

**Konfeksiyonda Simülasyon Tekniđiyle Yalın
Üretim Sistemlerinin İncelenmesi**

Selin BİLGET

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Danışman: Doç. Dr. Can ÜNAL

2015

T.C.
NAMIK KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

**KONFEKSİYONDA SİMÜLASYON TEKNİĞİYLE YALIN ÜRETİM
SİSTEMLERİNİN İNCELENMESİ**

Selin BİLGET

TEKSTİL MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: Doç. Dr. Can ÜNAL

TEKİRDAĞ-2015

Her hakkı saklıdır

Bu tez NKUBAP tarafından 00.17.AR.12.12 numaralı proje ile desteklenmiştir.

Doç. Dr. Can ÜNAL danışmanlığında, Selin BİLGET tarafından hazırlanan “Konfeksiyonda Simülasyon Tekniğiyle Yalın Üretim Sistemlerinin İncelenmesi” isimli bu çalışma aşağıdaki jüri tarafından Tekstil Mühendisliği Anabilim Dalı’nda Yüksek Lisans tezi olarak oy birliği ile kabul edilmiştir.

Juri Başkanı : Prof. Dr. H. Ziya ÖZEK

İmza :

Üye : Doç. Dr. Can ÜNAL

İmza :

Üye : Yrd. Doç. Dr. Senem KURŞUN BAHADIR

İmza :

Fen Bilimleri Enstitüsü Yönetim Kurulu adına

Prof. Dr. Fatih KONUKCU

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

Konfeksiyonda Simülasyon Tekniğiyle Yalın Üretim Sistemlerinin İncelenmesi

Selin BİLGET

Namık Kemal Üniversitesi

Fen Bilimleri Enstitüsü

Tekstil Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Can ÜNAL

Günümüzde uluslararası rekabet hızla büyümektedir. Firmalar çağın dinamik yapısına ayak uydurmak için, yeni sistemleri, teknikleri ve teknolojileri bünyelerine adapte etmek zorundadırlar. Özellikle konfeksiyon sektörü, sipariş adetlerinin azalması ve ürün çeşitliliğinin artması nedeniyle geçmişteki çok adetli, az çeşitli siparişleri karşılamaya yönelik üretim yapısından sıyrılmakta, üretim bantlarını yalın üretim prensiplerine göre oluşturma çabası içerisinde. Ancak sistem değişikliklerinin en dikkatli şekilde planının yapılması bile bazen analizcileri büyük hata yapmaktan alıkoyamamaktadır. Bu yapılan hatalara makine kapasitelerinin yanlış eşleştirilmesi, beklemedeki parçalar için yetersiz alanın ayrılması örnek olarak verilebilir. Bu çalışmada, konfeksiyonda yalın üretim yapılan bir işletmede, üç farklı ürünün üretim süreçleri incelenerek üretim süreleri belirlenmiş, sürelerin gösterdiği istatistiksel dağılım göz önüne alınarak, simülasyon modelleri oluşturulmuştur. Modelin geçerliliği günlük üretim adedi baz alınarak istatistiksel olarak kontrol edilmiştir. Daha sonra yalın üretim gereklerine bağlı olarak mevcut modeller yeniden tasarlanıp SIMUL8 yazılımında alternatif modeller oluşturulup incelenmiştir. Bunun yanı sıra, simülasyon uygulamasıyla hat dengelemeyi sağlayacak yeni bir algoritma geliştirilmiştir. “Kalp Algoritması” adı verilen uygulama, tüm modeller üzerinde denenmiş ve kişi başı üretimi arttıracak, olumlu geri dönüşler alınmıştır.

Anahtar Kelimeler: konfeksiyon, simülasyon, hat dengeleme, yalın üretim, kalp algoritması.

2015, 100 sayfa

ABSTRACT

MSc. Thesis

EXAMINATION OF LEAN MANUFACTURING SYSTEMS BY SIMULATION TECHNIQUE IN APPAREL INDUSTRY

Selin BİLGET

Namık Kemal Üniversitesi

Graduate School of Natural and Applied Sciences

Department of Textile Engineering

Supervisor: Assoc. Prof. Dr. Can ÜNAL

Today, international competition is rapidly growing. In order to keep pace with the dynamic nature of time, companies are required to adapt new systems techniques and technologies to their structure. Especially apparel industry is stripped from the former structure of bulk production and is in an effort to form their production line according to lean manufacturing principles, due to the increasing number of orders and product diversity. On the other hand, even making the most careful plans cannot prevent the analysts to make big mistakes sometimes. Examples of these mistakes can be given as mismatch of machine capacities and allocation of insufficient space for items in queue. In this study, production time of three different products are determined by examining their production process and simulation models of these products are created by implementing statistical distribution of production times in an apparel company which is considering lean manufacturing principles. The validation of simulation models are done on the basis of total daily production statistically. Afterwards, recent models are redesigned according to lean production principles, alternative models are created and examined in SIMUL8 software. In addition, a new algorithm which allows line balancing in simulation applications is developed. "Heart Algorithm" called application tested on all models and positive feedback which increase the production per person has received.

Keywords: apparel industry, simulation, line balancing, lean production, heart algorithm.

2015, 100 pages

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ÇİZELGE DİZİNİ	v
ŞEKİL DİZİNİ	vi
ÖNSÖZ	ix
1. GİRİŞ	1
2. KURAMSAL TEMELLER	5
2.1 Literatür Taraması	5
2.2 Simülasyon	6
2.2.1 Simülasyonun Tanımı.....	6
2.2.2 Simülasyonun Tarihsel Gelişimi	7
2.2.3 Simülasyonun Genel Özellikleri.....	8
2.2.4 Simülasyonun Kullanım Amaçları	9
2.2.5 Simülasyonun Uygulama Alanları	11
2.2.6 Üretimde Simülasyon	12
2.2.7 Simülasyonun Avantaj ve Dezavantajları.....	14
2.2.7.1 Simülasyonun avantajları	14
2.2.7.2 Simülasyonun dezavantajları	16
2.3 Sistem ve Modelleme	17
2.3.1 Sistem	17
2.3.2 Sistemin Bileşenleri.....	18
2.3.2.1 Sürekli ve süreksiz (kesikli) sistem	19
2.3.2.2 Sistemin yapısal elemanları	20
2.3.2.3 Sistemin dinamik elemanları	22
2.3.3 Model.....	22
2.3.3.1 Model Türleri.....	23
2.3.4 Sistemlerin İncelenmesi ve Modelleme.....	24
2.3.5 Bir Simülasyon Modelindeki Değişkenler	26
2.3.5.1 Çıktı değişkenleri.....	27
2.3.5.2 Süreksiz Sistemlerde Simülasyon.....	28
2.3.5.3 Süreksiz sistemlerde ortam yaklaşımları	29
2.3.5.4 Sürekli (Continuous) Simülasyon Yöntemi.....	31
2.4 Simülasyon Çalışmasındaki Adımlar	31
2.4.1 Problemi Tanımlama ve Sistemi Anlama.....	33
2.4.2 Çalışmanın planlanması ve amaçların belirlenmesi	33
2