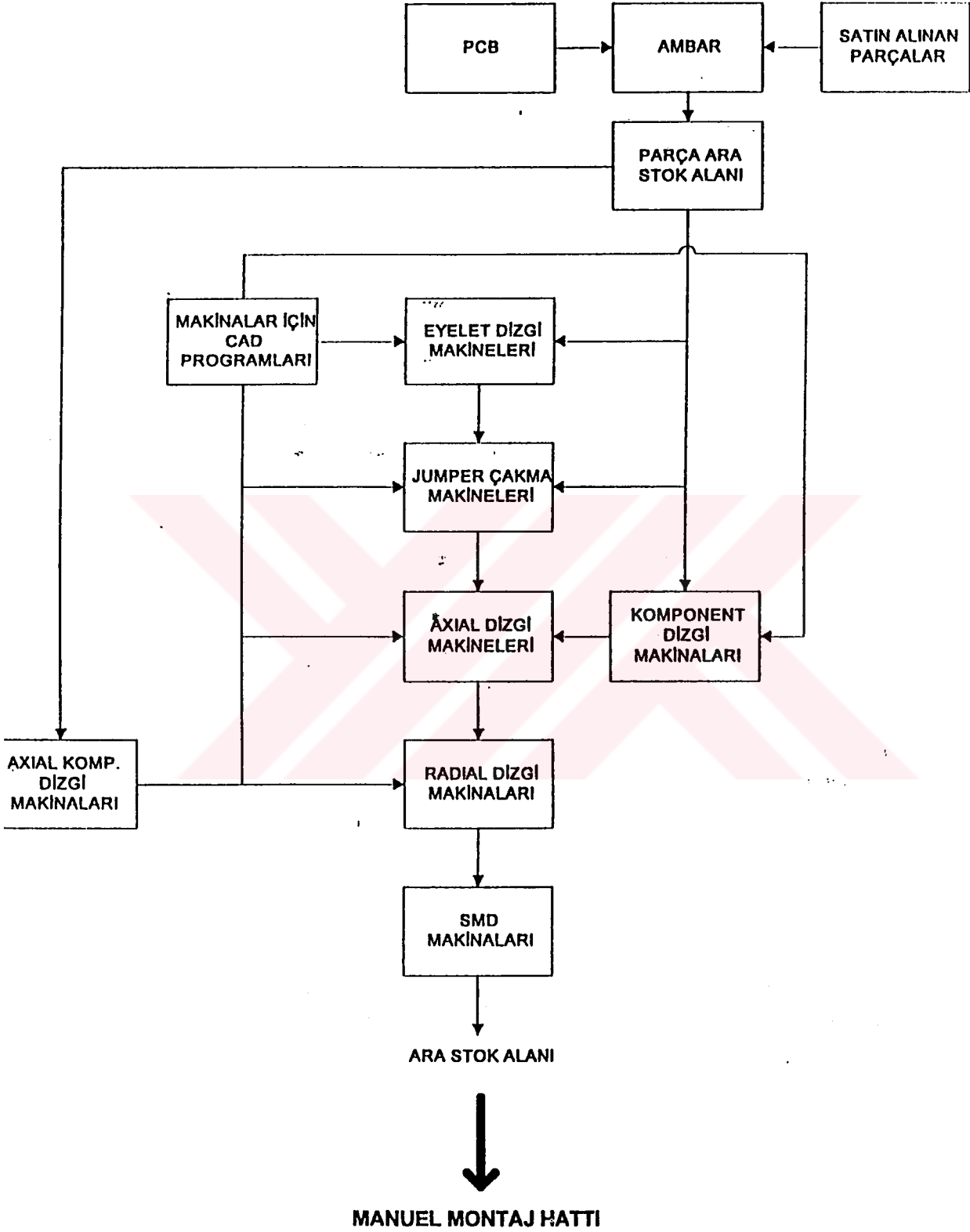
Üretim hattındaki makine grupları arasında üretim planlarına bağlı olarak yarı mamul stoku kullanarak ya da kullanmadan aynı anda birden fazla ürünün üretimi devam ettiğinden bu çalışmada söz konusu ürünün atandığı makineler incelenmeye esas alınmıştır.

Bu amaçla ürünün iş akışını oluşturan mevcut durum gözlemlenerek modellenmesi ortaya konmuş, daha sonra bu model üzerinde planlanan düzenlemeler yapılarak önerilen durumlardaki çıktı miktarları ARENA 2.2 simülasyon programıyla analiz edilmiştir.



Şekil 10 Otomatik Dizgi Bölümündeki Makine Yerleşim Planı

OTODİZGİ BÖLÜMÜNDE ÜRETİM İŞLEMİ



Şekil 11 Otomatik Dizgi Bölümünde İş akışı

Çalışmanın uzun dönemdeki hedefleri de şu şekildedir:

- Makinelerin kuyruk analizi yapılarak gelecekteki makine yatırımı projeksiyonuna ön bilgi oluşturmak.
- Önerilen değişiklikler ile hızlanacak olan üretim hattı için yeni bir hammadde (boş şase) besleme hızı belirlemek
- Yeni dönemdeki bakım planlaması için ön bilgi elde etmek

Otomatik Dizgi bölümü makine parkında şu makineler bulunmaktadır.

Perçin Makineleri:

5 adet Universal ve 1 adet Panasonic olmak üzere toplam 6 adet perçin makinesi vardır. Bu makineler elektronik devreleri birbirine bağlayan ve perçin olarak isimlendirilen son derece küçük elemanları şaseye yerleştirir. Yükleme, boşaltma ve dizgi olmak üzere üç üniteden oluşmaktadır.

Jumper Makineleri:

Bu makineler de Perçinler gibi, şaseler üzerinde belirli elemanları birbirine bağlayan ve yükleme, boşaltma, dizgi ünitelerinden oluşan özel cihazlardır. 5 adet Universal, 3 adet Panasonic Jumper Makinesi vardır. Bu uygulamada ele alınan AK 19 kodlu şase, sadece 6 Jumper makinesinden geçmektedir.

Sequencer Makineleri:

6 adet Universal Sequencer makinası vardır. Bu makineler axial dizgi malzeme hazırlığında kullanılmaktadır.

Axial Makineleri:

6 adet Universal Axial makinesi vardır. Şaseler üzerinde genellikle yatay durumda konumlanan komponentleri şaseye çıkar.

Sequencer-Axial Makineleri:

6 adet Universal Sequencer-Axial makinesi vardır. Şaselerin içerdiği komponentlerin niteliğine göre bazen makineler ayrı ayrı Sequencer ve Axial makinelere girmek yerine sadece bu özel makinelerde işlenmektedir. İncelediğimiz ürün bu makinelere uğramamaktadır.

Radial Makineleri

17 adet TDK Radial makine vardır. Şaseler üzerinde genellikle dikey konumlanan komponentleri şaseye çıkar. Seçilen ürünün üretimine 8 Radial makinesi atanmıştır.

SMD Makineleri

9 adet Panasonic, 3 adet Fuji SMD makinesi vardır. Ana şase dizgisinde kullanılan bu makineler yükleme, yapıştırma, dizgi, ve boşaltma ünitelerinden oluşmaktadır. Seçilen ürüne 9 adet SMD makinesi atanmıştır.

Otomatik Dizgi bölümündeki makinelerde çalışan her personel sorumlu olduğu makinenin çalışmasını yönlendirir, malzeme takibini yapar ve malzeme yokluğundan doğabilecek aksaklıkların önüne geçmek için makine "malzeme az" uyarısı verdiğinde hemen yeni parti malzemeyi makineye getirerek yüklemesini yapar. Makinede çalışan personel kendisinin çözemeyeceği bir arıza durumunda postabaşına haber vererek bakım elemanının gelmesini ister.

4.4 MODELİN HAZIRLANMASINDA YAPILAN VARSAYIMLAR

Modelin hazırlanması sırasında sistem ile ilgili bazı varsayımlarda bulunulmuştur:

- Sistemdeki malzeme aktarımları çok kısa mesafeler içinde ve hızlı bir şekilde gerçekleştiğinden dolayı, modelde taşıma ile ilgili süreler dikkate alınmamıştır.

- Tüm ürünlerin üretim süreleri çok yakın olduğundan ve ayar süreleri kısa sürdüğünden dolayı sadece tek bir ürünün üretildiği varsayılmıştır.
- Modelde makinelerin arızalar arası ve tamir süreleri, bakım kayıtlarından elde edilmiştir.
- Depolama alanı yeterli olduğundan, makinelerin kuyruklarının sonsuz olduğu varsayımı yapılmıştır.

4.5 MODELİN KURULMASI

Simülasyon modeli ARENA 2.2 Programı ile hazırlanmıştır. Modelin ana hatları şöyledir:

Mevcut durum ve önerilen durumlar için farklı olmak üzere, şaseleri temsilen modele sabit bir hızda giren varlıklar yaratılır (CREATE). Daha sonra varlıklar ilk istasyon bloğunun kuyruğuna girer (ENTER). Seçici kuyruk (PICKQUEUE) ile izleyen makine grubu arasında kuyrukta bekleyen varlık sayısı en az olan makineye gönderilir. Belirlenen işlem süresi kadar işlem gören (SEIZE) varlıklar, bir sonraki makine grubunun kuyruğuna gönderilir. Ardışık işlemler boyunca kısa kuyruk parametresine göre işlem görerek en son makine grubundan çıkan varlıklar, sistemden çıkartılır (DISPOSE).

Sistemin akış şeması şekil 12'de görülmektedir.

4.6 MODELDE KULLANILAN ARENA ELEMANLARI

1. Süreçler (Processes)

Modelde, her bir makine bir süreç olarak modellenmiştir.

Perçin1, Perçin2, Perçin3, Perçin4, Perçin5, Perçin6: Bu süreçler perçin çakma operasyonunu belirtir.

Jumper1, Jumper2, Jumper3, Jumper4, Jumper5, Jumper6: Bu süreçler jumper çakma operasyonunu belirtir.

Sequencer1, Sequencer2, Sequencer3, Sequencer4, Sequencer5, Sequencer6: Bu süreçler Sequencer sarma operasyonunu göstermektedir.

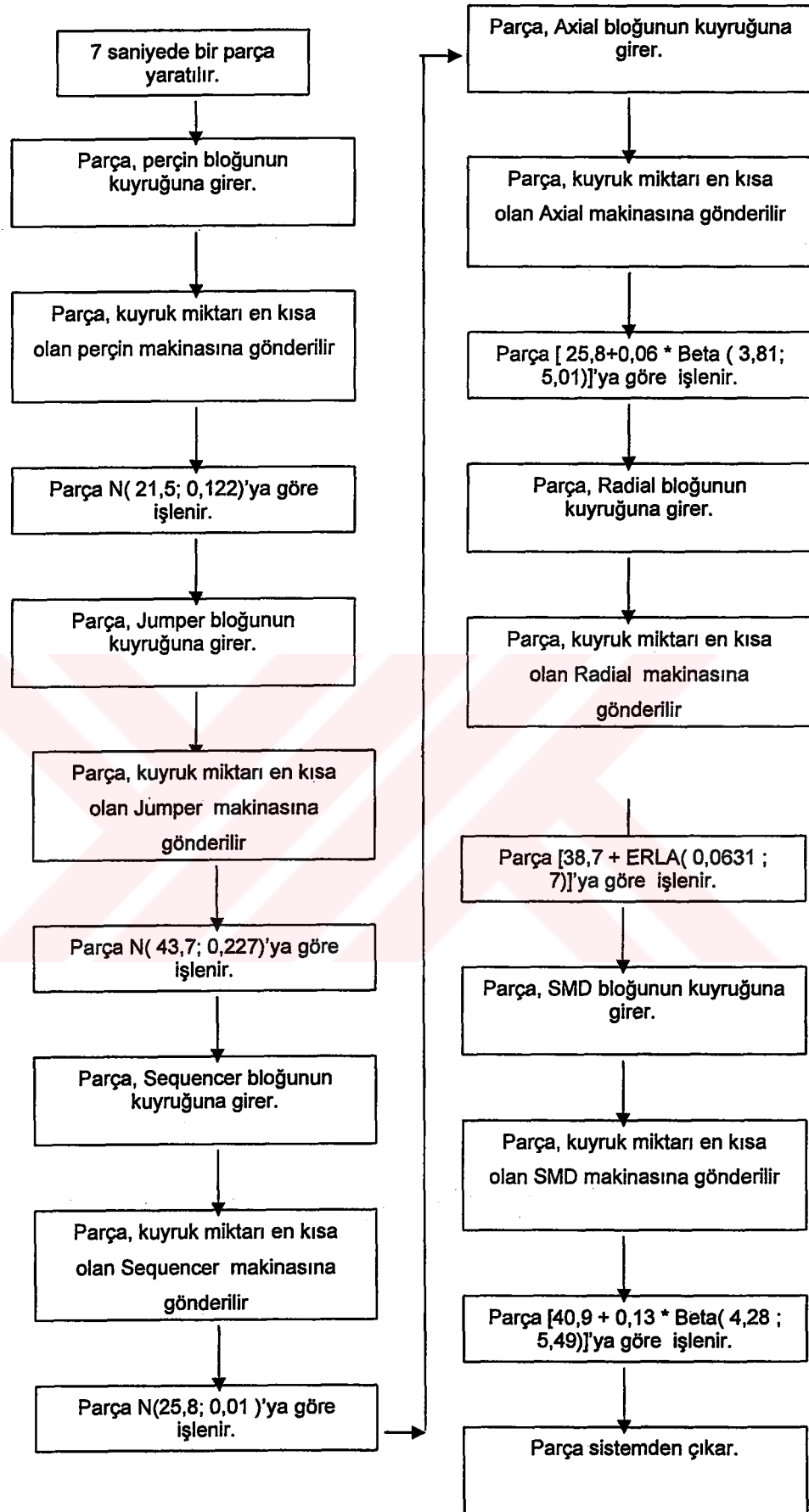
Axial1, Axial2, Axial3, Axial4, Axial5, Axial6: Bu süreçler Axial çakma işlemini göstermektedirler

Radial1, Radial2, Radial3, Radial4, Radial5, Radial6, Radial7, Radial8: Bu süreçler Radyal çakma işlemini göstermektedir.

SMD1, SMD2, SMD3, SMD4, SMD5, SMD6, SMD7, SMD8: Bu süreçler SMD çakma işlemini göstermektedir.

2. Kuyruklar

Modelde, her bir makine grubu için bir ortak kuyruk ve her bir makine için kendi ayrı kuyruğu bulunmaktadır. Makine grubuna gelen varlıklar önce grubun ortak kuyruğuna girmekte, daha sonra kuyruk miktarı en az olan makineye gönderilerek onun kuyruğuna dahil olmaktadır



Şekil 12 Simülasyon Akış Diyagramı

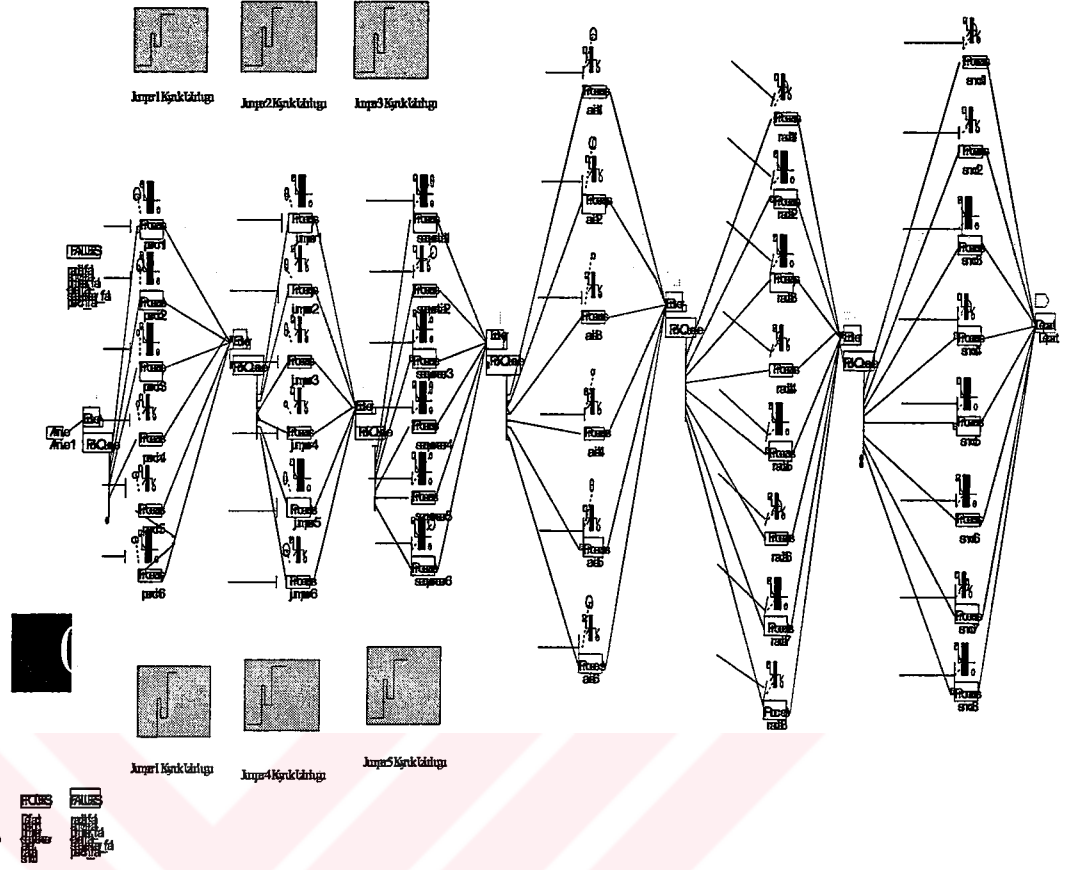
Şekil 13 ARENA modeli ile genel iş akışını yansıtırken; Şekil 14 , Şekil 15 ve Şekil 16'da modeldeki her makine grubunun şekilleri görülmektedir.

3. Arızalar

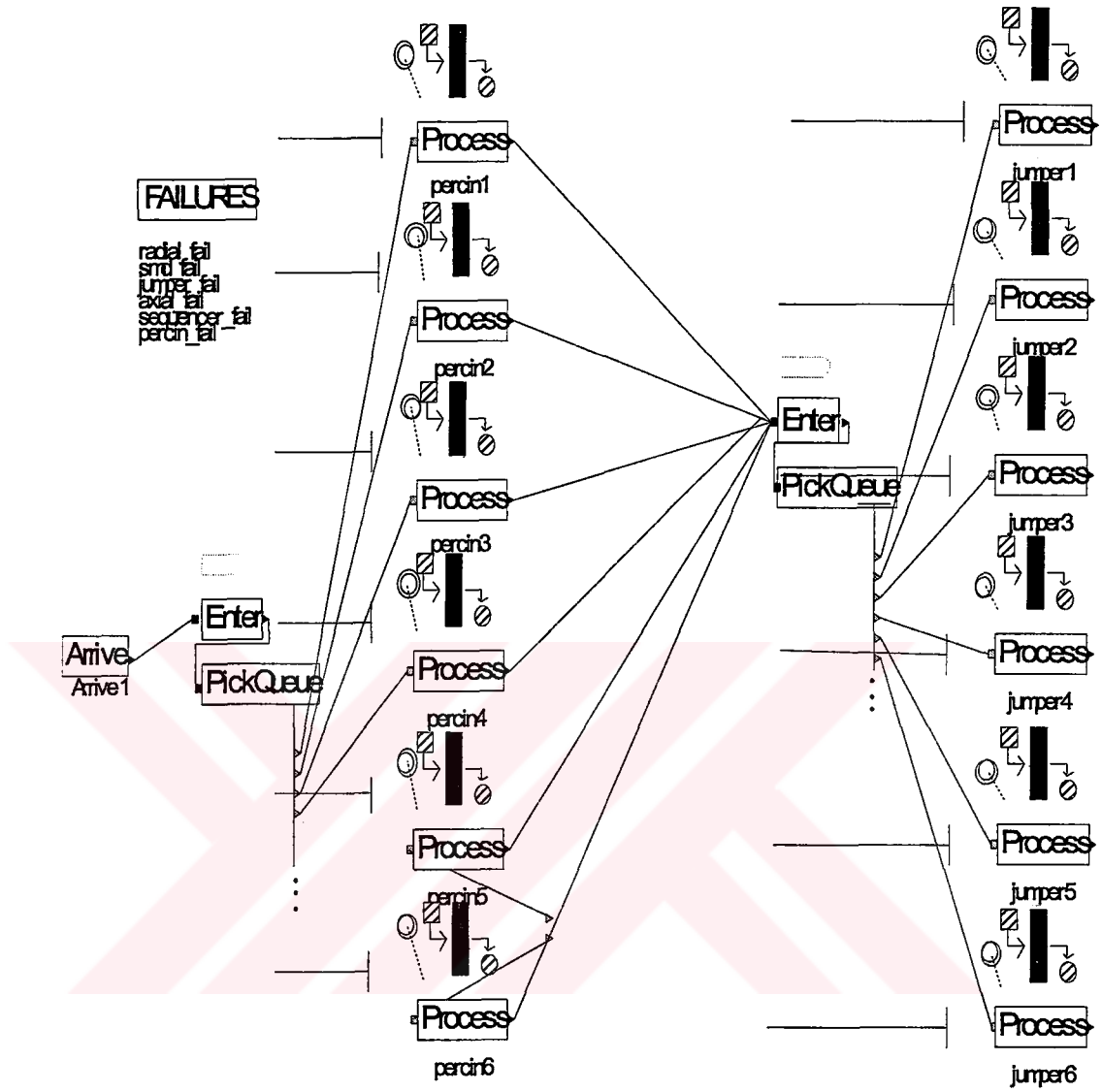
Tüm makinelerin Tablo 12'de gösterilen geçmiş verilere dayalı arıza ve tamir süreleri Bakım bölümü raporlarından alınmış ve FAILURES bloğu ile modele dahil edilmiştir.

Tablo 12 Mevcut Durum İçin Tamir Ve Arıza İstatistikleri

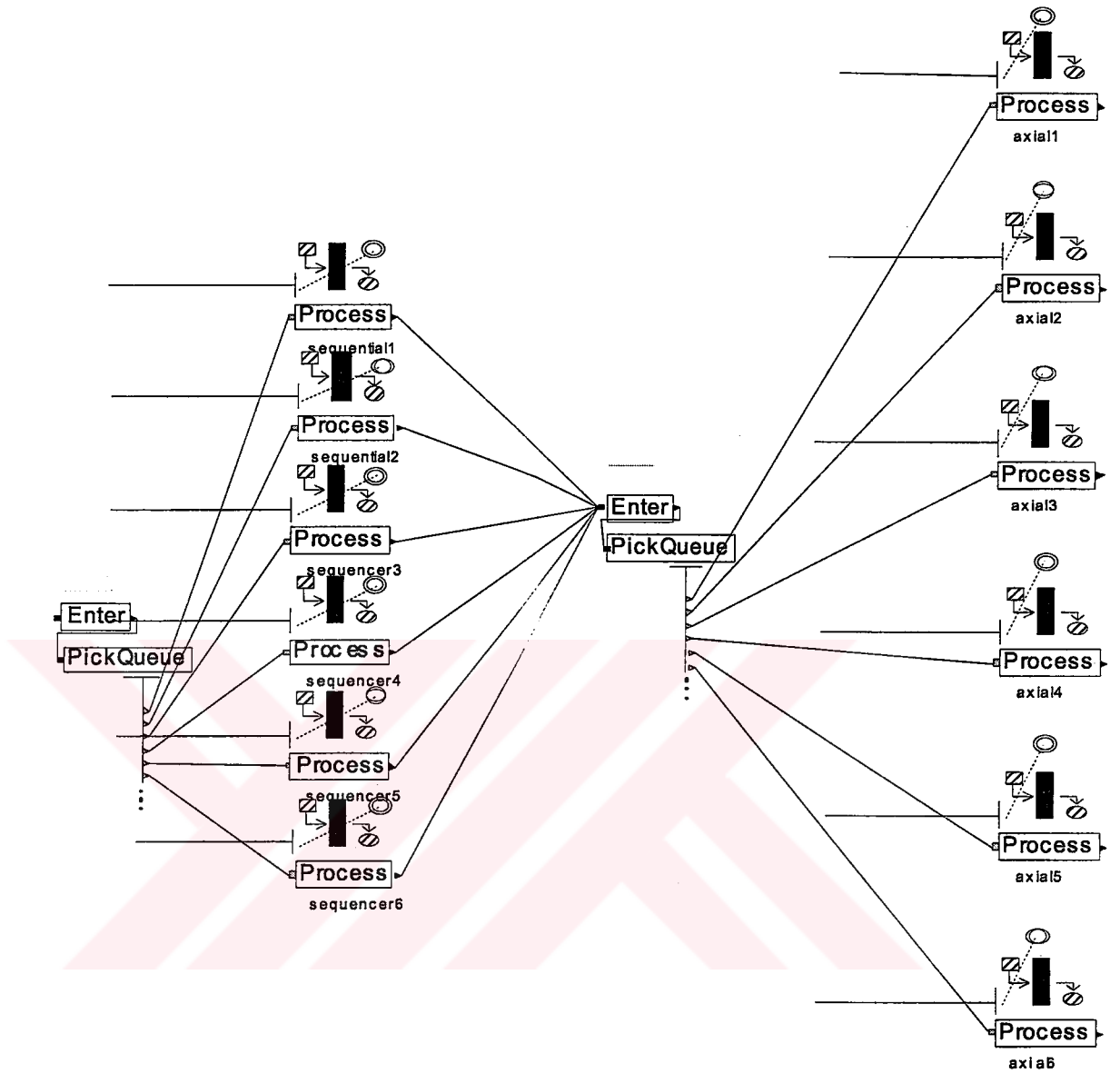
Makine	Dağılım	Arıza	Tamir
		Ort- Std. sap.(Dak.)	Ort- Std. sap.(Dak.)
Perçin	Normal	120 – 5	4 – 1
Jumper	Normal	120 – 5	6 – 1
Sequencer	Normal	100 – 5	4 – 1
Axial	Normal	90 – 5	4 – 1
Radial	Normal	80 – 5	5 – 1
Smd	Üstel	60	6 – 1



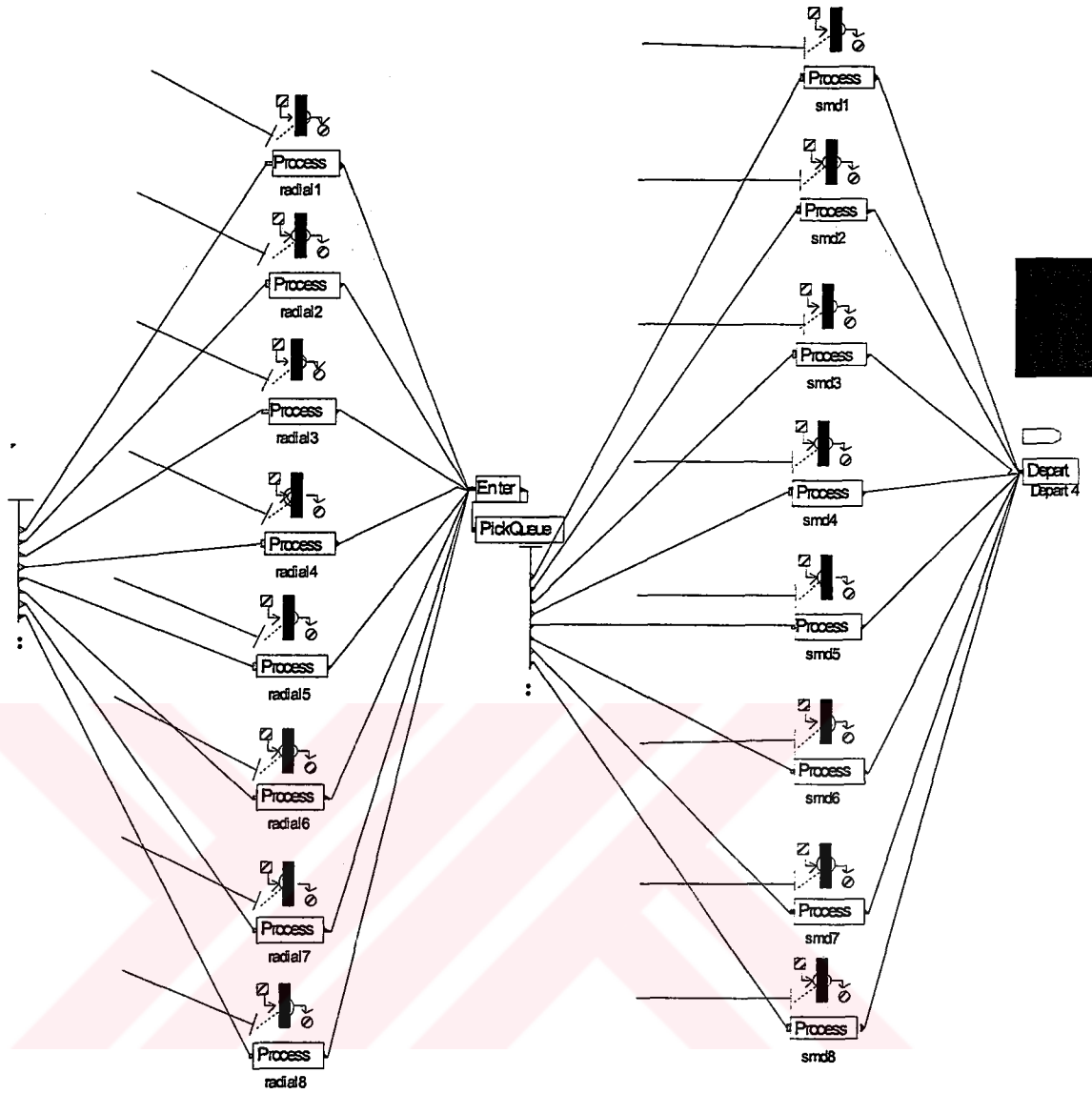
Şekil 13 Arena Modelinin Genel İş Akışı



Şekil 14 Perçin Ve Jumper Gruplarının Yerleşimi



Şekil 15 Sequencer Ve Axial Gruplarının Yerleşimi



ŞEKİL 16 Radial Ve Smd Gruplarının Yerleşimi

4.7 VERİLERİN (İŞLEM SÜRELERİNİN) HAZIRLANMASI

Her makine türünün işlem süresi standart olmakla birlikte, ihmal edilebilecek kadar düşük bir miktar farklılık göstermektedir. Bu nedenle her makine türünden bir tanesine zaman etüdü yapılarak 50'şer gözlem alınmıştır..

Örnek olarak perçin makinesinde yapılan 50 gözlem değeri aşağıdaki gibidir.

21,4 21,6 21,5 21,7 21,5 21,6 21,5 21,3 21,6 21,5
 21,4 21,4 21,7 21,4 21,4 21,5 21,6 21,5 21,3 21,1
 21,6 21,5 21,3 21,6 21,5 21,3 21,6 21,5 21,6 21,7
 21,5 21,6 21,6 21,6 21,5 21,5 21,5 21,3 21,3 21,5
 21,4 21,3 21,6 21,4 21,8 21,6 21,7 21,5 21,6 21,6

Bu gözlemler her makine için tekrarlanmış ve bilgisayarda bir text dosyasına girilerek ARENA'nın Input Analyzer modülü çalıştırılmıştır. Analiz sonucunda özet olarak Tablo13'teki işlem süreleri ve dağılımları elde edilmiştir.

Tablo 13 Makinelerin Çalışma Zamanlarının istatistikleri

Perçin	Norm	(21,5 ; 0,122)
Jumper	Normal	(43,7 ; 0,227)
Sequencer	Normal	(25,8 ; 0,01)
Axial	Beta	$25.8 + 0.06 * \text{BETA}(3,81 ; 5,01)$
Radial	Erlang	$38.7 + \text{ERLA}(0,0631 ; 7)$
Smd	Beta	$40.9 + 0.13 * \text{BETA}(4,28 ; 5,49)$

Özet tablonun açılımı, diğer bir deyişle her bir makine grubu için simülasyon analizi yapabilmek için gerekli istatistiki tüm analiz ve detaylı dağılım raporları tablodan sonra verilmiştir

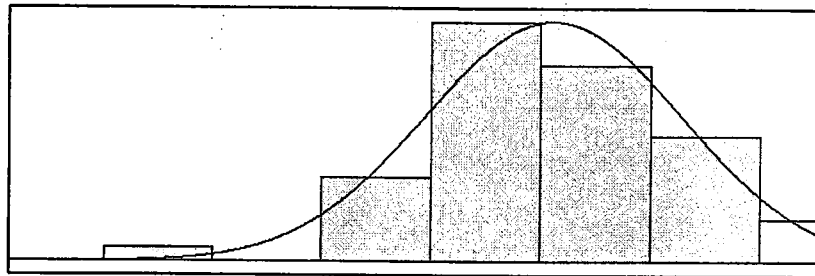
Tablo 13.A Perçin Analizi

Fit All Summary

Data File: C:\ARENA22\work\percin.dst

Function Sq Error

Normal	0.00531
Weibull	0.00704
Beta	0.00816
Gamma	0.00921
Erlang	0.00962
Lognormal	0.0192
Triangular	0.0375
Uniform	0.102
Exponential	0.176

**Distribution Summary**

Distribution: Normal
 Expression: NORM(21.5, 0.122)
 Square Error: 0.005311

Chi Square Test

Number of intervals = 4
 Degrees of freedom = 1
 Test Statistic = 1.03
 Corresponding p-value = 0.335

Kolmogorov-Smirnov Test

Test Statistic = 0.0579
 Corresponding p-value > 0.15

Data Summary

Number of Data Points = 50
 Min Data Value = 21.1
 Max Data Value = 21.7
 Sample Mean = 21.5
 Sample Std Dev = 0.123

Histogram Summary

Histogram Range = 21 to 21.8
 Number of Intervals = 7

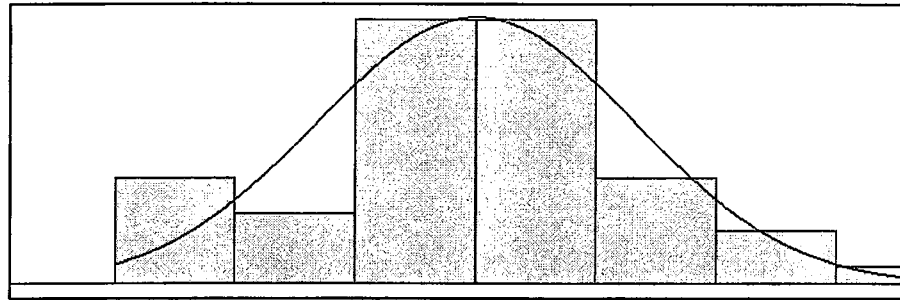
Tablo 13.B Jumper Analizi

Fit All Summary

Data File: C:\ARENA22\work\jumper.dst

Function Sq Error

Normal 0.0136
 Weibull 0.0209
 Triangular 0.0226
 Beta 0.023
 Gamma 0.0356
 Erlang 0.0364
 Lognormal 0.0564
 Uniform 0.0763
 Exponential 0.106
 Distribution Summary



Distribution: Normal
 Expression: $NORM(43.7, 0.227)$
 Square Error: 0.013596

Chi Square Test

Number of intervals = 4
 Degrees of freedom = 1
 Test Statistic = 0.831
 Corresponding p-value = 0.392

Kolmogorov-Smirnov Test

Test Statistic = 0.0665
 Corresponding p-value > 0.15

Data Summary

Number of Data Points = 50
 Min Data Value = 43.3
 Max Data Value = 44.3
 Sample Mean = 43.7
 Sample Std Dev = 0.23

Histogram Summary

Histogram Range = 43.2 to 44.4
 Number of Intervals = 7

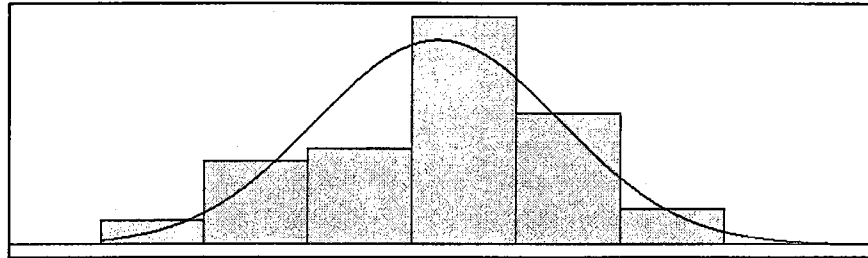
Tablo 13.C Sequencer Analizi

Fit All Summary

Data File: C:\ARENA22\work\sequencer.dst

Function Sq Error

Normal	0.0177
Weibull	0.0197
Beta	0.0208
Triangular	0.0212
Gamma	0.0408
Erlang	0.0413
Lognormal	0.056
Uniform	0.1
Exponential	0.152

**Distribution Summary**

Distribution: Normal
 Expression: NORM(25.8, 0.0999)
 Square Error: 0.017676

Chi Square Test
 Number of intervals = 4
 Degrees of freedom = 1
 Test Statistic = 3.69
 Corresponding p-value = 0.0564

Kolmogorov-Smirnov Test
 Test Statistic = 0.0862
 Corresponding p-value > 0.15

Data Summary

Number of Data Points = 50
 Min Data Value = 25.6
 Max Data Value = 26
 Sample Mean = 25.8
 Sample Std Dev = 0.0101

Histogram Summary

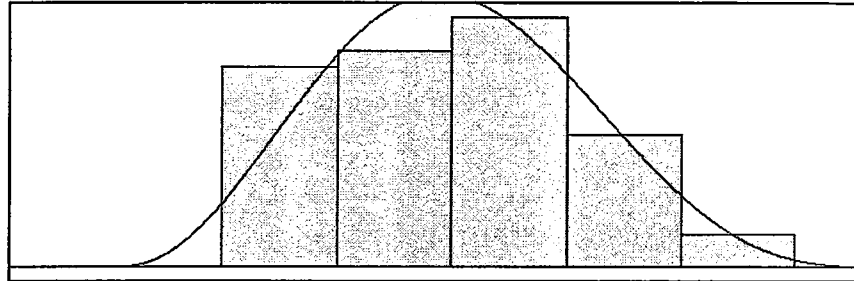
Histogram Range = 25.8 to 26.1
 Number of Intervals = 7

Tablo 13.D Axial Analizi**Fit All Summary**

Data File: C:\ARENA22\work\axial.dst

Function Sq Error

Beta	0.00815
Weibull	0.0105
Normal	0.0141
Gamma	0.0151
Erlang	0.0154
Triangular	0.0186
Lognormal	0.0199
Uniform	0.0995
Exponential	0.141

**Distribution Summary**

Distribution: Beta

Expression: $25.8 + 0.06 * \text{BETA}(3.81, 5.01)$

Square Error: 0.008153

Chi Square Test

Number of intervals = 4

Degrees of freedom = 1

Test Statistic = 1.04

Corresponding p-value = 0.332

Kolmogorov-Smirnov Test

Test Statistic = 0.0622

Corresponding p-value > 0.15

Data Summary

Number of Data Points = 50

Min Data Value = 25.8

Max Data Value = 25.9

Sample Mean = 25.9

Sample Std Dev = 0.00948

Histogram Summary

Histogram Range = 25.8 to 25.9

Number of Intervals = 7

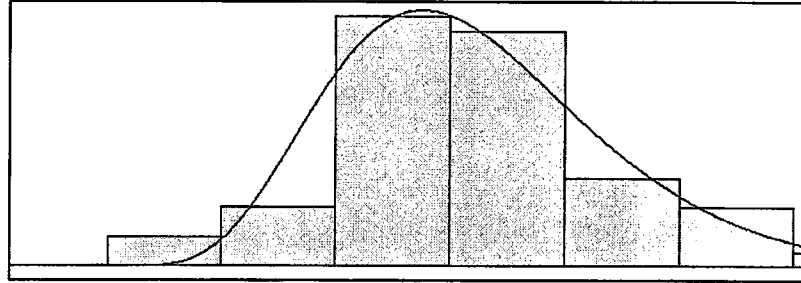
Tablo 13.E Radial Analizi

Fit All Summary

Data File: C:\ARENA22\work\radial.dst

Function Sq Error

Erlang	0.00775
Gamma	0.0093
Normal	0.0105
Beta	0.0113
Weibull	0.0114
Lognormal	0.0166
Triangular	0.0239
Uniform	0.104
Exponential	0.153



Distribution Summary

Distribution: Erlang
 Expression: $38.7 + \text{ERLA}(0.0631, 7)$
 Square Error: 0.007752

Chi Square Test

Number of intervals = 4
 Degrees of freedom = 1
 Test Statistic = 0.865
 Corresponding p-value = 0.382

Kolmogorov-Smirnov Test

Test Statistic = 0.113
 Corresponding p-value > 0.15

Data Summary

Number of Data Points = 50
 Min Data Value = 38.8
 Max Data Value = 39.6
 Sample Mean = 39.2
 Sample Std Dev = 0.156

Histogram Summary

Histogram Range = 38.7 to 39.7
 Number of Intervals = 7

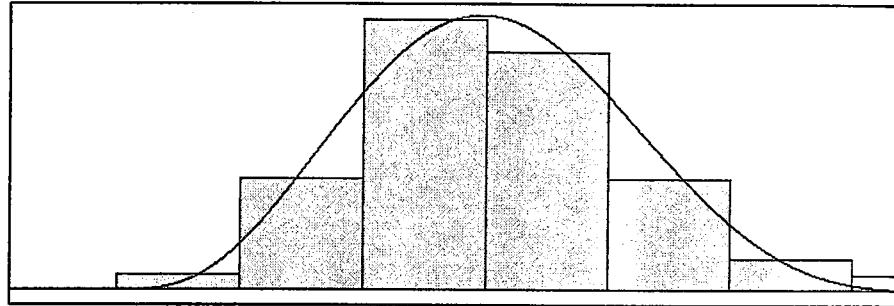
Tablo 13.F Smd Analizi

Fit All Summary

Data File: C:\ARENA22\work\smd.dst

Function Sq Error

Beta	0.00163
Normal	0.00207
Erlang	0.00211
Weibull	0.00227
Gamma	0.00254
Lognormal	0.00737
Triangular	0.0178
Uniform	0.104
Exponential	0.152

**Distribution Summary**

Distribution: Beta

Expression: $40.9 + 0.13 * \text{BETA}(4.28, 5.49)$

Square Error: 0.001626

Chi Square Test

Number of intervals = 4

Degrees of freedom = 1

Test Statistic = 0.256

Corresponding p-value = 0.641

Kolmogorov-Smirnov Test

Test Statistic = 0.0872

Corresponding p-value > 0.15**Data Summary**

Number of Data Points = 50

Min Data Value = 40.9

Max Data Value = 41

Sample Mean = 41

Sample Std Dev = 0.0211

Histogram Summary

Histogram Range = 40.9 to 41

Number of Intervals = 7

4.8 MODELİN DOĞRULANMASI VE GERÇEKLENMESİ (VERIFICATION AND VALIDATION)

Modelin Doğrulanması: Modelin planlandığı gibi çalışıp çalışmadığının belirlenmesidir. Modelin beklenildiği gibi çalışması, modelde mantıksal veya programlama olarak bir hata olmamasına bağlıdır. Ayrıca doğru verilerin kullanılması da modelin doğru çalışması için gereklidir.

Modelin doğrulanması için şunlar yapılmıştır:

- Modeldeki arıza ve tamir süreleri ihmal edilerek, işlem sürelerine basit ve kolay hesaplanabilir değerler verilmiş ve modele az miktarda varlık gönderilerek sistemin çalışması kontrol edilmiştir.
- İşlem süreleri sabit tutularak FAILURES bloğunun çalışması kontrol edilmiştir.
- ARENA'nın Trace (izleme) özelliği sayesinde sistemdeki varlık hareketlerinin detaylı incelemesi yapılmıştır. Bu sayede eğer modelde varlık akışını etkileyen yanlış bir durum varsa bunun nerede olduğunun görülmesi ve önlem alınması kolaylaşmaktadır.

Modelin Gerçeklenmesi: Model sonuçlarının gerçek sistemi yansıtmayı yansıtmadığının belirlenmesi için yapılmaktadır. Gerçekleme, model sonuçlarının doğruluğunun modelleyici tarafından kabul edilebilir bir düzeye gelmesini sağlama işlemidir. Bu modelin gerçekleştirilmesi için modelin çalışması sonucu bulunan günlük ortalama çıktı sayısı 10.763 ve standart sapma değeri de 287,25 bulunmuştur. ARENA'nın %95 güven aralığında simülasyonu çalıştırdığı gözönüne alınarak, ortalama çıktının güven aralığı:

$$10200 = \text{ortalama çıktı} = 11326$$

içerisinde yer aldığı ifade edilir.

Ayrıca makinelerin önündeki kuyruk miktarları da incelenmiş, modelin çalışması sonucunda, gerçek sistemde olduğu gibi Jumper makinelerinin önündeki kuyruğun en uzun olduğu belirlenmiştir.

4.9 MODELİN SONUÇLARI

ARENA, sonuçları "Output Data Analysis" modülü aracılığıyla vermektedir.

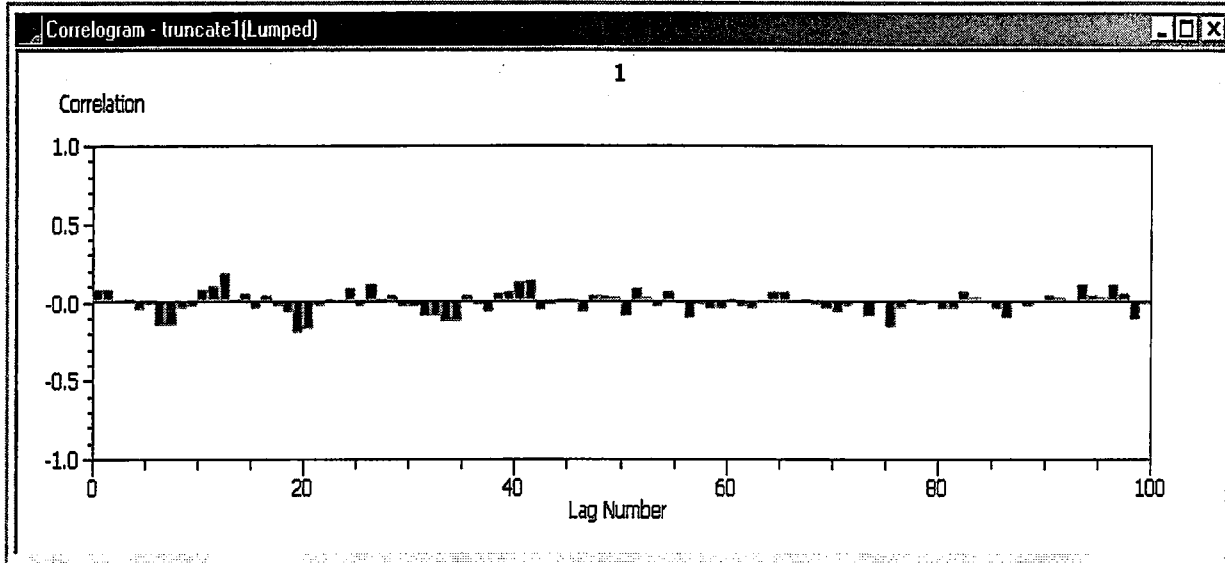
1) Mevcut durum

Mevcut durum, uygulamanın yapıldığı 2002 yılı itibari ile sistemin durumunu ve üretim kapasitesini yansıtmaktadır. AK 19 kodlu şaseyi temsilen, varlıklar belirli zaman aralıklarında yaratılmakta, bunlar ilk makine grubunda işlenmek üzere kuyruğa girmektedirler. Mevcut durumda, önceden verilen şekil 13'teki Arena Modelinin Genel İş akışında görüldüğü gibi, ilk makine grubu olan perçin makinelerinde işlenmek üzere, gözlemlenilen 7 saniyede bir varlık yaratılmaktadır. Yaratılan varlıklar, 6 perçin makinesinin en kısa kuyruk olan makinesine gitmektedir. Perçin makinelerinden geçen varlıklar iş akışındaki rotaya göre ilerlemekte ve sonuçta günde (1440 dakikada) ortalama 10763 şase üretilmektedir.

Simülasyonda gözönünde tutulması gereken önemli noktalardan biri, karar vermeden önce durgun hale (steady state) ulaşmış olmasıdır. Replikasyonlar belli bir sayıya ulaştığında sonuçların ortalamaları arasındaki fark ihmal edilebilir¹²⁰. Mevcut üretim sisteminin Arena dili ile modellenmesine yönelik olarak, Arena simülasyon programının Run Setup modülü 24 saatlik çalışma süresi girilerek koşulmuş ve simülasyonun kararlı hale gelmesi (Warm up Period) için gerekli replikasyon sayısının 30 olduğu raporlanmış ve ardından,

¹²⁰ Hulusi, DEMİR, Cengiz YILMAZ, Benzetim ve Envanter Kontrolü, **Sanayi Mühendisliği**, 1985, Sayı5, s.24-28

modül tarafından Correlogramda 200 replikasyondan sonra sonucun tutarlı olacağı raporlanmıştır.



Şekil 17 Simulasyon Çıktılarının Korelasyon Diyagramı

Correlogram Summary

Statistic 2 (Lumped)

Sample mean : 10763
 Sample std. deviation : 287,25
 Sample size: 200
 Weighted Sum of cov.: 8,857

Correlation / Covariance Values

Lag	Covariance	Correlation
1	4,4565	0,081305

Speed		Run Control		Reports	
Project Parameters		Replication Parameters			
Number of Replications:		Initialize Between Replications			
1		<input checked="" type="checkbox"/> Statistics		<input checked="" type="checkbox"/> System	
Warm-up Period:		Time Units:			
30		Hours			
Replication Length:		Time Units:			
230		Hours			
Hours Per Day:		Base Time Units:			
24		Hours			
Terminating Condition:					
Tamam		İptal		Uygula	
				Yardım	

Şekil 18 Simülasyon Replikasyon Raporu

Şekil 17 ve 18' deki değerlerle simülasyon sonuçlarının ancak 200 replikasyonla birbirinden bağımsız hale geldiği ve sistemin kararlı hale geçmesi için gerekli 30 replikasyonla birlikte toplam 230 replikasyon yapılması gerektiği ve sonuçların arasında korelasyon bulunmadığı görülmektedir.

Modelle ilgili makinelerin önünde oluşan kuyrukların ortalama zamanları Tablo 14'de görülmektedir. Tablo 15'te her bir makine önünde oluşan kuyruk miktarlarının maksimum ve ortalama değerleri ARENA tarafından raporlanmıştır. Tablo incelendiğinde, özellikle Jumper grubunun darboğaz oluşturarak çıktı miktarlarının düşük kalmasına neden olduğu gözlemlenmektedir.

Bu gözlemden hareket edilerek, gerçek sistemin akışına müdahale etmeden bazı önerilerde bulunarak ve bu önerilerin simülasyon modelinde uygulaması yapılarak 10.763 değerindeki sistem çıktısının artırılmasına yönelik çalışmalara geçilmiştir.

Tablo 14 Mevcut Durum İçin Kuyrukta Ortalama Bekleme Süreleri
Toplam Üretim Miktarı: 10763

Identifier	Average	Maximum
Axial1_Q Queue Time	0,19342	5,2553
Axial2_Q Queue Time	0,19245	6,0084
Axial3_Q Queue Time	0,1933	6,5145
Axial4_Q Queue Time	0,16018	4,5749
Axial5_Q Queue Time	0,14946	5,1363
Axial6_Q Queue Time	0,2031	5,3666
jumper1_Q Queue Time	1,1938	6,1317
jumper2_Q Queue Time	1,5909	6,8481
jumper3_Q Queue Time	1,61	6,9055
jumper4_Q Queue Time	1,806	8,2357
jumper5_Q Queue Time	1,9966	9,0719
jumper6_Q Queue Time	1,8597	7,7712
percin1_Q Queue Time	0,09603	4,7753
percin2_Q Queue Time	0,13567	5,822
percin3_Q Queue Time	0,11661	6,719
percin4_Q Queue Time	0,104	5,064
percin5_Q Queue Time	0,0949	5,0068
percin6_Q Queue Time	0,09318	4,5027
radial1_Q Queue Time	0,3845	6,7178
radial2_Q Queue Time	0,31396	5,8807
radial3_Q Queue Time	0,34243	6,7181
radial4_Q Queue Time	0,37636	6,0682
radial5_Q Queue Time	0,33823	6,044
radial6_Q Queue Time	0,38365	6,7521
radial7_Q Queue Time	0,33114	6,2308
radial8_Q Queue Time	0,37478	7,7747
sequencer3_Q Queue Tim	0,1433	5,4109
sequencer4_Q Queue Tim	0,1461	5,4476
sequencer5_Q Queue Tim	0,13619	4,494
sequencer6_Q Queue Tim	0,16721	5,0952
sequential1_Q Queue Ti	0,15379	5,19
sequential2_Q Queue Ti	0,18907	5,1044
Smd1_Q Queue Time	0,87705	9,3945
Smd2_Q Queue Time	0,59181	8,303
Smd3_Q Queue Time	1,4974	12,14
Smd4_Q Queue Time	0,75435	13,722
Smd5_Q Queue Time	0,78436	10,561
Smd6_Q Queue Time	0,69721	10,19
Smd7_Q Queue Time	0,6003	7,014
Smd8_Q Queue Time	1	10,537

Tablo 15 Mevcut Durum İçin Kuyruktaki Ortalama Varlık Sayısı
Toplam Üretim Miktarı : 10763

Identifier	Average	Maximum	Final Value
# in axial1_Q	0,24164	8	0
# in axial2_Q	0,24043	8	0
# in axial3_Q	0,2415	8	0
# in axial4_Q	0,20011	6	0
# in axial5_Q	0,18662	7	0
# in axial6_Q	0,25359	8	0
# in jumper1_Q	1,4923	8	0
# in jumper2_Q	1,9887	9	0
# in jumper3_Q	2,0126	10	0
# in jumper4_Q	2,2564	11	2
# in jumper5_Q	2,4957	12	0
# in jumper6_Q	2,3246	10	4
# in percin1_Q	0,12004	6	0
# in percin2_Q	0,16958	8	0
# in percin3_Q	0,14576	9	0
# in percin4_Q	0,13001	7	0
# in percin5_Q	0,11862	7	0
# in percin6_Q	0,11648	6	0
# in radial1_Q	0,36047	7	0
# in radial2_Q	0,29434	6	0
# in radial3_Q	0,3271	7	4
# in radial4_Q	0,35257	6	0
# in radial5_Q	0,31686	6	0
# in radial6_Q	0,35941	8	0
# in radial7_Q	0,31021	7	0
# in radial8_Q	0,3511	8	0
# in sequencer3_Q	0,17903	8	0
# in sequencer4_Q	0,18252	7	0
# in sequencer5_Q	0,17041	7	1
# in sequencer6_Q	0,2089	8	0
# in sequential1_Q	0,19213	7	0
# in sequential2_Q	0,2362	8	0
# in smd1_Q	0,82101	10	0
# in smd2_Q	0,554	8	0
# in smd3_Q	1,4087	11	6
# in smd4_Q	0,70988	12	4
# in smd5_Q	0,73425	9	0
# in smd6_Q	0,66178	9	6
# in smd7_Q	0,5624	9	2
# in smd8_Q	0,9369	11	3

2) Önerilen Durum 1

Mevcut durumun iyileştirilmesi için çıktığı sınırlayan nedenler gözlemlenmiş ve bunda en büyük payın makine duruşlarından ileri gelen kayıplar olduğu saptanmıştır. Yapılmak istenen ilk değişiklik, duruş zamanlarını azaltabilmek için bakım operatörü sayısını artırarak tamir sürelerinin minimuma indirilmesidir. Bakım operatörü sayısı, her vardiya için birden ikiye çıkarılabileceği, bu durumda da tamir sürelerinin yarısına inebileceği, metod mühendisliği bölümünün önceden yaptığı çalışmalarda tespit edilmiş ancak bundan bir sonuç alamadıkları vurgulanmıştır. Bu iyileştirmenin sonucunda bulunan tamir süreleri Tablo 16'da görülmektedir.

Tablo 16 Öneri 1 İçin Tamir Ve Arıza İstatistikleri

Makine	Dağılım	Öneri1 (Dak)	
		Arıza	Tamir
		Ort-std. sapma	Ort-std. sapma
Perçin	Normal	120 - 5	2 - 1
Jumper	Normal	120 - 5	2 - 1
Sequencer	Normal	100 - 5	2 - 1
Axial	Normal	90 - 5	2 - 1
Radial	Normal	80 - 5	2.5 - 1
Smd	Üstel	60	3 - 1

Gerçekten de önerilen durumun değerleri modele uygulandığında günlük çıktı miktarı 10796'ya yükselmiştir. Bu, çok düşük bir iyileştirme miktarıdır. Tablo 17'de ortalama kuyruk zamanlarına ve Tablo 18'de ortalama kuyruk değerlerine bakıldığında, kuyruk miktarlarında ve bekleme sürelerinde nispi bir azalma olmakla birlikte, günlük çıktının sadece 33 adet artırılabilirdiği görülmektedir.

Tablo 17 Öneri 1 İçin Kuyrukta Ortalama Bekleme Süreleri
Toplam Üretim Miktarı: 10796 Adet

Identifier	Average	Maximum
axial1_Q Queue Time	0,06392	3,7631
axial2_Q Queue Time	0,04959	3,8301
axial3_Q Queue Time	0,04704	3,5857
axial4_Q Queue Time	0,05456	3,137
axial5_Q Queue Time	0,05678	2,8923
axial6_Q Queue Time	0,05087	3,7136
jumper1_Q Queue Time	0,44621	4,4506
jumper2_Q Queue Time	0,38981	5,3689
jumper3_Q Queue Time	0,34833	4,1259
jumper4_Q Queue Time	0,30721	3,743
jumper5_Q Queue Time	0,39976	3,9899
jumper6_Q Queue Time	0,47516	4,4382
percin1_Q Queue Time	0,02249	2,8156
percin2_Q Queue Time	0,03538	3,3969
percin3_Q Queue Time	0,03859	3,0733
percin4_Q Queue Time	0,02896	3,1382
percin5_Q Queue Time	0,02531	2,707
percin6_Q Queue Time	0,02852	2,9096
radial1_Q Queue Time	0,09734	4,3663
radial2_Q Queue Time	0,09505	4,5148
radial3_Q Queue Time	0,0809	3,8459
radial4_Q Queue Time	0,09011	3,6989
radial5_Q Queue Time	0,09486	3,7222
radial6_Q Queue Time	0,08517	3,4412
radial7_Q Queue Time	0,09288	4,0078
radial8_Q Queue Time	0,11228	5,4659
sequencer3_Q Queue Tim	0,034	3,7081
sequencer4_Q Queue Tim	0,04457	3,2031
sequencer5_Q Queue Tim	0,06215	4,8346
sequencer6_Q Queue Tim	0,04543	4,0923
sequential1_Q Queue Ti	0,05247	3,4664
sequential2_Q Queue Ti	0,04969	3,1939
smd1_Q Queue Time	0,24563	4,3217
smd2_Q Queue Time	0,17394	4,0678
smd3_Q Queue Time	0,12557	3,8138
smd4_Q Queue Time	0,31599	6,4455
smd5_Q Queue Time	0,12154	4,3789
smd6_Q Queue Time	0,17124	4,9913
smd7_Q Queue Time	0,17119	4,1147
smd8_Q Queue Time	0,20032	4,3517

Tablo 18 Öneri 1 İçin Kuyruktaki Ortalama Varlık Sayısı
Toplam Üretim Miktarı : 10796 Adet

Identifier	Average	Maximum	Final Value
# in axial1_Q	0,07986	5	0
# in axial2_Q	0,06195	5	0
# in axial3_Q	0,05877	5	0
# in axial4_Q	0,06816	4	0
# in axial5_Q	0,07094	4	0
# in axial6_Q	0,06355	5	0
# in jumper1_Q	0,55776	6	0
# in jumper2_Q	0,48817	7	2
# in jumper3_Q	0,43541	6	0
# in jumper4_Q	0,38637	5	3
# in jumper5_Q	0,4997	6	0
# in jumper6_Q	0,59362	6	0
# in percin1_Q	0,02811	4	0
# in percin2_Q	0,04433	5	1
# in percin3_Q	0,04824	4	0
# in percin4_Q	0,0362	4	0
# in percin5_Q	0,03163	4	0
# in percin6_Q	0,03565	4	0
# in radial1_Q	0,09125	5	0
# in radial2_Q	0,08911	5	0
# in radial3_Q	0,07585	5	0
# in radial4_Q	0,08441	4	0
# in radial5_Q	0,08886	4	0
# in radial6_Q	0,07978	4	0
# in radial7_Q	0,08701	4	0
# in radial8_Q	0,10518	6	0
# in sequencer3_Q	0,04247	4	0
# in sequencer4_Q	0,05568	4	0
# in sequencer5_Q	0,07764	7	0
# in sequencer6_Q	0,05676	6	0
# in sequential1_Q	0,06558	5	0
# in sequential2_Q	0,06208	4	0
# in smd1_Q	0,23011	5	0
# in smd2_Q	0,16295	5	0
# in smd3_Q	0,11764	4	0
# in smd4_Q	0,29602	7	0
# in smd5_Q	0,11386	4	0
# in smd6_Q	0,16053	5	0
# in smd7_Q	0,16049	5	0
# in smd8_Q	0,1878	5	0

3) Önerilen Durum 2

İyileştirme için önerilen diğer bir değişiklik, tamir sürelerinin kısaltılmasının yanısıra, yani her vardiya için bakımcı sayısının 2 olmasının yanında, yeni bakım planları ile makine duruş zamanlarının azaltılmasına yönelik önerilerdir.

Bu amaçla makine duruşlarının temelde hangi nedenlerden kaynaklandığı gözlemlenmiş ve aşağıdaki bilgilere ulaşılmıştır.

<u>Kayıp nedeni</u>	<u>Kayıp Süre(dak)</u>	<u>Kayıp Yüzdesi</u>
a) Periyodik Bakımlar	5,23	%4
b) Küçük Duruşlar ve Arıza Bakımları	73,9	%58
c) Model Değişimi	19	%15
d) Temizlik	10	%8
e) Operatör Bekleme	19,5	%15

Yukarıdaki veriler bir vardiya için ortalama gözlem değerleridir. Model değişimi ile ilgili kayıp değer, sadece tek bir şase üretimi üzerine kurulu modelimiz için geçerli değildir. Bu verilere dayanarak vardiya süresindeki zaman kayıplarının %58'ni oluşturan Küçük Duruşlar ve Arıza Bakımlarının azaltılması hedeflenmiştir. Tam otomatik olarak çalışan ve şaselere komponent çakan Otomatik Dizgi makinelerindeki küçük duruşlar incelendiğinde aşağıdaki nedenlerden dolayı, kendi operatörünün ya da bakımcının müdahalesini gerektirecek şekilde makinelerin arıza kodu vererek ışıklı uyarı ile duruşa geçtiği saptanmıştır.

<u>Arıza Kodu</u>	<u>Tanımı</u>
707	Baskı Hatası
403	Bacağı Kesememe
400	Bacak Okuyamama
401	Çevirici Hatası
853	PCB ters gelmesi
737	Malzeme Bitti

Baskı Hatası şaselerin otomatik Dizgi hattına gelmeden önce, baskısındaki eksiklik ya da yanlışlıklardan kaynaklanır. Bacağı kesememe komponentin şaseye bağlı olduğu bacaklardan birinin otomatik keski ile kesilemediği durumlarda oluşur. Bacağı okuyamama komponentin şaseye olan bağlantısında eksikliği ifade eder. Çevirici hatası komponent çakma işlemleri sırasında şasenin oturduğu hareketli, 3 boyutlu zemindeki mekanik çeviremememe sorununu gösterir. PCB ters uyarısı şasenin istendiği yönde gelmediğini ifade eder. Malzeme bitti ilgili komponentin makinedeki stoğu'nun tükendiği anlamına gelir.

Yukarıdaki arızalardan ileri gelen duruşların azaltılması için aşağıdaki önlemler önerilmiştir:

- Makinelerin elektronik aksamına yapılacak küçük müdahalelerle bazı hataların oluşmadan önce uyarımı(Feeder uyarı Alarmı)
- Operatörlerin hizmet içi eğitimi ile her duruşta bakımçı çağırarak yerine belirli arızalara kendilerinin müdahale etmesi
- Operatörün malzeme getirmek için makinesinden ayrılması yerine, besleyici pozisyonundaki bir eleman ile malzeme yokluğundan kaynaklanan arızaların ortadan kaldırılması. Bu seçenek için yandaki atölye gözlemlenmiş ve bir başka üretim bandı için, belirli aralıklarla çeşitli malzemelerin ambalajlarını açarak yerleştiren ve uzun süreli boşta

kalan işçinin görevlendirilmesi önerilmiştir. Bu sayede yeni işçi alımının getireceği ek maliyetlerden kaçınılmıştır.

- Ayrıca, temizlik için her vardiya bitimine 10 dk. kala makinelerin durdurulduğu tespit edilmiş, şaselere çakılan elektronik komponentlerin kesilmiş parçalarının makine üzerinden temizlenmesi için hava basınçlı tabanca ile 5 dakikanın yeterli olacağı belirlenmiştir. Buna bağlı olarak simülasyon modelinde temizlik süresi yarıya indirilmiştir.

Bu öneriler ve geçmişteki veriler ile yapılan istatistiki değerlendirmeler sonucu Tablo 19'a ulaşılmıştır. Ancak, arızalardan ileri gelen duruşlara yönelik ilk öneri olarak görünen, Feeder uyarı alarmı konusunda hiç bir yetkilinin yeterli bilgi ve deneyimi olmaması nedeniyle, modelde dikkate alınmamıştır.

Tabloda öneri 1'e kıyasla, yeni değişikliklerle arızalar arası sürelerin uzadığı, belirli arızaların bakımçıya gerek kalmadan hizmet içi eğitimden geçirilen operatör tarafından çözümlendiği varsayılmıştır. Bu sayede, bakımçı gerektiren arızalar arası sürenin açılmasına karşın, oluşan arızaların gerek operatör, gerekse bakımçı müdahalesi ile tamiri yapılması durumunda, ortalama tamir süresinin değişmediği görülmüştür.

Tablo 19 Öneri 2 İçin Arıza Ve Tamir İstatistikleri

Makine	Dağılım	Öneri 2 (Dak)	
		Arıza	Tamir
		Ort-std. sapma	Ort-std.sapma
Perçin	Normal	150 - 5	2 - 1
Jumper	Normal	180 - 5	2 - 1
Sequencer	Normal	120 - 5	2 - 1
Axial	Normal	120 - 5	2 - 1
Radial	Normal	100 - 5	2.5 - 1
Smd	Üstel	120	3 - 1

Bir arıza oluştuğunda ortalama tamir süresi aynı kaldığı halde, arızalar arası zamanın uzaması ve malzeme beklemenin yarattığı kayıpların elimine edilmesi ile ilgili model üzerinde gerekli değişiklikler yapılmıştır. Yukarıdaki verilerin simülasyon modeline uygulanıp, programın çalıştırılmasıyla günlük çıktı 10950 olmaktadır. Modelle ilgili kuyruktaki bekleme süreleri Tablo 20'de, kuyruktaki ortalama varlık miktarları ise Tablo 21'de görülmektedir.

İstatistiklere bakıldığında kuyruk uzunluklarının oldukça azaldığı ve bazı makinelerin uzun süre boş kaldığı görülmüştür. Makine önlerinde hiç ya da çok az kuyruk olmasının atıl kalması anlamına da geleceğinden hareketle sistemin daha hızlı yüklenmesi durumundaki davranışı araştırılmış ve simülasyon modelinde bir iyileştirme daha yapılarak, sistemin besleme hızı 7 saniyede bir varlık yerine, 6 saniyede bir varlık yaratılacak şekilde artırılmıştır. Besleme hızının artırılması sonucunda sistemden günde 11767 çıktı alınmıştır. Sistemin besleme hızının 7 sn.'den, 6sn.'ye yükseltilmesi durumunda analitik olarak günlük çalışma süresi kriterine bakarak hesaplanacak olan çıktı sayısının oldukça altında çıktı değerinin alınmış olması, duruşların öneminin simülasyonla modelleme sayesinde belirginleşen sonuçlar açısından dikkat çekicidir. Başlangıçtaki 10763 değerinden hareketle, sistemde %10 çıktı artışı elde edilmiştir. Bu durumla ilgili istatistikler Tablo 22 ve Tablo 23'te Öneri 3 ile gösterilmiştir.

Daha sonra, besleme hızının artırılması durumunda sistemin davranışı araştırılmış, ancak mevcut şartlar dahilinde sistemden daha fazla çıktı alınamayacağı görülmüştür.

Tablo 20 Öneri 2 İçin Kuyrukta Ortalama Bekleme Süreleri
Toplam Üretim Miktarı : 10950 Adet

Identifier	Average	Maximum
axial1_Q Queue Time	0,02547	2,2933
axial2_Q Queue Time	0,02688	2,3455
axial3_Q Queue Time	0,02527	2,9003
axial4_Q Queue Time	0,03443	3,4096
axial5_Q Queue Time	0,05254	3,3354
axial6_Q Queue Time	0,04259	3,4296
jumper1_Q Queue Time	0,43001	3,122
jumper2_Q Queue Time	0,41002	2,9197
jumper3_Q Queue Time	0,42036	4,4248
jumper4_Q Queue Time	0,72948	3,765
jumper5_Q Queue Time	0,18078	2,0282
jumper6_Q Queue Time	0,50647	3,9124
percin1_Q Queue Time	0,02814	3,6895
percin2_Q Queue Time	0,04002	4,1314
percin3_Q Queue Time	0,0303	3,3329
percin4_Q Queue Time	0,01922	2,9991
percin5_Q Queue Time	0,03793	3,7565
percin6_Q Queue Time	0,01863	3,0977
radial1_Q Queue Time	0,07956	3,8848
radial2_Q Queue Time	0,08665	4,5991
radial3_Q Queue Time	0,06809	3,4513
radial4_Q Queue Time	0,10899	4,2919
radial5_Q Queue Time	0,0816	3,4128
radial6_Q Queue Time	0,0702	3,7625
radial7_Q Queue Time	0,05729	3,1323
radial8_Q Queue Time	0,06016	3,5471
sequencer3_Q Queue Tim	0,04353	3,7562
sequencer4_Q Queue Tim	0,03216	3,2392
sequencer5_Q Queue Tim	0,04009	2,9883
sequencer6_Q Queue Tim	0,03352	2,8749
sequential1_Q Queue Ti	0,02681	2,754
sequential2_Q Queue Ti	0,0387	2,8659
smd1_Q Queue Time	0,11504	3,7758
smd2_Q Queue Time	0,11629	4,0029
smd3_Q Queue Time	0,11586	4,1262
smd4_Q Queue Time	0,0559	5,1055
smd5_Q Queue Time	0,10619	4,0561
smd6_Q Queue Time	0,1534	5,3306
smd7_Q Queue Time	0,05829	3,479
smd8_Q Queue Time	0,11114	2,9836

Tablo 21 Öneri 2 İçin Kuyruktaki Ortalama Varlık Sayısı
Toplam Üretim Miktarı: 10950 Adet

Identifier	Average	Maximum	Final Value
# in axial1_Q	0,03396	4	0
# in axial2_Q	0,03698	4	2
# in axial3_Q	0,03368	5	0
# in axial4_Q	0,04589	5	0
# in axial5_Q	0,07002	5	0
# in axial6_Q	0,05789	5	2
# in jumper1_Q	0,57568	5	4
# in jumper2_Q	0,5467	4	0
# in jumper3_Q	0,56049	6	0
# in jumper4_Q	0,97265	6	0
# in jumper5_Q	0,24104	3	0
# in jumper6_Q	0,6753	6	0
# in percin1_Q	0,03752	5	0
# in percin2_Q	0,05336	6	0
# in percin3_Q	0,0404	5	0
# in percin4_Q	0,02563	4	0
# in percin5_Q	0,05057	6	0
# in percin6_Q	0,02484	5	0
# in radial1_Q	0,0795	5	0
# in radial2_Q	0,08659	5	0
# in radial3_Q	0,06804	4	0
# in radial4_Q	0,10891	5	0
# in radial5_Q	0,08155	4	0
# in radial6_Q	0,07015	5	0
# in radial7_Q	0,06195	4	3
# in radial8_Q	0,06012	4	0
# in sequencer3_Q	0,05815	6	1
# in sequencer4_Q	0,04286	5	0
# in sequencer5_Q	0,05343	4	0
# in sequencer6_Q	0,04467	4	0
# in sequential1_Q	0,03575	4	0
# in sequential2_Q	0,0516	4	0
# in smd1_Q	0,11496	4	0
# in smd2_Q	0,11621	5	0
# in smd3_Q	0,11578	5	0
# in smd4_Q	0,05586	5	0
# in smd5_Q	0,10612	4	0
# in smd6_Q	0,1533	6	0
# in smd7_Q	0,05825	4	0
# in smd8_Q	0,11098	3	0

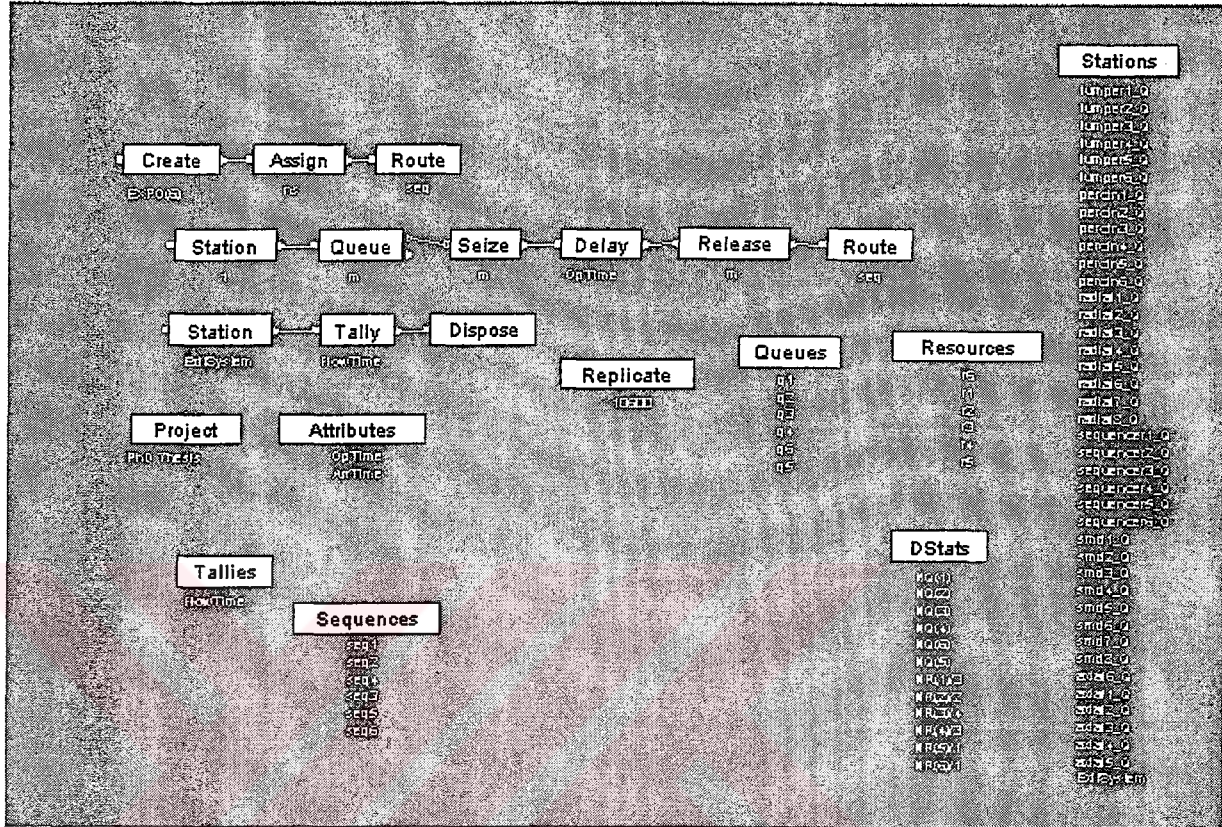
Tablo 22 Öneri 3 İçin Kuyrukta Ortalama Bekleme Süreleri Toplam
 Üretim Miktarı : 11767 Adet Besleme Hızı 6 sn.'de 1 şase

Identifier	Average	Maximum
axial1_Q Queue Time	0,02256	2,7907
axial2_Q Queue Time	0,02471	2,5589
axial3_Q Queue Time	0,03057	2,788
axial4_Q Queue Time	0,02353	2,7762
axial5_Q Queue Time	0,03546	3,2649
axial6_Q Queue Time	0,05397	4,409
jumper1_Q Queue Time	3,0234	7,3139
jumper2_Q Queue Time	6,362	11,879
jumper3_Q Queue Time	3,3867	6,7169
jumper4_Q Queue Time	5,3241	10,164
jumper5_Q Queue Time	2,735	6,0429
jumper6_Q Queue Time	4,8339	10,572
percin1_Q Queue Time	0,02404	3,6091
percin2_Q Queue Time	0,03542	3,7697
percin3_Q Queue Time	0,0259	2,8024
percin4_Q Queue Time	0,02638	2,9095
percin5_Q Queue Time	0,02156	3,2559
percin6_Q Queue Time	0,02148	3,9585
radial1_Q Queue Time	0,10195	4,0874
radial2_Q Queue Time	0,10382	3,5668
radial3_Q Queue Time	0,08345	3,3216
radial4_Q Queue Time	0,06571	3,4161
radial5_Q Queue Time	0,06087	3,5332
radial6_Q Queue Time	0,09598	3,8014
radial7_Q Queue Time	0,09004	3,4573
radial8_Q Queue Time	0,12569	5,0013
sequencer3_Q Queue Tim	0,03158	3,4127
sequencer4_Q Queue Tim	0,04476	3,0394
sequencer5_Q Queue Tim	0,04169	3,4553
sequencer6_Q Queue Tim	0,02922	3,0626
sequential1_Q Queue Ti	0,05169	4,745
sequential2_Q Queue Ti	0,03123	3,5819
smd1_Q Queue Time	0,18796	6,3277
smd2_Q Queue Time	0,19318	5,0774
smd3_Q Queue Time	0,1267	4,8049
smd4_Q Queue Time	0,07946	4,6976
smd5_Q Queue Time	0,13326	4,4163
smd6_Q Queue Time	0,09845	3,6698
smd7_Q Queue Time	0,06145	3,5185
smd8_Q Queue Time	0,08266	3,3395

Tablo 23 Öneri 3 İçin Kuyruktaki Ortalama Varlık Sayısı
 Toplam Üretim : 11767 Adet Besleme Hızı 6 sn.'de 1 şase

Identifier	Average	Maximum	Final Value
# in axial1_Q	0,03074	4	0
# in axial2_Q	0,03365	4	0
# in axial3_Q	0,04163	4	0
# in axial4_Q	0,03204	4	0
# in axial5_Q	0,04829	5	0
# in axial6_Q	0,07349	7	0
# in jumper1_Q	4,1442	11	9
# in jumper2_Q	8,7067	17	15
# in jumper3_Q	4,6372	10	8
# in jumper4_Q	7,2889	14	13
# in jumper5_Q	3,7462	9	7
# in jumper6_Q	6,6257	15	14
# in percin1_Q	0,03294	5	0
# in percin2_Q	0,04854	6	0
# in percin3_Q	0,03548	4	0
# in percin4_Q	0,03615	4	0
# in percin5_Q	0,02954	5	0
# in percin6_Q	0,02943	6	0
# in radial1_Q	0,10415	5	0
# in radial2_Q	0,10605	4	0
# in radial3_Q	0,08525	4	0
# in radial4_Q	0,06712	4	0
# in radial5_Q	0,06213	4	0
# in radial6_Q	0,09798	4	0
# in radial7_Q	0,09198	4	0
# in radial8_Q	0,1284	6	0
# in sequencer3_Q	0,04301	4	0
# in sequencer4_Q	0,06095	5	0
# in sequencer5_Q	0,05678	5	0
# in sequencer6_Q	0,03979	5	0
# in sequential1_Q	0,07038	7	0
# in sequential2_Q	0,04253	5	0
# in smd1_Q	0,192	7	0
# in smd2_Q	0,19733	5	0
# in smd3_Q	0,12943	5	0
# in smd4_Q	0,08117	5	0
# in smd5_Q	0,13613	5	0
# in smd6_Q	0,10057	4	0
# in smd7_Q	0,06273	4	0
# in smd8_Q	0,08438	4	0

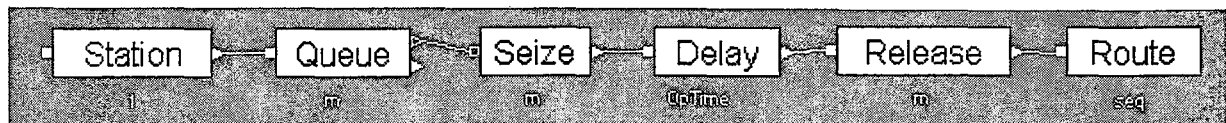
4.10 SİSTEMİN ELEMENTER MODELİ



Şekil 19 Arena Simülasyon Programının Blok Ve Elemanları

Yukarıdaki model ARENA simülasyon programının blok ve elemanlarından oluşmaktadır. Bu elemanlar daha öncede belirtildiği gibi bloklardır. Bloklarda tanımlanan değişkenler ve varlıklar ARENA'daki arayüz biriminden alınıp SIMAN diline dönüştürülür.

4.10.1 Bir Makinenin Genel Davranışı



Şekil 20 Bir Makinenin Genel Davranışı

Sistemdeki bir makinenin genel davranışı yukarıda belirtilen süreci izler. Her makinenin kendi kuyruğu vardır. Kuyruğa gelen varlık (Entity) makinenin boşta

olmasına baęlı olarak kuyruktan alınır. (Seize). Makine alıřır hale gelerek operasyon sresi boyunca alıřır halde kalır.

4.10.2 Create Bloęu

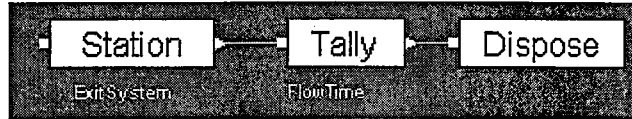
Őekil 21 Create Bloęu

Sistemin gözlemlenmesi sonucu ilk makineye řaselerin geliř hızı 6 saniye olduęu için, CREATE bloęu geliřler arası srelerin stel daęılıma gre ortalaması 6 saniye olacak Őekilde yaratılır. ArrTime deęiřkenine ASSIGN bloęu aracılıęıyla atanır.

Delay: Operasyon sresi bu blokta ifade edilir. Makine operasyon sresi kadar alıřtıktan sonra , operasyon tamamlanır. Bunun sonucunda makine serbest hale geer. Bu durum RELEASE bloęunda ifade edilir. Eęer makinede takım aparat deęiřimi olacaksa, oluřacak gecikme yine bu bloęun farklı kullanımıyla tanımlanır.

Son olarak işlemi tamamlanan parça belirlenen rotayı izleyerek bir sonraki makine grubuna geçer.

4.10.3 Bir İşlemin Sistem İçinde Sona Erme Durumu



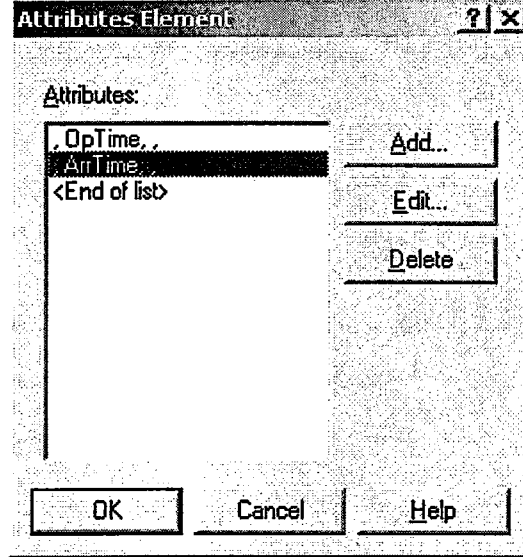
Şekil 22 Bir İşlemin Sistem İçinde Sona Ermesi

Bir işlemin sistem içinde sona erdirilmesinde amaç, simülasyonun performans değerlerinin elde edilmesidir. ARENAda sistemden çıkabilmek için *ExitSystem* kavramı kullanılır. Bu simülasyon çalışmasında makinenin performans değeri Flowtime'dır. Flowtime parçanın ilgili makinenin kuyruğuna gelişinden operasyon zamanının bitimine kadar olan zamanı ifade eder.

4.10.4 ATTRIBUTES Elemanı

Makinelerin temel iki niteliği bulunmaktadır. Bunlar;

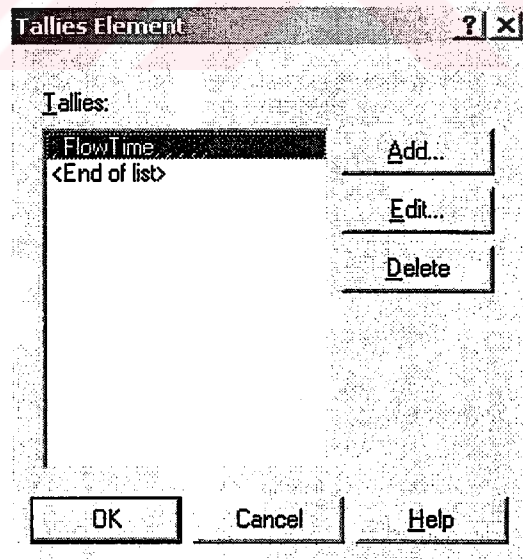
1. OpTime (Operation Time): İlgili makinenin operasyon (işlem) zamanını göstermektedir.
2. ArrTime (Arrival Time) : İlgili makineye uğrayan varlıkların gelişler arası süreleridir.



Şekil 23 Attributes Elemanı

4.10.5 TALLIES Elemanı

ARENA simülasyon dilinde genel performans değerlerinin saptanmasında kullanılan bir elemandır. Bu simülasyon modelinde genel performans ölçütü FlowTime'dır. Bu ölçüt, makinelerin kullanım oranlarını saptamak ve makinelerin dolu-boş sürelerinin tespit etmek amacıyla kullanılır.

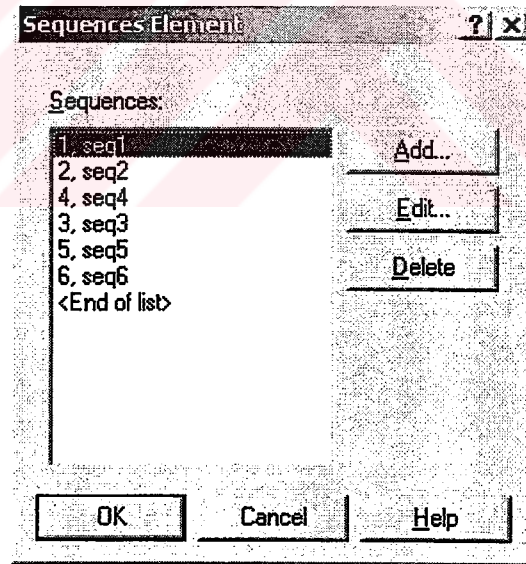


Şekil 24 Tallies Elemanı

4.10.6 SEQUENCES Elemanı

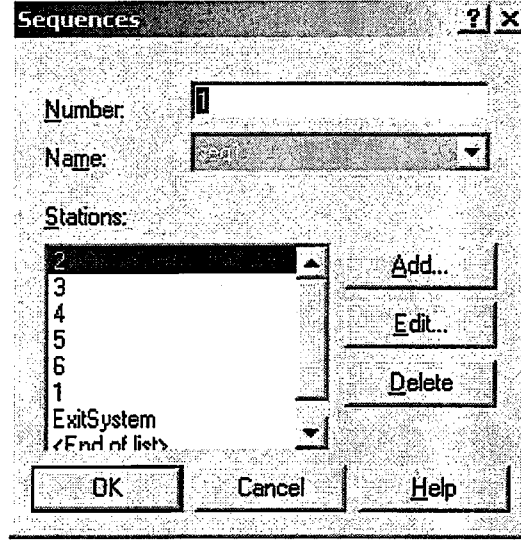
Bu elemanda makine grupları ile her bir makine grubu içerisindeki makinelerin işlem öncelikleri tanımlanır. Mevcut simülasyon modelimizde 6 makine grubu olduğundan 6 işlem sırası (sequence) tanımlanır. Makine grupları şöyle isimlendirilmiştir:

seq1	→	Perçin makine grubu
seq2	→	Jumper makine grubu
seq3	→	Sequencer makine grubu
seq4	→	Axial makine grubu
seq5	→	Radial makine grubu
seq6	→	Smd makine grubu



Şekil 25 Sequences Elemanı

- **PERÇİN Makine Grubu**

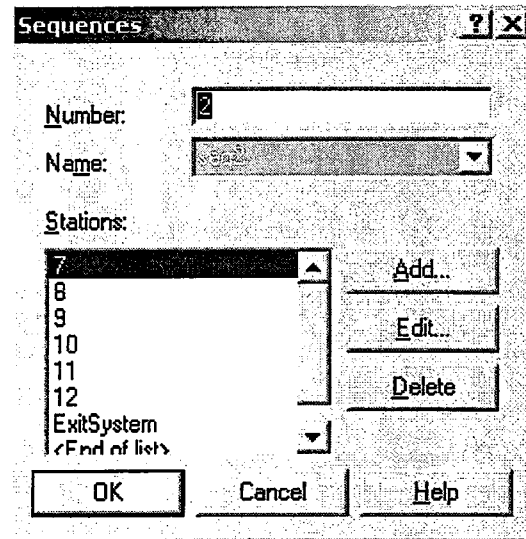


Şekil 26 Perçin Makine Grubu

Perçin makine grubu 6 makineden oluşmuştur. İşlem sırası küçükten büyüğe öncelik alacak şekilde sıralanmıştır. Bu makine grubunun kuyruğuna gelen bir varlık, işlem sonucunda makinenin ExitSystem tanımlı davranışına bağlı olarak bu makine grubunu terk eder.

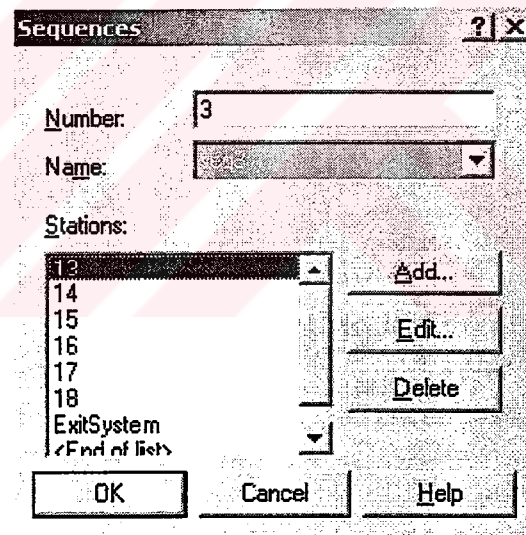
- **JUMPER Makine Grubu**

Perçin makine grubunu terk eden varlık, daha sonra Sequences elemanına bağlı olarak Jumper makine grubunun kuyruğuna gelir. Burada da 6 adet jumper makinesi bulunmaktadır. Aynı şekilde işlem sırası küçükten büyüğe göre olacak şekilde işlem önceliğine sahiptir.



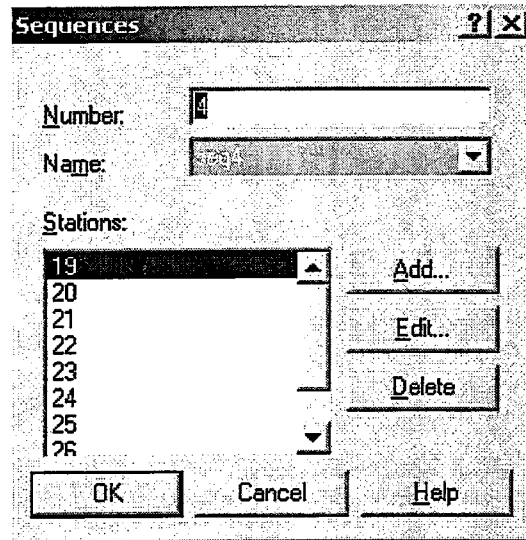
Şekil 27 Jumper Makine Grubu

- SEQUENCER Makine Grubu



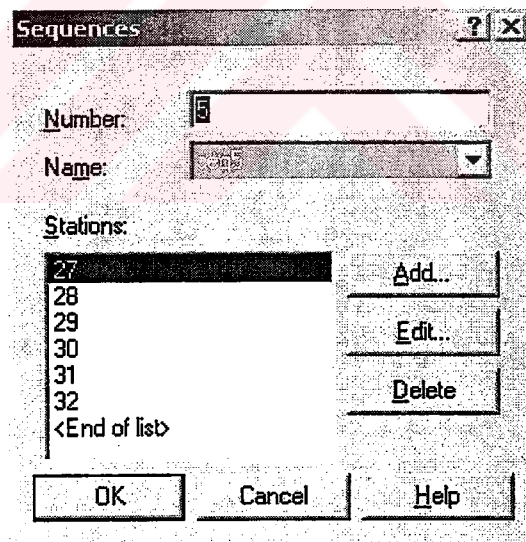
Şekil 28 Sequencer Makine Grubu

- **AXIAL Makine Grubu**



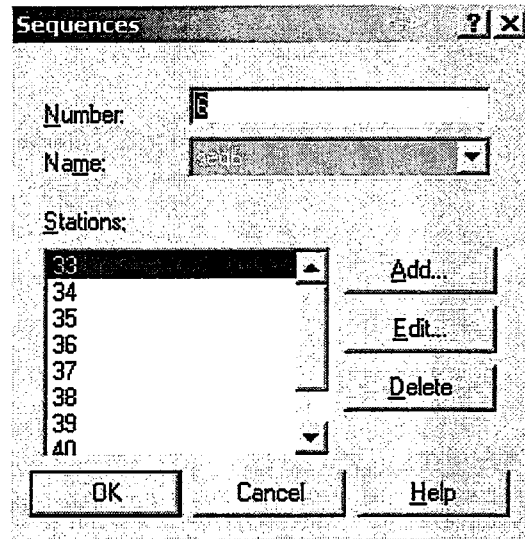
Şekil 29 Axial Makine Grubu

- **RADIAL Makine Grubu**



Şekil 30 Radial Makine Grubu

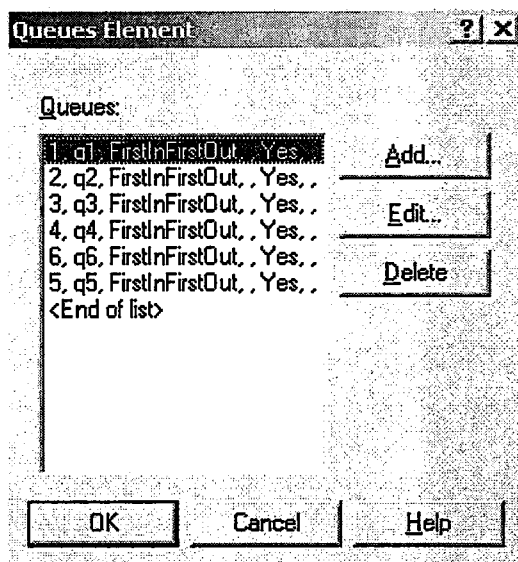
- SMD Makine Grubu



Şekil 31 Smd Makine Grubu

4.10.7 QUEUES Elemanı

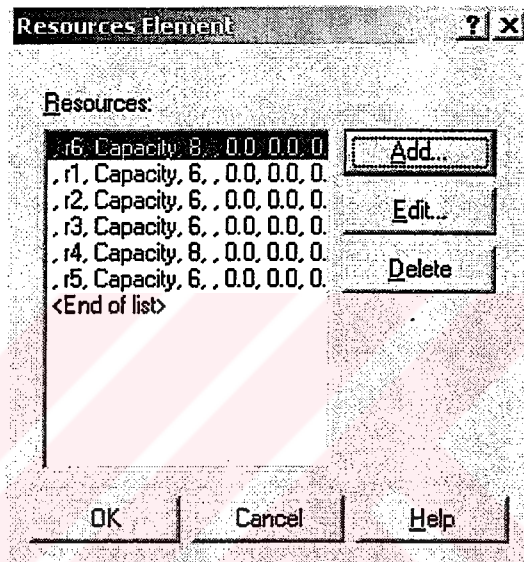
Bu elemanda kuyrukların sıralı tanımları ve davranış biçimleri ele alınır. Modelimizde 6 makine grubuna ait 6 kuyruk bulunmaktadır. Her bir kuyruğun işlem sırası, ismi, kuyruk tipi ifade edilmiştir. Buna göre, kuyruklar FIFO (ilk giren ilk çıkar) yöntemine göre işlemektedir. Kuyruktan alınan varlık, kuyruқта bekleyen varlık sayısını bir azaltmaktadır.



Şekil 32 Queues Elemanı

4.10.8 RESOURCES Elemanı

Bu elemanda, model içerisinde işlemleri gerçekleştirecek nesnel kapasiteleri ile modele tanıtır. Modelimizde işlemleri gerçekleştirecek olan nesnel makine grupları olduğu için, altı kaynak oluşturulmuştur. Her bir makine grubuna ait olan makine sayısı kaynak kapasitesi olarak ifade edilir.



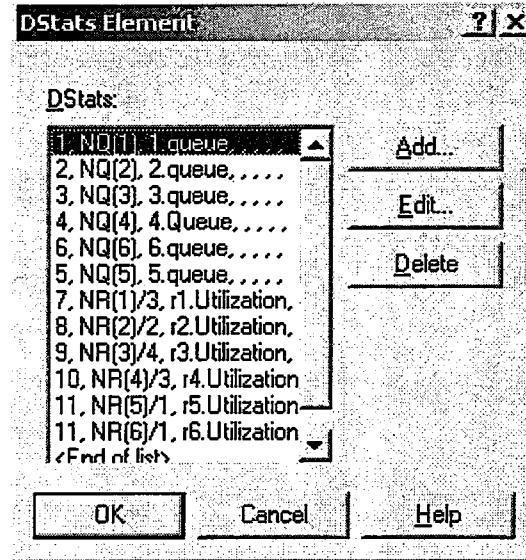
Şekil 33 Resources Elemanı

4.10.9 DSTATS Elemanı

Modelin çalıştırılması sonucunda elde edilen sonuçların anlamlı şekilde ifade edilmesi gerekmektedir. Bunun için bazı istatistiki değerlere ihtiyacımız bulunmaktadır. Bu modelde ARENA' da daha önceden tanımlı olan NQ(sayı) ve NR(sayı) deyimlerine ihtiyaç duyulur.

NQ(sayı) : Belirtilen sayıda olan kuyruktaki varlık sayısını ifade eder.

NR(sayı) : Belirtilen sayıdaki kaynak miktarı.

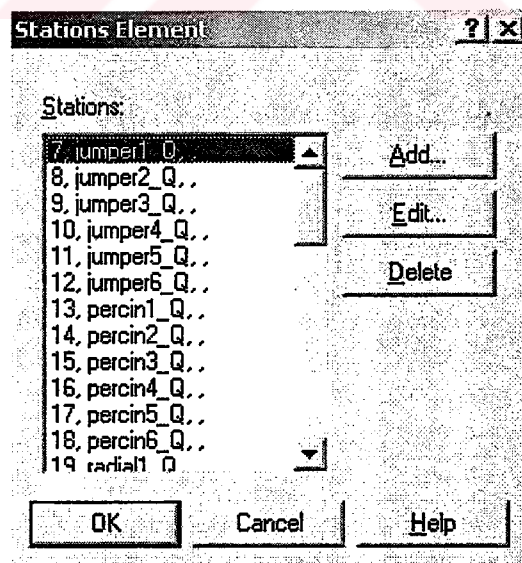


Şekil 34 Dstats Elemanı

Modelimizde, kuyruklardaki varlık miktarı ile makine gruplarının ortalama kullanım oranları ifade edilmiştir. Ayrıca modelimizde daha önceden tanımlı olan TALLY değeri *FlowTime* aracılığı ile makinelerin dolu / boş süreleri elde edilir.

4.10.10 STATIONS Elemanı

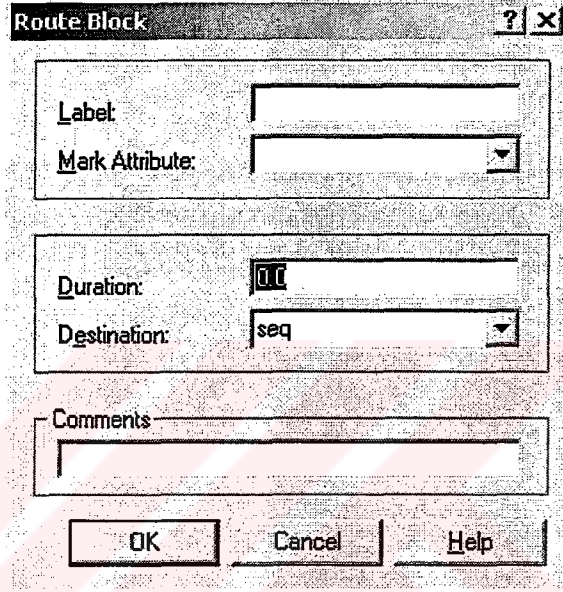
Bu elemanda her bir makinenin modele tanıtımı gerçekleştirilir.



Şekil 35 Stations Elemanı

4.10.11 ROUTE Elemanı

Varlıkların model içerisinde nasıl hareket edeceğini belirleyen sistem bloğudur. Modelimizde varlıklar ROUTE bloğu aracılığıyla seq olarak tanımlanmış SEQUENCES elemanını adres olarak ele alır.



The image shows a dialog box titled "Route Block" with a standard Windows-style title bar containing a question mark and a close button. The dialog is divided into several sections. The first section contains a "Label:" text box and a "Mark Attribute:" dropdown menu. The second section contains a "Duration:" text box with the value "00" and a "Destination:" dropdown menu with "seq" selected. The third section is a "Comments" text area. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

Şekil 36 Route Elemanı

DEĞERLENDİRME VE SONUÇ

Günümüzde, örgütler, artan müşteri memnuniyeti hedefini sağlayabilmek için, müşteri odaklı çalışmaya yönelmekte ve bunu sağlayabilmek için de; örgütteki idari ve fiziksel sistemleri etkin biçimde yöneterek, hızlı kararlar almak ve uygulamak zorundadırlar.

Simülasyon, örgütlerde, teorik ya da gerçek fiziksel bir sistemin neden sonuç ilişkilerini bir bilgisayar modeline yansıtarak değişik koşullar altında sistemin davranışlarını izlemek ve analiz etmek, onu revize etmek ya da mevcut sistemlerden esinlenerek yeni bir sistem modeli kurmak amacıyla çok geniş çalışma alanlarında yoğun olarak uzun yıllar boyunca kullanılan bir yöneylem araştırması tekniği olmasına rağmen, 1990'lerden sonra bilgisayar teknolojilerindeki hızlı gelişmeler, bilgisayarların donanım, yazılım ve hızlarındaki olağanüstü ilerlemeler sonucu, bilgisayarlı simülasyon yazılımları da bu gelişime paralel bir hızla kendini yenilemiştir. Art arda geliştirilen yeni diller ve simülasyon paketleri ile bugün yapay zeka ve sanal gerçeklik kavramları dahi tartışılır olmuştur.

Simülasyon analizlerinin uzmanlara ihtiyaç duyması, modellemenin uzun zaman alması ve pahalı olması gibi dezavantajlarına rağmen, üretim sistemlerinin tasarım ve analizinde, envanter politikalarında, işçi ve makine kapasite sorunlarında, tamir ve bakım sorunlarında, bankacılıktan hipermarket, sağlık, akaryakıt vb. sektörlerle kadar tüm kasa/gişe/servis istasyonu sayısı ile ilgili kuyruk sorunlarında, havacılık ve savunma sanayinde özellikle uçuş ve silah tasarımı benzetimlerine varıncaya kadar, analitik yolların çözüm üretmediği, içinde çok sayıda öge ve karmaşık ilişkiler bulunan sistemlerin analizinde simülasyonun getirdiği esneklik ve analiz yeteneği o'nu eşsiz bir karar destek aracı haline getirmiştir.

Bu uygulamada, TV üretim tesisindeki otomatik dizgi makinelerinin modelini kurmak amacıyla, önce ürün rotalaması yapılmış, iş akışları güncelleştirilmiştir.

İş akışlarına göre, makine parkında ürünün geçtiği her makine detaylı olarak incelenmiş, 6 grup makine için, her gruptaki 6 ya da 7 eş makinenin kendinden önce gelen makineden parça çekmeleri, her parçanın makinenin dolu olması halinde kuyruğa girmesi, boşalan ilk makineye geçişi ve birbirleri arasındaki ilişkiler farklı günler, farklı saatler boyunca defalarca gözlemlenmiştir. Arıza süreleri ve oluş sıklıkları kayıt edilmiş ayrıca planlama departmanından geçmiş zamana ait kayıtları çıkartılarak çalışmada kullanılmıştır. Malzeme aktarmalar incelenmiş, aktarma zamanları sıklıkları kayıt edilmiştir. Makinelerde hatalı takılan elektronik komponentlerin sayıları titizlikle kayıt edilmiştir. Makine duruşları ayrıntılı olarak incelenerek duruşların nerelerden kaynaklandığı belirlenmiş ve bunların üzerine gidilerek önerilerde bulunulmuştur.

Modeli kurmak için şaseleri temsilen varlıklar yaratılarak ilk istasyona sokulmuş sonra da diğer makinelerde kuyruğu kısa olana yönlmesi sağlanmıştır. Rota üzerinde şasenin gideceği makinelerin tümünün dolu olması durumunda boşalacak ilk makineye gitmesi sağlanmıştır. Ardışık işlemler modelde oluşturulup model tamamlanmıştır. Şaselerin makinelere gelişleri zaman etüdü ile saptanan dağılımlara göre tanımlanarak modele konmuştur. Kurulan modelin gerçek sistemi doğru olarak temsil edip etmediği, model ile gerçek sistem birebir karşılaştırılarak denenmiş, oluşan hatalar ayıklanmış ve geçerliliği sağlandıktan sonra model çalıştırılmıştır.

Kurulan ARENA modeli ile metot etüdü sonucu yapılan öneriler modele uygulanmış ve sonuçları raporlanmıştır. Mevcut durumda günde ortalama 10763 şase üretilmektedir. Modelle ilgili makinelerin önlerinde oluşan kuyruklara bakıldığında, aynı gerçek sistemde olduğu gibi jumper grubunun darboğaz oluşturduğu tespit edilmiştir. 1. öneri, her vardiya için yeni bir bakımcı olarak tamir süresini kısaltma düşüncesi, modele 1+1=2 bakımcı olarak yansıtılmış; ancak bu önerinin, çıktı sayısını ancak 33 adetlik artışla 10796 adetten öteye götüremeyeceği bulunmuştur.

Ardından üretim akışını en çok yavaşlatan nedenlerin, değişik faktörlerden kaynaklanan küçük duruşlar olduğu tespit edilmiş ve yeni bakımcıya ilave olarak öneri 2 olarak tanımlanan bir grup iyileştirme çalışması ile:

- Operatörlerin arıza durumunda bakımcı çağırmak yerine, hizmet içi eğitimle sık karşılaşılan arızalara kendilerinin müdahale ettiği,
- Operatörün malzeme bittiğinde, malzeme getirmek üzere yerinden ayrılması yerine, besleyici pozisyonunda boş zamanı olan mevcut bir elemanın makinelere malzeme bitmeden parça taşıdığı,
- Temizlik süresinin yarıya indirildiği,

varsayılarak, modelde çalışma, boşta bekleme, temizlik süreleri ve arıza ile geçen süreler yeniden düzenlenmiş ve model yeniden çalıştırılmıştır. Modelin sonuçlarına göre günlük çıktı 11767ye yükselmiştir. Görüldüğü gibi bu, yaklaşık %10 luk bir üretim artışını işaret etmektedir.

Sonuç olarak, müşteri siparişlerini yetiştirebilmek için günde 24 saat aralıksız çalışan otomatik dizgi makine parkında, revizyon ve iyileştirme çalışmaları için dahi montaj bandının durdurulmasına tahammül edilemezken, gelişmiş simülasyon programı ARENA sayesinde sistem tüm öğeleri ile modellenerek metot mühendisliği önerileri modelde tam anlamıyla olumlu sonuçlanmış ve de bunun gerçek hayata taşınması üst yönetimin iznini alarak kısmen gerçekleşmiştir.

Bu üretim örneği ve benzerlerinde, bir çok nedenden dolayı gerçek sistem yerine, gerçek sistemin sanal bir modeli üzerinde her türlü denemeyi yapmak yöneticiler için bulunmaz bir fırsattır. Sadece kapasite artışı değil, bu çalışmada söz edilen sayısız alanda doğru ve yerinde yeni simülasyon paketlerinin kullanımı ile sorunların çözümünde ve yeni sistemlerin tasarımında simülasyon tekniğinin, yöneticiler için vazgeçilmez bir karar destek aracı olacağı söylenebilir.

KAYNAKÇA

KİTAPLAR

ANDERSON, D., SWENEY, D., WILLIAMS, T. (2000), **Management Science**, South Western College Pub.

AVERELL, M. Law, LARMEY, Chris (1992), **Introduction to Simulation Using Simscript II.5**, CA.

AVERILL, M. Law, KELTON, W. David (1991), **Simulation Modeling and Analysis**, Newyork, Mc Graw Hill.

BACKER Michael, SCHALLNER Harald (1997), **Model and Simulation based Flexible Production Management System, Computer Applications in Production and Engineering**, Chapman & Hall.

BANKS, Jerry, CARSON, John, NELSON, Barry, NICOL David. (2000), **Discrete Event System Simulation**, New Jersey, Prentice Hall.

BRANDIMARTE, Paulo, VILLA, Agostina (1999), **Modeling Management Systems (From aggregate Planning to Real Time Control)**, Berlin, Springer.

BUFFA, Edwood, SARIN, K. Rakesh (1987), **Modern Production and Operations Management**, Canada, John Wiley Sons.

BUZACOTT, John, SHANTIKUMAR, J. George (1993), **Stochastic Models of Manufacturing Systems**, NewJersey, Prentice Hall.

CARRIE, Allan (1998), **Simulation of Manufacturing Systems**, GB, John Wiley Sons.

COLLIS, N., WATSON, S.M., Introduction to Arena, **Proceedings of the 1993 Winter Simulation Conference** .

CONSTANTIN, V. Negoita, RALESCU, Don, NOSTRAND, Van Rainhold (1987), **Simulation Knowledge-based Computing and Fuzzy Statistics**, USA.

ÇAKICI, M., OĞUZHAN, A., ÖZDİL, T.(2000), **Temel İstatistik**, İstanbul, Özal Matbaası.

DAELLENBACH, H.G.(1995); **Systems and Decision Making / A management Science Approach**; John Willey Sons; USA.

DOĞAN, A., ALP, K. (2002), **Yapay Zeka**, İstanbul, Kariyer Yayıncılık.

FABRICKY, W.Y., MIZE, J.H. (1998), **Simulation Modeling and Simnet**, New Jersey, Prentice Hall.

FISCHWICK, P.A. (1995), **Simulation Model Design Execution**, New Jersey Prentice Hall.

GOGG, Thomas J., MOTT, Jack R. (1992), **Improve Quality and Productivity With Simulation**, JMI Consulting Group.

GUASCH, A. (1990), **Object Oriented Simulation; Simulation Councils**, Inc. USA.

HALLAÇ, Osman (1982), **İşletmelerde Simülasyon Teknikleri**, İstanbul, İstanbul Matbaası.

HANEY, R. M. (1991), **Computer Simulation**, California, Academic Press Inc.

HARRELL, Charles, TUMAY, Kerim (2000), **Simulation Made Easy: A Manager's Guide**, Atlanta, Institute of Industrial Engineers.

HARRINGTON, J., TUMAY, K. (2000), **Simulation Modeling Methods to Reduce Risks and Increase Performance**, USA, Mc Graw Hill.

KELTON, David, SADOWSKI, Randall, SADOWSKI, Deborah (2002), **Simulation With Arena**, Newyork, Mc Graw Hill.

KLEIJNEN, J., GROENENDAAL, W.V. (1992), **Simulation a Statistical Perspective**, England, John Willey Sons .

KOBU Bülent (2003), **Üretim Yönetimi**, Onbirinci Basım, İstanbul .

LAWRANCE, John A., PASTERNAK, Barry A. (1998), **A computer Integrated Approach for decision Making**, USA, John Willey Sons.

LUKER, P.A, SCHMIDT, B. (1991), **Knowledge-Based Simulation (Methodology and Application)**, Springer-Verland; Newyork.

MARTIN, Dale E, WILSEY Philip A. (2004), Scheduling Optimization on the Simbus Backplane, **Proceedings of The 37. Annual Simulation Symposium**, Virginia.

MONTGOMORY D., RUNGER G. (2003), **Applied Statistics and Probability For Engineers**, John Wiley & Sons, New York.

NABIYEV, Vasif V. (2004), **Yapay Zeka, Problemler, Yöntemler, Algoritmalar**, İstanbul, Seçkin Yayıncılık.

NEELAMKAVIL, F. (1987), **Computer Simulation and Modeling**, G.B., John Willey Sons.

NEGOITA, Constantin V., RALESCU, Dan (1987), **Simulation Knowledge-Based Computing and Fuzzy Statistics**, USA, Van Nostrand Reinhold Co.

NEWBOLD, Paul(2000), (4. Basımdan Çeviren Ümit ŞENESEN), **İşletme ve İktisat için İstatistik**, İstanbul , Literatür Yayıncılık.

PARSEI, Hamid, SULLIVAN, William G. (1993), **Concurrent Engineering**, Londra, Chapman Hall.

PEGDEN C.D., SHANNON R.E., SADOWSKI.P. (1990), **Introduction to Simulation Using SIMAN**.

PIDD, M. (1998), **Computer simulation in Management Science**, Newyork John Willey Sons.

PIDD, M. (1998) **Tools for Thinking (Modeling in Management Science**, Newyork, John Willey Sons.

PIDD, M. (1999), **Computer Modeling for Discrete Simulation**, UK John Willey Sons.

PRITSKER A.A., REILLY J.J. (1995), **Simulation with Visual Slam and Awesim**, Newyork, John Willey Sons.

RARDIN, L., RONALD, R. (1998), **Optimization in Operations Research**, Prentice Hall, New Jersey, Sage Publications.

REITER, B. Scholz, STAHLMAN, U.U, NETHE, A.(1999), **Process Modeling**, Germany, Spiringer.

ROSS, Sheldon M. (1990), **A Course in Simulation**, USA, Mc Millan Pub. Co.

ROSS, Shepley L. (1984), **Differential Equations**, Newyork, John Willey Sons.

RUBENSTEIN, Reuven Y., MELAMET, Benjamin (1998), **Modern Simulation Modeling**, Newyork, Wiley Interscience Pub.

RUSSEL, Stuart, NORVIG, Peter (2002), **Artificial Intelligence: A Modern Approach**, New York, Prentice Hall.

SMEDS, R., RIIS, J. (1998), **Learning Production Management (the Effects of Using Simulation Games in Universities and Industry)**; Londra, Chapman Hall.

ŞEN, Zekai (2004), **Mühendisler İçin Bulanık (Fuzzy) Mantık ile Modelleme Prensipleri**, İstanbul, Su Vakfı Yayınları.

TANENBAUM, Andrew S., STEEN Maarten Van (2002), **Distributed Systems: Principles and Paradigms**, New York, Prentice-Hall .

TERSINE, Richard (1990), **Production Operations Management**, Newyork, Elsevier.

TOP, Aykut (1996), **Üretim Sistemleri**, İstanbul, Alfa Basım Dağıtım.

TÜTEK, Hülya, GÜMÜŞOĞLU, Şevkinaz (2000), **İşletme İstatistiği**, İzmir, Barış Kitabevi.

WU, B. (1992), **Manufacturing Systems Design**, Londra, Chapman Hall.

YAMAK, Oygur (1999), **Üretim Yönetimi**, İstanbul, Alfa Basım Yayın.

ZEIGLER, Bernard P. (2000), **Theory of Modeling and Simulation**, New York, Academic Press.

MAKALELER

AKHUN, M., DURMUŞOĞLU, B. Worker Allocation Problem In Cellular manufacturing : A case study, **Proceedings of the 1994 European Simulation Symposium.**

ANGLANI, A., GRIECO, M., PACELLA, M., TOLIO, T., Object-Oriented Modeling and Simulation of Flexible Manufacturing Systems: a rule-based procedure , **Proceedings of 2002 Winter Simulation Conference.**

AVARELL, M.Law, MC COMAS, M.G., How to select Simulation Software For Manufacturing Applications, **Industrial Engineering**, July 1992.

BANKS, J., GIBSON, R.R., Getting Started in Simulation Modeling, **IEE Solutions**, November 1996.

BANKS, J., GIBSON, R.R., Simulation Modeling, **IIE Solutions**, February 1997.

BAYRAKTAR, Erkan, Karmaşık Endüstriyel Sistemlerin Tasarım ve Analizi için Simülasyon, **Makine Magazin**, Ocak 1997.

BOSTON Russel R., Designing Simulation Experiments, **Proceedings of Winter Simulation Conference**, 2004, Bredley, P.; Bennet L. F.; Linus E.; (1997); Springer Verlang; Newyork.

CARSON John S., Introduction To Simulation, **Proceedings of the 2003 Winter Simulation Conference.**

CHAN, K.K., SPEDDING, T.A., An Integrated Multidimensional Process Improvement Methodology For Manufacturing Systems, **Computers and Industrial Engineering**, Vol. 44, 2003 .

DEMİR, Hulusi, YILMAZ, Cengiz, Benzetim ve Envanter Kontrolü, **Sanayi Mühendisliği**, 1985.

DİNÇMEN, Murat, Benzetim Tekniği ve Sanayideki Uygulamaları, **Sanayide Bilgisayar Kullanımı ve Otomasyon**, 1990.

DRAKE, Glenn R., SMITH, Jeffrey S., Simulation As a Planning and Scheduling Tool for Flexible Manufacturing Ssystems, **Proceedings of the 1995 Winter Simulation Conference**.

DREVNA, Michael, KSALES, Cynthia J., Introduction to Arena, **Proceedings of the 1994 Winter Simulation Conference**.

GRANT, Hank, LAI, Chuen-Ki, Simulation Modeling With Artificial Reality Technology (Smart): An Integration Of Virtual Reality And Simulation Modeling, **Proceedings of the 1998 Winter Simulation Conference**.

HADDOCK, J., A simulation Generator for Flexible Manufacturing Systems Design and Control, **IIE Transactions**, Vol 20, 1988.

HAJJAR, Dany, ABOURIZK, Siman, MATHER, Kevin, Integrating Neural Networks With Special Purpose Simulation, **Proceedings of the 1998 Winter Simulation Conference**.

HAPCHI Georges, BERCHET Claire, A Model For Manufacturing Systems Simulation With A Control Dimension , **Simulation Modelling Practice And Theory**, Vol. 11, 2003.

HARDING, J.A., POPPLEWELL, K., Simulation: An Application of Factory Design Process Methodology, **Journal of the Operational Research Society**, April 2000.

HOIST Lars, BOLMSJO Gunner, Simulation Integration in Manufacturing System Development: A study of Japanese Industry, **Industrial Management and Data Systems**, 2001.

HOUSHYAR, A., NUILA, V., Required Steps For Successful Design And Implementation Of Simulation, **Computers and Industrial Engineering**, Vol. 25, 1993.

IŞLIER, Atilla, ERDEM, Mümtaz, Ölçüm hataları açısından, Ekonomik Ölçüm Sayısının Belirlenmesi, **Endüstri Mühendisliği Dergisi**, Kasım 1998.

KARATZA, Helen D., Modeling And Simulation Of Distributed Systems And Networks, Simulation Modelling, **Practice and Theory**, Volume 12, 2004.

KHJAN M.R., HARLOCK S.C. Computer simulation of production systems for woven fabric manufacture, **Computers and Industrial Engineering**, Vol 37, 1999.

KLEIJNEN, J.P., Theory and Methodology-Verification and Validation of Simulation Models, **European Journal of Operational Research**, Vol 82, 1995.

LAW, M., MCCOMAS, M.G., How to Select simulation software for Manufacturing Applications, **Industrial Engineering**, July, 1992.

LEEMIS, Lawrence, Input Modelling, **Proceedings of 2003 Winter Simulation Conference**.

LIN, C.K., LAI, K.F., HUNG S.L., Develop of a Workforce Management System For a Customer Hotline Service, **Computers & Operations Research** , Vol. 27, 2000.

MARTIN, Dale E., WILSEY, Philip A., Scheduling Optimization on the Simbus Backplane, **Proceedings of The 37. Annual Simulation Symposium, Virginia, 2004.**

MOUSSA, S.E., MOGRABI, C., EID, M.S., Simulation The First Operation in an Assembly Line, **Computer & Industrial Engineering, Vol 37, 1999.**

NIELSEN, N.R., Application of Artificial Intelligence Techniques to Simulation, **Advances In Simulation, Vol.4, 2003 .**

NIKOLAIDOV M., ANAGNOSTOPOULOS D., A Distributed System Simulation Modelling Approach, **Simulation Modelling Practise And Theory, Vol. 11, 2003 .**

PIDD, Michael, An Introduction to Computer Simulation, **Proceedings of the 1994 Winter Simulation conference .**

PIERREVAL, Henri, PARIS, Jean Lue, From Simulation Optimization To Simulation Configuration Of Systems, **Simulation Modelling Practise And Theory, Vol. 11, 2003 .**

ROBINSON, Stewart, The Application of Computer Simulation in Manufacturing, **Integrated Manufacturing Systems, Vol 4, 1993.**

ROHRER, M., Visualization and Its Importance in Manufacturing Simulation, **Industrial Management, May/June, 1996.**

SANCHEZ, Susan M., ABC of Output Analysis, **Proceedings of 2001 Winter Simulation Conference.**

SHANNON, R., Introduction to Simulation, **Proceedings of the 1992 Winter Simulation Conference.**

SWITEK, Wieslaw, MAJEWSKI, Tadeusz, Dynamic Modeling and Optimization For Technology Management, **Computeres and Industrial Engineering**, Vol.33, 1997.

THOMAS Andre, CHARPENTIER Patrick, Reducing Simulation Models For Scheduling Manufacturing Facilities, **European Journal Of Operational Research**, Vol. 161, 2005.

TUNALI, Semra, Benzetim Tekniğinin bir Çevrim İçi Karar Destek Aracı Olarak Esnek İmalat Sistemlerinde Kullanılması, **Endüstri Mühendisliği Dergisi**, Ağustos, 1996.

İNTERNETTEN SAĞLANAN DÖKÜMANLAR VE UYGULAMADA YARARLANILAN BİLGİLER

ALTUNTAŞ E., ÇELİK T., <http://members.tripod.com/~Bagem/bagem/yz3.html>.

ALTUNTAŞ E., ÇELİK T., <http://members.tripod.com/~Bagem/bagem/yz4.html>.

ARSHAM Hossein, Modeling & Simulation,
<http://cs.sun.za/lynette/simulation/harsham.html>.

Business process improvement with Arena,
[URL: soft.sm.com/arenasim/arenabpi.htm](http://soft.sm.com/arenasim/arenabpi.htm).

Computer Simulation of Business,
www.otterbein.edu/registrar/new2000.htm .

Fundamentals of Simulation,
<http://euclid.ii.metu.edu.tr/ion561/demo/lectures/charter01/indexhtml>.

Key to Arena Simulation Output. for IE/MS D27,

[URL: iems.nwu.edu/~4er/d27/MATERIALS/arenaOut.html](http://iems.nwu.edu/~4er/d27/MATERIALS/arenaOut.html) .

Modeling And Simulation Manufacturing processes,

[URL:www.itri.loyola.edu/polymers/c6_s3.htm](http://www.itri.loyola.edu/polymers/c6_s3.htm).

Path to Succes, <http://www.sm.com.overview/hitepapers/pathtosucces.html>.

Sim Arena Japan - The Japanese Simulation Community,

[URL: mitsu.sim-arena.com/](http://mitsu.sim-arena.com/)

Simulation, Scheduling and Consulting Services by Systems Modeling,

[URL: www.sm.com/](http://www.sm.com/)

Simulation Software Review,

[http://.sepa.tudelft.nl/webstaf/edvinv/SimulationSoftwarepackageges.html](http://sepa.tudelft.nl/webstaf/edvinv/SimulationSoftwarepackageges.html).

Simulation PPT Slide,

<http://www.csupomona.edu/~sparisay/websima/ie429/power429/sld001.html>.

SMS Simulation Modeling Services -. [URL: www.simulation.com.au/](http://www.simulation.com.au/)

<http://matworld.wolfram.com>.

<http://mech.bee.gut.edu.au/men170smsch2.html>.

<http://mech.bee.gut.edu.au/men170smsch4.html>.

<http://mech.bee.gut.edu.au/men170smsch5.html>.

<http://tip.erciyes.edu.tr/anabil/kbb/EGITIM/DERS/>

[http://www.fortunecity.com/simulation.](http://www.fortunecity.com/simulation)

[http://www.hksystems.com/asrs/index.cfm.](http://www.hksystems.com/asrs/index.cfm)

[http://www.sm.com/overview/whitepapers/arenafamilyy.htm.](http://www.sm.com/overview/whitepapers/arenafamilyy.htm)

[http://www.uytes.com.tr/simulasyon/index.html.](http://www.uytes.com.tr/simulasyon/index.html)

[http://www.witness-for-simulation.com/ simulation/witnessoptimization.htm.](http://www.witness-for-simulation.com/simulation/witnessoptimization.htm)



EK 1: ARENA 2.2 SİMÜLASYON PROGRAMI TANITIMI

(Geniş Bilgi için bakınız: ARENA Getting Started Guide, ARENA Variables Guide, System Modeling Corporation, 1994)

ARENA 2.2 Simülasyon yazılımı, önceden SIMAN simülasyon programını da üreten Systems Modelling Co. Adlı bir firmanın geliştirdiği bir üründür. ARENA, başarılı bir simülasyon için gerekli olan sisteme giren ve çıkan varlıkların detaylı istatistiksel analizi, animasyon, esnek modelleme, akış takibi gibi fonksiyonları içermektedir.

ARENA bir program modeli oluştururken ve çalıştırırken SIMAN komutlarını kullanır. Fakat programın görsellik özelliği sayesinde neredeyse komut yazılımına hiç ihtiyaç duyulmaz. ARENA'da şablonlarda kısa yolları verilmiş modülleri program sayfamıza ekleyerek, sonrada bunları çift tıklatarak istenen bilgileri (Geliş zamanı, İşlem süresi, Arıza değerleri vb.) istasyona girerek model programımızı oluşturabiliriz.

ARENA Windows altında çalıştığı için araç çubukları, menüler ve pencerelerle çalışmak konusunda büyük kolaylık sağlar. Dosya işlemleri bir Windows penceresinde olduğu gibi çalışır. Yeni bir dosya açmak için File altında New seçilebileceği gibi mevcut bir dosya da File/Open ile açılabilir. Modellerin kayıt edilmesi için de File/Save komutu kullanılır. File/Print ile modellerin çıktısı alınırken, File/Preview ile ön izleme yapılabilir.

ARENA'nın bazı özellikleri:

- ARENA'nın nesne temelli çizim sisteminde, öncelikle bir nesne seçmeli ve onu izleyen adımlarda gerekli işlemler yapılmalıdır.
- Nesnelerin özelliklerinin değiştirilmesi için File/Undo ve File/Redo komutları vardır.

- Çizim yaparken Shift tuşunu basılı tutmak yapılan işleme bazı özellikler kazandırır. Bu özellikler çizilen şekle göre değişir. Çizgiler yatay, dikey, açılı, daire, elips, dörtgen vs. olarak ayarlanabilir.
- Birden fazla ARENA pencereleri arasında Ctrl + Tab tuşu ile geçiş sağlanır. Klavyedeki Insert tuşuna basmak model içinde ne seçili ise o nesnenin kopyasını yaratır. Böylece yeni nesne taşınabilir.

1.1 Program Prosedürleri

A. Input Analyzer

Veri girdisinin analizini yapar. Simülasyon için ihtiyaç duyulan gelişler arası zaman, servis zamanı gibi süreler istatistiksel analizlere tabi tutulur. Bu amaçla analiz edilecek veriler bir text dosyasına isim verilerek kaydedilir. Daha sonra Input Analyzer da Data File/ Use Existing sekmesi tıklanarak önceden kayıtlı text dosyası seçilip açılır. Fit menüsü içinde Fit All seçilerek, verilere istatistiksel analiz uygulanıp, Verilere en uygun dağılım tespit edilip, raporu alınabilir.

B. ARENA

Modelin oluşturulup çalıştırıldığı ve sonuçlarının alındığı esas programdır. 50 kritere kadar tanıtım versiyonu kullanılırken bu değer üstündeki modellemelerde programın orijinali gereklidir.

C. Output Analyzer

Modelin kurulup Simülasyonun çalıştırılmasının ardından program işleyişi bittiğinde sonuçlar bu modülden raporlanır. İstenirse sonuçlar birbiriyle grafik gibi görsel araçlarla karşılaştırılabilir.

D. Scenario Manager:

SIMAN ve ARENA'da, SIMAN kodlarıyla oluşturulmuş programların derlenmesinde ve link edilmesinde kullanılır. Önce SIMAN kodlarıyla yazılan model program Scenario Manager'da derlenerek "program.e" elde edilir. Daha sonra model program ve deneysel program birleştirilir. Eğer hata yoksa "program.p" elde edilir. Bu program ARENA'da çalıştırılıp Simülasyon yapılır.

E. ARENA Viewer:

Ana programda hazırlanmış model programın çalıştırılıp sonuçlarının alınmasında kullanılır. Aynı işlev ARENA'nın Run menüsü ile de yapılabilir. Ancak ARENA Viewer'ın avantajı lisans gerektirmeden her bilgisayara yüklenip ana programda yapılmış olan simülasyonların çalıştırılmasıdır. Bu program sadece yapılmış programları çalıştırdığından bilgisayarın hafızasında fazla yer kaplamaz. Ana programda hazırlanmış simülasyonlar, file menüsünde bulunan Pack and Go komutuyla paketlenerek Avf uzantılı dosyalara dönüştürülürler. Yalnızca bu dosyalar ARENA Viewer'da çalıştırılabilir. Fakat bu dosyalar üzerinde değişiklik yapılamaz.

1.2 Menüler

ARENA'da temelde File, View, Tools, Help menüleri bulunmaktadır. Bir modelin yerleşmesiyle birlikte Edit, Arrange, Module, Run ve Window menüleri eklenir. Her menü altındaki çeşitli işlev ve yeteneğe sahip komutların mantıksal olarak bağlanmasıyla oluşturulmuştur. Menülerin içindeyken çalışılabilecek kısa yollar her menü komutunun sağ tarafında yer alan kısa yol klavye kombinasyonlarıdır.

- **File Menüsü**

Bu menü yeni ARENA modellerinin yaratıldığı, mevcutlarının açıldığı, pencerelerin kapatıldığı, modellerin kayıt edildiği kısımdır. Burada ayrıca AutoCAD (veya Dxf formatındaki başka CAD programlarından) gibi programlardan CAD çizimleri alınabilir, bunları ARENA içinde kullanabilir ve aktif elementler halinde kullanabiliriz. ARENA'nın yazdırma ayarları ile ilgili komutları bu menüdedir. ARENA son kullanılan dört dosyayı hatırlar ve bunları çabuk açabilmek için yine bu menü altında saklar.

- **Edit Menüsü**

Bu menü içinde ARENA'ya nesnelere ekledikçe onlarla ilgili seçenekleri içeren komutlar vardır. Yapılan işlemler geri alınabilir, seçili bir nesne kopyalanabilir, kesilebilir veya bunlar başka bir modele hatta başka bir uygulamaya taşınabilir. Kopyalanıp clipboard'a yerleştiren nesnelere yapıştırmak için Paste komutu, bu işlemin iki dosya da otomatik olarak güncellenmesi için Paste Link komutu kullanılır. Duplicate komutu ile seçilen nesnenin hemen yanında o nesneden bir tane daha yaratılabilir. Delete komutu ile seçili nesnelere silinebilirken, Select all ile tüm nesnelere aynı anda istenen uygulamaya tabi tutulabilir. Properties ile nesnelere detaylı özelliklerine girilebilir. Modelin diğer dosyalar ile olan bağlantıları Links komutu ile incelenip üzerinde değişiklikler yapılabilir. Diğer uygulamalardan grafik ya da multimedya transferi gerekirse bu durumda Insert New Object komutu kullanılmalıdır.

- **View Menüsü**

Bu menü sayesinde oluşturulan modelin ekranda istenilen şekilde görüntülenmesini sağlayabiliriz. Zoom seçeneği ile modelin istenilen bölümün ekrandaki ayrıntı seviyesi azaltılıp, artırılabilir. Bunu yaparken bu işlemin derecesi Zoom Factor aracılığıyla ayarlanır. Grid modeli karelere

bölerek parçalı görünümle ilgili avantajlar sunar. Layers, modelin düzenlenmesi ve çalıştırılması sırasında hangi nesnelerin gösterilip gösterilmeyeceğini belirler. Araç çubukları seçeneği ile çok sayıda özellik içeren farklı düğme grupları ekrana taşınabilir.

- **Tools Menüsü**

Model Jump Wizard basit girdiler ile yeni bir ARENA modeli oluşturmanın adımlarını sunar. Input Analyzer model girdilerinin girişi ve olasılık dağılımları için kullanılır. Output Analyzer simülasyon sonuçlarının istatistiksel analizi için raporlar hazırlar. Scenario Manager aynı model üzerinde değişik veriler için farklı ayarlar ile çalışabilmeyi böylece onları analiz edebilmeyi sağlar. Show Visual Basic Editor ile Visual Basic kodlarını yazabileceğimiz bir ortam açılır. Options ile ARENA'nın birçok özelliği seçenekler sunularak değiştirilebilir.

- **Arrange Menüsü**

Model ve grafikler için yerleşim ayarlarını içerir. Seçili nesnelerin ekranda fiziksel görünüşlerini bozmadan bir arada tutulmaları veya ayrılmaları ile ilgili düzenlemeleri yapmaya olanak sağlayan Group Ungroup komutları bulunur. Rotate ile seçili nesne 90 derece döndürülürken Flip komutu ile bir eksen etrafında nesnelere döndürmek mümkündür.

- **Module Menüsü**

Bu komutlar modelin mantık komutlarıdır. Connect ile bir modülden diğerine geçilebilir ve iki modül bağlanabilir. Auto Connect seçeneği ile önceden yerleştirilmiş bir modülün yanına yerleşim yapılması halinde otomatik olarak bağ kurma işlemi yapılır. Smart Connections ile yapılan bağlantıların açılı yerleşimlerini sağlamak mümkündür. Breakpoint seçili noktaya gelindiğinde

animasyonu durdururken, Show modülü ile modülün adı ya da animasyonun görüntülenmesi ayarlanabilir.

- **Run Menüsü**

Bu menü, simülasyonun çalıştırılması, kontrol edilmesi, duraksatılması veya yavaş yavaş adımlanması için gerekli komutları içerir. Bunun yanısıra hata ve sonuçlara da yine bu menüden erişilebilir.

- **Window Menüsü**

Aynı anda birçok açık modelin bulunması durumunda bunları fiziksel olarak düzenleyen, Cascade, Tile gibi opsiyonları sunar. Modelleri simge olarak tutmaya olanak sağlayan Arrange Icons komutu bulunur. Açık olan bir modele geçiş bu menü altında ilgili isme tıklamak suretiyle yapılır.

- **Help Menüsü**

ARENA Help Topics ile yardım başlıkları ve bunlardan ilgili seçeneğe tıklanarak detaylı yardım almak mümkündür.

EK 2A : MEVCUT DURUM İÇİN MAKİNELERİN DOLU, BOŞ VE ARIZA ZAMANLARI

Identifier	Category	AvgTime	Percent
STATE(percın6)	BUSY	0,37461	44,8
	IDLE	0,43946	52,46
	FAILED	3,5904	2,74
STATE(jumper2)	BUSY	1,7502	91,04
	IDLE	0,09186	4,71
	FAILED	5,5627	4,25
STATE(jumper3)	BUSY	1,9337	91,05
	IDLE	0,10219	4,73
	FAILED	5,5263	4,22
STATE(jumper4)	BUSY	1,7555	90,95
	IDLE	0,08604	4,39
	FAILED	6,1109	4,67
STATE(jumper5)	BUSY	2,0512	91,02
	IDLE	0,10188	4,44
	FAILED	5,9443	4,54
STATE(jumper6)	BUSY	2,1269	90,84
	IDLE	0,10664	4,47
	FAILED	6,1493	4,7
STATE(sequential1)	BUSY	0,46128	53,72
	IDLE	0,36744	42,64
	FAILED	4,0345	3,64
STATE(axial1)	BUSY	0,47269	53,93
	IDLE	0,3697	41,92
	FAILED	3,9773	4,14
STATE(sequential2)	BUSY	0,46713	53,72
	IDLE	0,37052	42,48
	FAILED	4,207	3,8
STATE(axial2)	BUSY	0,47355	53,93
	IDLE	0,37111	42,01
	FAILED	3,8978	4,06
STATE(axial3)	BUSY	0,47558	53,93
	IDLE	0,37054	41,74
	FAILED	4,1571	4,33

STATE(axial4)	BUSY	0,47154	53,93
	IDLE	0,37035	42,13
	FAILED	3,7828	3,94
STATE(radial1)	BUSY	0,75819	61,23
	IDLE	0,41269	33,1
	FAILED	4,7984	5,66
STATE(axial5)	BUSY	0,46711	53,91
	IDLE	0,36935	42,37
	FAILED	3,5661	3,71
STATE(radial2)	BUSY	0,75336	61,21
	IDLE	0,416	33,57
	FAILED	4,4223	5,22
STATE(smd1)	BUSY	0,91895	63,94
	IDLE	0,38065	26,04
	FAILED	5,771	10,02
STATE(sequencer3)	BUSY	0,46074	53,72
	IDLE	0,36968	42,95
	FAILED	3,6882	3,33
STATE(axial6)	BUSY	0,47717	53,91
	IDLE	0,37227	41,85
	FAILED	4,0629	4,23
STATE(radial3)	BUSY	0,7531	60,98
	IDLE	0,41359	33,23
	FAILED	4,9036	5,79
STATE(smd2)	BUSY	0,85762	63,96
	IDLE	0,38718	28,47
	FAILED	5,4444	7,56
STATE(sequencer4)	BUSY	0,46072	53,72
	IDLE	0,36729	42,62
	FAILED	3,7647	3,66
STATE(radial4)	BUSY	0,75845	61,2
	IDLE	0,41517	33,21
	FAILED	4,7297	5,58
STATE(smd3)	BUSY	1,0783	63,72
	IDLE	0,39126	22,39
	FAILED	5,8814	13,89
STATE(perc1n1)	BUSY	0,37414	44,79
	IDLE	0,43901	52,35

	FAILED	3,7462	2,86
STATE(sequencer5)	BUSY	0,4602	53,69
	IDLE	0,36872	42,84
	FAILED	3,5713	3,47
STATE(radial5)	BUSY	0,7531	61,19
	IDLE	0,41476	33,35
	FAILED	4,6224	5,46
STATE(smd4)	BUSY	0,85349	63,77
	IDLE	0,38764	28,64
	FAILED	6,0662	7,58
STATE(perc2)	BUSY	0,37903	44,8
	IDLE	0,43952	51,86
	FAILED	4,3771	3,34
STATE(sequencer6)	BUSY	0,46236	53,72
	IDLE	0,3686	42,65
	FAILED	4,0293	3,64
STATE(radial6)	BUSY	0,76179	61,21
	IDLE	0,41757	33,29
	FAILED	4,9528	5,5
STATE(smd5)	BUSY	0,89057	64,01
	IDLE	0,38991	27,62
	FAILED	5,7407	8,37
STATE(perc3)	BUSY	0,37652	44,79
	IDLE	0,44004	52,1
	FAILED	4,0681	3,11
STATE(radial7)	BUSY	0,75115	61,19
	IDLE	0,41494	33,45
	FAILED	4,8222	5,36
STATE(smd6)	BUSY	0,86006	63,73
	IDLE	0,38928	28,38
	FAILED	5,1625	7,89
STATE(perc4)	BUSY	0,37498	44,79
	IDLE	0,43917	52,3
	FAILED	3,8053	2,91
STATE(radial8)	BUSY	0,75959	61,19
	IDLE	0,41434	33
	FAILED	4,9192	5,81

STATE(smd7)	BUSY	0,85017	63,88
	IDLE	0,39417	29,18
	FAILED	5,2594	6,94
STATE(perc5)	BUSY	0,37479	44,79
	IDLE	0,43948	52,34
	FAILED	3,7511	2,87
STATE(jumper1)	BUSY	1,439	91,04
	IDLE	0,08232	5,14
	FAILED	4,9998	3,82
STATE(smd8)	BUSY	0,94751	63,83
	IDLE	0,37645	24,91
	FAILED	6,0058	11,26



EK 2B ÖNERİ 1 İÇİN MAKİNELERİN DOLU, BOŞ VE ARIZA ZAMANLARI VE YÜZDELERİ

Identifier	Category	AvgTime	Percent
STATE(jumper2)	BUSY	0,96463	90,9
	IDLE	0,07195	6,73
	FAILED	2,8453	2,37
STATE(percın6)	BUSY	0,36505	44,79
	IDLE	0,43947	53,65
	FAILED	2,0334	1,55
STATE(jumper3)	BUSY	0,97537	91,03
	IDLE	0,07649	7,08
	FAILED	2,7146	1,89
STATE(jumper4)	BUSY	0,98375	90,86
	IDLE	0,07641	6,99
	FAILED	2,5757	2,15
STATE(jumper5)	BUSY	1,0737	91,04
	IDLE	0,08606	7,24
	FAILED	2,4743	1,72
STATE(jumper6)	BUSY	1,1989	91
	IDLE	0,09322	7,01
	FAILED	2,8607	1,99
STATE(axial1)	BUSY	0,45213	53,94
	IDLE	0,36752	43,67
	FAILED	2,2932	2,39
STATE(sequential1)	BUSY	0,44438	53,73
	IDLE	0,36775	44,39
	FAILED	1,9408	1,89
STATE(axial2)	BUSY	0,44693	53,94
	IDLE	0,36791	44,05
	FAILED	1,9306	2,01
STATE(sequential2)	BUSY	0,4451	53,72
	IDLE	0,36766	44,25
	FAILED	2,0901	2,03
STATE(axial3)	BUSY	0,4484	53,93
	IDLE	0,36838	44,05
	FAILED	1,9353	2,02

STATE(radial1)	BUSY	0,69273	61,24
	IDLE	0,41038	35,97
	FAILED	2,3676	2,8
STATE(axial4)	BUSY	0,45258	53,93
	IDLE	0,37066	43,89
	FAILED	2,2432	2,18
STATE(smd1)	BUSY	0,78919	64,01
	IDLE	0,37852	30,23
	FAILED	2,9613	5,76
STATE(radial2)	BUSY	0,68958	61,25
	IDLE	0,40949	36,09
	FAILED	2,5591	2,67
STATE(axial5)	BUSY	0,45416	53,93
	IDLE	0,37132	43,81
	FAILED	2,3221	2,26
STATE(smd2)	BUSY	0,75246	64,01
	IDLE	0,37677	31,79
	FAILED	2,8792	4,2
STATE(radial3)	BUSY	0,68833	61,23
	IDLE	0,41051	36,26
	FAILED	2,1225	2,51
STATE(axial6)	BUSY	0,45153	53,93
	IDLE	0,37135	44,17
	FAILED	1,9471	1,89
STATE(sequencer3)	BUSY	0,44077	53,72
	IDLE	0,36751	44,71
	FAILED	1,6122	1,57
STATE(smd3)	BUSY	0,73618	64,06
	IDLE	0,38021	32,71
	FAILED	2,5828	3,23
STATE(radial4)	BUSY	0,69079	61,21
	IDLE	0,41043	36
	FAILED	2,3636	2,79
STATE(sequencer4)	BUSY	0,44484	53,72
	IDLE	0,36888	44,34
	FAILED	1,9931	1,94
STATE(smd4)	BUSY	0,79113	64,06

	IDLE	0,37691	29,92
	FAILED	2,891	6,02
STATE(radial5)	BUSY	0,69278	61,2
	IDLE	0,40865	35,87
	FAILED	2,3471	2,93
STATE(sequencer5)	BUSY	0,44611	53,72
	IDLE	0,36883	44,29
	FAILED	2,3951	2
STATE(perc1n1)	BUSY	0,3642	44,79
	IDLE	0,43929	53,87
	FAILED	1,7462	1,33
STATE(smd5)	BUSY	0,74572	64,06
	IDLE	0,37948	32,2
	FAILED	2,6913	3,74
STATE(radial6)	BUSY	0,69276	61,19
	IDLE	0,41173	36,11
	FAILED	2,2819	2,69
STATE(sequencer6)	BUSY	0,44255	53,72
	IDLE	0,36783	44,5
	FAILED	1,9746	1,78
STATE(perc2n2)	BUSY	0,36635	44,78
	IDLE	0,43883	53,57
	FAILED	1,9795	1,65
STATE(smd6)	BUSY	0,75487	64,06
	IDLE	0,37854	31,78
	FAILED	2,6044	4,16
STATE(radial7)	BUSY	0,69721	61,2
	IDLE	0,41253	35,84
	FAILED	2,5086	2,96
STATE(perc3n3)	BUSY	0,36729	44,79
	IDLE	0,43953	53,45
	FAILED	2,3109	1,77
STATE(smd7)	BUSY	0,75559	64,02
	IDLE	0,38321	32,17
	FAILED	2,4946	3,81
STATE(radial8)	BUSY	0,69567	61,21
	IDLE	0,41346	36,12
	FAILED	2,2618	2,67

STATE(perc4)	BUSY	0,36553	44,8
	IDLE	0,43963	53,73
	FAILED	1,9185	1,47
STATE(smd8)	BUSY	0,76944	64,01
	IDLE	0,37619	31,04
	FAILED	2,7422	4,95
STATE(jumper1)	BUSY	0,97709	91,06
	IDLE	0,07139	6,6
	FAILED	3,0662	2,34
STATE(perc5)	BUSY	0,36478	44,79
	IDLE	0,43916	53,8
	FAILED	1,8543	1,42



EK 2C ÖNERİ 2 İÇİN MAKİNELERİN DOLU, BOŞ VE ARIZA ZAMANLARI VE YÜZDELERİ

Identifier	Category	AvgTime	Percent
STATE(perc6)	BUSY	0,36367	49,1
	IDLE	0,37031	49,94
	FAILED	1,7348	0,96
STATE(jumper2)	BUSY	12,611	98,97
	IDLE	0,00895	0,07
	FAILED	1,9856	0,97
STATE(jumper3)	BUSY	14,594	99,32
	IDLE	0,01506	0,1
	FAILED	1,6772	0,58
STATE(jumper4)	BUSY	12,741	99,1
	IDLE	0,01015	0,07
	FAILED	1,7048	0,83
STATE(jumper5)	BUSY	12,666	99,4
	IDLE	0,01036	0,08
	FAILED	1,5191	0,53
STATE(jumper6)	BUSY	13,582	99,04
	IDLE	0,01153	0,08
	FAILED	1,5864	0,88
STATE(sequential1)	BUSY	0,44342	58,57
	IDLE	0,30258	39,82
	FAILED	2,3222	1,61
STATE(axial1)	BUSY	0,44106	58,81
	IDLE	0,30079	40
	FAILED	1,5591	1,19
STATE(sequential2)	BUSY	0,44059	58,56
	IDLE	0,30344	40,18
	FAILED	1,8047	1,25
STATE(axial2)	BUSY	0,44161	58,79
	IDLE	0,30159	39,98
	FAILED	1,7695	1,23
STATE(axial3)	BUSY	0,44392	58,79
	IDLE	0,30165	39,84
	FAILED	1,7901	1,37
STATE(axial4)	BUSY	0,44161	58,79
	IDLE	0,30155	40,06

	FAILED	1,506	1,15
STATE(radial1)	BUSY	0,69643	66,74
	IDLE	0,32532	30,88
	FAILED	2,4439	2,38
STATE(axial5)	BUSY	0,44416	58,79
	IDLE	0,30264	39,91
	FAILED	2,0807	1,3
STATE(radial2)	BUSY	0,69897	66,74
	IDLE	0,32421	30,75
	FAILED	2,5745	2,5
STATE(smd1)	BUSY	0,75291	69,8
	IDLE	0,29667	27,36
	FAILED	3,1457	2,84
STATE(sequencer3)	BUSY	0,43922	58,56
	IDLE	0,30276	40,26
	FAILED	1,69	1,17
STATE(axial6)	BUSY	0,44863	58,79
	IDLE	0,30182	39,47
	FAILED	2,2827	1,74
STATE(radial3)	BUSY	0,6922	66,72
	IDLE	0,32517	31,09
	FAILED	2,4204	2,19
STATE(smd2)	BUSY	0,76201	69,8
	IDLE	0,29597	26,88
	FAILED	2,9861	3,32
STATE(sequencer4)	BUSY	0,44474	58,56
	IDLE	0,3035	39,81
	FAILED	2,1327	1,63
STATE(radial4)	BUSY	0,68688	66,73
	IDLE	0,32565	31,32
	FAILED	2,0025	1,95
STATE(smd3)	BUSY	0,74075	69,81
	IDLE	0,29607	27,74
	FAILED	2,9501	2,46
STATE(perc1n1)	BUSY	0,36404	49,09
	IDLE	0,37022	49,85
	FAILED	1,6864	1,05
STATE(sequencer5)	BUSY	0,44239	58,56

	IDLE	0,30295	40,02
	FAILED	2,0583	1,43
STATE(radial5)	BUSY	0,68151	66,73
	IDLE	0,32372	31,52
	FAILED	1,9401	1,75
STATE(smd4)	BUSY	0,71846	69,8
	IDLE	0,29628	28,64
	FAILED	2,8063	1,56
STATE(percin2)	BUSY	0,36632	49,1
	IDLE	0,36999	49,54
	FAILED	2,1846	1,37
STATE(sequencer6)	BUSY	0,43983	58,55
	IDLE	0,30291	40,16
	FAILED	1,6892	1,29
STATE(radial6)	BUSY	0,69775	66,72
	IDLE	0,32476	30,87
	FAILED	2,6624	2,4
STATE(smd5)	BUSY	0,74233	69,8
	IDLE	0,29383	27,49
	FAILED	2,7925	2,71
STATE(percin3)	BUSY	0,36544	49,11
	IDLE	0,37073	49,64
	FAILED	2,0111	1,26
STATE(radial7)	BUSY	0,69743	66,74
	IDLE	0,32497	30,89
	FAILED	2,4329	2,37
STATE(smd6)	BUSY	0,73148	69,8
	IDLE	0,29731	28,18
	FAILED	2,9112	2,02
STATE(percin4)	BUSY	0,365	49,1
	IDLE	0,37013	49,71
	FAILED	1,907	1,19
STATE(radial8)	BUSY	0,69608	66,71
	IDLE	0,32522	31,05
	FAILED	2,6851	2,24
STATE(smd7)	BUSY	0,7173	69,79
	IDLE	0,29594	28,67
	FAILED	2,4689	1,54

STATE(percin5)	BUSY	0,3642	49,09
	IDLE	0,37021	49,75
	FAILED	1,8592	1,16
STATE(jumper1)	BUSY	12,016	99,3
	IDLE	0,00367	0,03
	FAILED	1,6075	0,67
STATE(smd8)	BUSY	0,72713	69,78
	IDLE	0,29787	28,42
	FAILED	2,5831	1,79



EK 3 SİMÜLASYON PROGRAMI

```

;
; Model statements for module: Arrive 1
;
172$ CREATE, 6:0.75,2000;

133$ STATION, Arrive 1;
181$ TRACE, -1,"-Arrived to system at station Arrive 1\n";
136$ ASSIGN: Picture=Default;
157$ DELAY: 0.;
185$ TRACE, -1,"-Transferred to next module\n":NEXT(0$);

;
;
; Model statements for module: Enter 1
;
0$ STATION, Processing;
221$ TRACE, -1,"-Arrived to station Processing\n";
201$ STORE: Processing_S1;
200$ DELAY: 0.;
215$ UNSTORE;
220$ DELAY: 0.000:NEXT(1$);

;
;
; Model statements for module: PickQueue 1
;
1$ TRACE, -1,"-Selecting between 6 queues\n";
229$ PICKQ, CYC:
        3$:
        6$:
        9$:
        12$:
        15$:
        18$;

;
;
; Model statements for module: Process 9
;
3$ QUEUE, percin1_Q:MARK(QueueTime);
230$ SEIZE, 1:
        percin1,1;

```

```

276$    TALLY:    perc1n1_Q Queue Time,INT(QueueTime),1;
        TRACE,    -1,"-Delay for processing time NORM( 0.358333333 ,
0.002033333)\n";
243$    DELAY:    NORM( 0.358333333 , 0.002033333);
259$    ASSIGN:    Picture=perc1n;
250$    RELEASE:    perc1n1,1;
294$    DELAY:    0.000;
304$    DELAY:    0.0:NEXT(20$);

```

```

;
;
; Model statements for module: Enter 2
;

```

```

20$    STATION,    Processing2;
336$    TRACE,    -1,"-Arrived to station Processing2\n";
316$    STORE:    Processing2_S1;
315$    DELAY:    0.;
330$    UNSTORE;
335$    DELAY:    0.000:NEXT(21$);

```

```

;
;
; Model statements for module: PickQueue 2
;

```

```

21$    TRACE,    -1,"-Selecting between 6 queues\n";
344$    PICKQ,    CYC:
                23$:
                26$:
                29$:
                32$:
                35$:
                38$;

```

```

;
;
; Model statements for module: Process 19
;

```

```

23$    QUEUE,    jumper1_Q:MARK(QueueTime);
345$    SEIZE,    1:
                jumper1,1;
391$    TALLY:    jumper1_Q Queue Time,INT(QueueTime),1;
        TRACE,    -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
358$    DELAY:    NORM( 0.72833,0.0038);
374$    ASSIGN:    Picture=jumper;
365$    RELEASE:    jumper1,1;

```



```

409$    DELAY:    0.000;
419$    DELAY:    0.0:NEXT(40$);

```

```

;
;
;

```

```

; Model statements for module: Enter 3
;

```

```

40$    STATION,   Processing3;
451$   TRACE,    -1,"-Arrived to station Processing3\n";
431$   STORE:    Processing3_S1;
430$   DELAY:    0.;
445$   UNSTORE;
450$   DELAY:    0.000:NEXT(41$);

```

```

;
;
;

```

```

; Model statements for module: PickQueue 3
;

```

```

41$    TRACE,    -1,"-Selecting between 6 queues\n";
459$   PICKQ,    CYC:
          58$:
          43$:
          46$:
          49$:
          52$:
          55$;

```

```

;
;
;

```

```

; Model statements for module: Process 26
;

```

```

58$    QUEUE,    sequential1_Q:MARK(QueueTime);
460$   SEIZE,    1:
          sequential1,1;
506$   TALLY:    sequential1_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
473$   DELAY:    NORM( 0.43, 0.0002);
489$   ASSIGN:   Picture=sequencer;
480$   RELEASE:  sequential1,1;
524$   DELAY:    0.000;
534$   DELAY:    0.0:NEXT(60$);

```

```

;
;

```

```
; Model statements for module: Enter 4
;
```

```
60$ STATION, Processing4;
566$ TRACE, -1,"-Arrived to station Processing4\n";
546$ STORE: Processing4_S1;
545$ DELAY: 0.;
560$ UNSTORE;
565$ DELAY: 0.000:NEXT(61$);
```

```
;
;
; Model statements for module: PickQueue 4
;
```

```
61$ TRACE, -1,"-Selecting between 6 queues\n";
574$ PICKQ, CYC:
        63$:
        66$:
        69$:
        72$:
        75$:
        78$;
```

```
;
;
; Model statements for module: Process 27
;
```

```
63$ QUEUE, axial1_Q:MARK(QueueTime);
575$ SEIZE, 1:
        axial1,1;
621$ TALLY: axial1_Q Queue Time,INT(QueueTime),1;
TRACE, -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
588$ DELAY: NORM( 0.4317,0.00002);
604$ ASSIGN: Picture=axial;
595$ RELEASE: axial1,1;
639$ DELAY: 0.000;
649$ DELAY: 0.0:NEXT(80$);
```

```
;
;
; Model statements for module: Enter 5
;
```

```
80$ STATION, Processing5;
681$ TRACE, -1,"-Arrived to station Processing5\n";
661$ STORE: Processing5_S1;
```

```

660$    DELAY:    0.;
675$    UNSTORE;
680$    DELAY:    0.000:NEXT(81$);

```

```

;
;
;

```

```

; Model statements for module: PickQueue 5
;

```

```

81$    TRACE,    -1,"-Selecting between 8 queues\n";
689$    PICKQ,    CYC:
          83$:
          86$:
          89$:
          92$:
          95$:
          98$:
          101$:
          104$;

```

```

;
;
;

```

```

; Model statements for module: Process 33
;

```

```

83$    QUEUE,    radial1_Q:MARK(QueueTime);
690$    SEIZE,    1:
          radial1,1;
736$    TALLY:    radial1_Q Queue Time,INT(QueueTime),1;
          TRACE,    -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
703$    DELAY:    NORM( 0.6533,0.0026);
719$    ASSIGN:    Picture=radial;
710$    RELEASE:    radial1,1;
754$    DELAY:    0.000;
764$    DELAY:    0.0:NEXT(106$);

```

```

;
;
;

```

```

; Model statements for module: Enter 6
;

```

```

106$    STATION,    Processing6;
796$    TRACE,    -1,"-Arrived to station Processing6\n";
776$    STORE:    Processing6_S1;
775$    DELAY:    0.;
790$    UNSTORE;
795$    DELAY:    0.000:NEXT(107$);

```

```

;
;
;   Model statements for module: PickQueue 6
;
107$   TRACE,      -1,"-Selecting between 8 queues\n";
804$   PICKQ,      CYC:
        124$:
        109$:
        112$:
        115$:
        118$:
        121$:
        127$:
        130$;

```

```

;
;
;   Model statements for module: Process 41
;
124$   QUEUE,      smd1_Q:MARK(QueueTime);
805$   SEIZE,      1:
        smd1,1;
851$   TALLY:      smd1_Q Queue Time,INT(QueueTime),1;
        TRACE,      -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
818$   DELAY:      NORM( 0.6833,0.0003);
834$   ASSIGN:     Picture=smd;
825$   RELEASE:    smd1,1;
869$   DELAY:      0.000;
879$   DELAY:      0.0:NEXT(132$);

```

```

;
;
;   Model statements for module: Depart 4
;
132$   STATION,    Depart 4;
920$   TRACE,      -1,"-Arrived to station Depart 4\n";
890$   DELAY:      0.;
912$   COUNT:      Depart 4_C,1;
927$   TRACE,      -1,"-Disposing entity\n";
919$   DISPOSE;

```

```

;
;
;   Model statements for module: Process 42

```

```

;
109$   QUEUE,      smd2_Q:MARK(QueueTime);
929$   SEIZE,      1:
          smd2,1;
975$   TALLY:      smd2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
942$   DELAY:      NORM( 0.6833,0.0003);
958$   ASSIGN:     Picture=smd;
949$   RELEASE:    smd2,1;
993$   DELAY:      0.000;
1003$  DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 43
;

```

```

112$   QUEUE,      smd3_Q:MARK(QueueTime);
1014$  SEIZE,      1:
          smd3,1;
1060$  TALLY:      smd3_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1027$  DELAY:      NORM( 0.6833,0.0003);
1043$  ASSIGN:     Picture=smd;
1034$  RELEASE:    smd3,1;
1078$  DELAY:      0.000;
1088$  DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 44
;

```

```

115$   QUEUE,      smd4_Q:MARK(QueueTime);
1099$  SEIZE,      1:
          smd4,1;
1145$  TALLY:      smd4_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1112$  DELAY:      NORM( 0.6833,0.0003);
1128$  ASSIGN:     Picture=smd;
1119$  RELEASE:    smd4,1;
1163$  DELAY:      0.000;
1173$  DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 45

```

```

;
118$    QUEUE,      smd5_Q:MARK(QueueTime);
1184$   SEIZE,      1:
          smd5,1;
1230$   TALLY:      smd5_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1197$   DELAY:      NORM( 0.6833,0.0003);
1213$   ASSIGN:     Picture=smd;
1204$   RELEASE:    smd5,1;
1248$   DELAY:      0.000;
1258$   DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 46
;

```

```

121$    QUEUE,      smd6_Q:MARK(QueueTime);
1269$   SEIZE,      1:
          smd6,1;
1315$   TALLY:      smd6_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1282$   DELAY:      NORM( 0.6833,0.0003);
1298$   ASSIGN:     Picture=smd;
1289$   RELEASE:    smd6,1;
1333$   DELAY:      0.000;
1343$   DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 47
;

```

```

127$    QUEUE,      smd7_Q:MARK(QueueTime);
1354$   SEIZE,      1:
          smd7,1;
1400$   TALLY:      smd7_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1367$   DELAY:      NORM( 0.6833,0.0003);
1383$   ASSIGN:     Picture=smd;
1374$   RELEASE:    smd7,1;
1418$   DELAY:      0.000;
1428$   DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 48

```

```

;
130$    QUEUE,      smd8_Q:MARK(QueueTime);
1439$   SEIZE,      1:
          smd8,1;
1485$   TALLY:      smd8_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6833,0.0003)\n";
1452$   DELAY:      NORM( 0.6833,0.0003);
1468$   ASSIGN:     Picture=smd;
1459$   RELEASE:    smd8,1;
1503$   DELAY:      0.000;
1513$   DELAY:      0.0:NEXT(132$);

```

```

;
;
; Model statements for module: Process 34
;

```

```

86$     QUEUE,      radial2_Q:MARK(QueueTime);
1524$   SEIZE,      1:
          radial2,1;
1570$   TALLY:      radial2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1537$   DELAY:      NORM( 0.6533,0.0026);
1553$   ASSIGN:     Picture=radial;
1544$   RELEASE:    radial2,1;
1588$   DELAY:      0.000;
1598$   DELAY:      0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 35
;

```

```

89$     QUEUE,      radial3_Q:MARK(QueueTime);
1609$   SEIZE,      1:
          radial3,1;
1655$   TALLY:      radial3_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1622$   DELAY:      NORM( 0.6533,0.0026);
1638$   ASSIGN:     Picture=radial;
1629$   RELEASE:    radial3,1;
1673$   DELAY:      0.000;
1683$   DELAY:      0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 36

```

```

;
92$    QUEUE,    radial4_Q:MARK(QueueTime);
1694$  SEIZE,    1:
        radial4,1;
1740$  TALLY:    radial4_Q Queue Time,INT(QueueTime),1;
        TRACE,   -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1707$  DELAY:    NORM( 0.6533,0.0026);
1723$  ASSIGN:    Picture=radial;
1714$  RELEASE:   radial4,1;
1758$  DELAY:    0.000;
1768$  DELAY:    0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 37
;

```

```

95$    QUEUE,    radial5_Q:MARK(QueueTime);
1779$  SEIZE,    1:
        radial5,1;
1825$  TALLY:    radial5_Q Queue Time,INT(QueueTime),1;
        TRACE,   -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1792$  DELAY:    NORM( 0.6533,0.0026);
1808$  ASSIGN:    Picture=radial;
1799$  RELEASE:   radial5,1;
1843$  DELAY:    0.000;
1853$  DELAY:    0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 38
;

```

```

98$    QUEUE,    radial6_Q:MARK(QueueTime);
1864$  SEIZE,    1:
        radial6,1;
1910$  TALLY:    radial6_Q Queue Time,INT(QueueTime),1;
        TRACE,   -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1877$  DELAY:    NORM( 0.6533,0.0026);
1893$  ASSIGN:    Picture=radial;
1884$  RELEASE:   radial6,1;
1928$  DELAY:    0.000;
1938$  DELAY:    0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 39

```



```

;
101$    QUEUE,    radial7_Q:MARK(QueueTime);
1949$   SEIZE,    1:
          radial7,1;
1995$   TALLY:    radial7_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
1962$   DELAY:    NORM( 0.6533,0.0026);
1978$   ASSIGN:    Picture=radial;
1969$   RELEASE:  radial7,1;
2013$   DELAY:    0.000;
2023$   DELAY:    0.0:NEXT(106$);

```

```

;
;
;
; Model statements for module: Process 40
;

```

```

104$    QUEUE,    radial8_Q:MARK(QueueTime);
2034$   SEIZE,    1:
          radial8,1;
2080$   TALLY:    radial8_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.6533,0.0026)\n";
2047$   DELAY:    NORM( 0.6533,0.0026);
2063$   ASSIGN:    Picture=radial;
2054$   RELEASE:  radial8,1;
2098$   DELAY:    0.000;
2108$   DELAY:    0.0:NEXT(106$);

```

```

;
;
; Model statements for module: Process 28
;

```

```

66$     QUEUE,    axial2_Q:MARK(QueueTime);
2119$   SEIZE,    1:
          axial2,1;
2165$   TALLY:    axial2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
2132$   DELAY:    NORM( 0.4317,0.00002);
2148$   ASSIGN:    Picture=axial;
2139$   RELEASE:  axial2,1;
2183$   DELAY:    0.000;
2193$   DELAY:    0.0:NEXT(80$);

```

```

;
;
; Model statements for module: Process 29

```

```

;
69$    QUEUE,    axial3_Q:MARK(QueueTime);
2204$  SEIZE,    1:
        axial3,1;
2250$  TALLY:    axial3_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
2217$  DELAY:    NORM( 0.4317,0.00002);
2233$  ASSIGN:    Picture=axial;
2224$  RELEASE:   axial3,1;
2268$  DELAY:    0.000;
2278$  DELAY:    0.0:NEXT(80$);

```

```

;
;
; Model statements for module: Process 30
;

```

```

72$    QUEUE,    axial4_Q:MARK(QueueTime);
2289$  SEIZE,    1:
        axial4,1;
2335$  TALLY:    axial4_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
2302$  DELAY:    NORM( 0.4317,0.00002);
2318$  ASSIGN:    Picture=axial;
2309$  RELEASE:   axial4,1;
2353$  DELAY:    0.000;
2363$  DELAY:    0.0:NEXT(80$);

```

```

;
;
; Model statements for module: Process 31
;

```

```

75$    QUEUE,    axial5_Q:MARK(QueueTime);
2374$  SEIZE,    1:
        axial5,1;
2420$  TALLY:    axial5_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
2387$  DELAY:    NORM( 0.4317,0.00002);
2403$  ASSIGN:    Picture=axial;
2394$  RELEASE:   axial5,1;
2438$  DELAY:    0.000;
2448$  DELAY:    0.0:NEXT(80$);

```

```

;
;
; Model statements for module: Process 32

```

```

;
78$    QUEUE,    axial6_Q:MARK(QueueTime);
2459$  SEIZE,    1:
        axial6,1;
2505$  TALLY:    axial6_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.4317,0.00002)\n";
2472$  DELAY:    NORM( 0.4317,0.00002);
2488$  ASSIGN:    Picture=axial;
2479$  RELEASE:   axial6,1;
2523$  DELAY:    0.000;
2533$  DELAY:    0.0:NEXT(80$);

```

```

;
;
; Model statements for module: Process 25
;

```

```

43$    QUEUE,    sequential2_Q:MARK(QueueTime);
2544$  SEIZE,    1:
        sequential2,1;
2590$  TALLY:    sequential2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
2557$  DELAY:    NORM( 0.43, 0.0002);
2573$  ASSIGN:    Picture=sequencer;
2564$  RELEASE:   sequential2,1;
2608$  DELAY:    0.000;
2618$  DELAY:    0.0:NEXT(60$);

```

```

;
;
; Model statements for module: Process 21
;

```

```

46$    QUEUE,    sequencer3_Q:MARK(QueueTime);
2629$  SEIZE,    1:
        sequencer3,1;
2675$  TALLY:    sequencer3_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
2642$  DELAY:    NORM( 0.43, 0.0002);
2658$  ASSIGN:    Picture=sequencer;
2649$  RELEASE:   sequencer3,1;
2693$  DELAY:    0.000;
2703$  DELAY:    0.0:NEXT(60$);

```

```

;
;
; Model statements for module: Process 22

```

```

;
49$    QUEUE,    sequencer4_Q:MARK(QueueTime);
2714$  SEIZE,    1:
                sequencer4,1;
2760$  TALLY:    sequencer4_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
2727$  DELAY:    NORM( 0.43, 0.0002);
2743$  ASSIGN:    Picture=sequencer;
2734$  RELEASE:  sequencer4,1;
2778$  DELAY:    0.000;
2788$  DELAY:    0.0:NEXT(60$);

```

```

;
;
; Model statements for module: Process 23
;

```

```

52$    QUEUE,    sequencer5_Q:MARK(QueueTime);
2799$  SEIZE,    1:
                sequencer5,1;
2845$  TALLY:    sequencer5_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
2812$  DELAY:    NORM( 0.43, 0.0002);
2828$  ASSIGN:    Picture=sequencer;
2819$  RELEASE:  sequencer5,1;
2863$  DELAY:    0.000;
2873$  DELAY:    0.0:NEXT(60$);

```

```

;
;
; Model statements for module: Process 24
;

```

```

55$    QUEUE,    sequencer6_Q:MARK(QueueTime);
2884$  SEIZE,    1:
                sequencer6,1;
2930$  TALLY:    sequencer6_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.43, 0.0002)\n";
2897$  DELAY:    NORM( 0.43, 0.0002);
2913$  ASSIGN:    Picture=sequencer;
2904$  RELEASE:  sequencer6,1;
2948$  DELAY:    0.000;
2958$  DELAY:    0.0:NEXT(60$);

```

```

;
;
; Model statements for module: Process 15

```

```

;
26$   QUEUE,    jumper2_Q:MARK(QueueTime);
2969$ SEIZE,    1:
      jumper2,1;
3015$ TALLY:    jumper2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
2982$ DELAY:    NORM( 0.72833,0.0038);
2998$ ASSIGN:   Picture=jumper;
2989$ RELEASE:  jumper2,1;
3033$ DELAY:    0.000;
3043$ DELAY:    0.0:NEXT(40$);

```

```

;
;
; Model statements for module: Process 16
;

```

```

29$   QUEUE,    jumper3_Q:MARK(QueueTime);
3054$ SEIZE,    1:
      jumper3,1;
3100$ TALLY:    jumper3_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
3067$ DELAY:    NORM( 0.72833,0.0038);
3083$ ASSIGN:   Picture=jumper;
3074$ RELEASE:  jumper3,1;
3118$ DELAY:    0.000;
3128$ DELAY:    0.0:NEXT(40$);

```

```

;
;
; Model statements for module: Process 17
;

```

```

32$   QUEUE,    jumper4_Q:MARK(QueueTime);
3139$ SEIZE,    1:
      jumper4,1;
3185$ TALLY:    jumper4_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
3152$ DELAY:    NORM( 0.72833,0.0038);
3168$ ASSIGN:   Picture=jumper;
3159$ RELEASE:  jumper4,1;
3203$ DELAY:    0.000;
3213$ DELAY:    0.0:NEXT(40$);

```

```

;
;
; Model statements for module: Process 18

```

```

;
35$   QUEUE,    jumper5_Q:MARK(QueueTime);
3224$ SEIZE,    1:
        jumper5,1;
3270$ TALLY:    jumper5_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
3237$ DELAY:    NORM( 0.72833,0.0038);
3253$ ASSIGN:   Picture=jumper;
3244$ RELEASE:  jumper5,1;
3288$ DELAY:    0.000;
3298$ DELAY:    0.0:NEXT(40$);

```

```

;
;
;   Model statements for module: Process 20
;

```

```

38$   QUEUE,    jumper6_Q:MARK(QueueTime);
3309$ SEIZE,    1:
        jumper6,1;
3355$ TALLY:    jumper6_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.72833,0.0038)\n";
3322$ DELAY:    NORM( 0.72833,0.0038);
3338$ ASSIGN:   Picture=jumper;
3329$ RELEASE:  jumper6,1;
3373$ DELAY:    0.000;
3383$ DELAY:    0.0:NEXT(40$);

```

```

;
;
;   Model statements for module: Process 10
;

```

```

6$   QUEUE,    perc2_Q:MARK(QueueTime);
3394$ SEIZE,    1:
        perc2,1;
3440$ TALLY:    perc2_Q Queue Time,INT(QueueTime),1;
TRACE,  -1,"-Delay for processing time NORM( 0.358333333,
0.002033333 )\n";
3407$ DELAY:    NORM( 0.358333333, 0.002033333 );
3423$ ASSIGN:   Picture=perc2;
3414$ RELEASE:  perc2,1;
3458$ DELAY:    0.000;
3468$ DELAY:    0.0:NEXT(20$);

```

```

;
;

```

```

; Model statements for module: Process 11
;
9$ QUEUE, perc3_Q:MARK(QueueTime);
3479$ SEIZE, 1:
      perc3,1;
3525$ TALLY: perc3_Q Queue Time,INT(QueueTime),1;
      TRACE, -1,"-Delay for processing time NORM( 0.358333333,
0.002033333)\n";
3492$ DELAY: NORM( 0.358333333, 0.002033333);
3508$ ASSIGN: Picture=perc3;
3499$ RELEASE: perc3,1;
3543$ DELAY: 0.000;
3553$ DELAY: 0.0:NEXT(20$);

```

```

;
;
; Model statements for module: Process 12
;
12$ QUEUE, perc4_Q:MARK(QueueTime);
3564$ SEIZE, 1:
      perc4,1;
3610$ TALLY: perc4_Q Queue Time,INT(QueueTime),1;
      TRACE, -1,"-Delay for processing time NORM(
0.358333333,0.002033333)\n";
3577$ DELAY: NORM( 0.358333333,0.002033333);
3593$ ASSIGN: Picture=perc4;
3584$ RELEASE: perc4,1;
3628$ DELAY: 0.000;
3638$ DELAY: 0.0:NEXT(20$);

```

```

;
;
; Model statements for module: Process 13
;
15$ QUEUE, perc5_Q:MARK(QueueTime);
3649$ SEIZE, 1:
      perc5,1;
3695$ TALLY: perc5_Q Queue Time,INT(QueueTime),1;
      TRACE, -1,"-Delay for processing time NORM( 0.358333333,
0.002033333 )\n";
3662$ DELAY: NORM( 0.358333333, 0.002033333 );
3678$ ASSIGN: Picture=perc5;
3669$ RELEASE: perc5,1;
3713$ DELAY: 0.000;
3723$ DELAY: 0.0:NEXT(20$);

```

```
;
;
;   Model statements for module: Process 14
;
18$   QUEUE,      perc6_Q:MARK(QueueTime);
3734$ SEIZE,      1:
      perc6,1;
3780$ TALLY:      perc6_Q Queue Time,INT(QueueTime),1;
      TRACE,      -1,"-Delay for processing time NORM( 0.358333333,
0.002033333 )\n";
3747$ DELAY:      NORM( 0.358333333, 0.002033333 );
3763$ ASSIGN:     Picture=perc6;
3754$ RELEASE:    perc6,1;
3798$ DELAY:      0.000;
3808$ DELAY:      0.0:NEXT(20$);
```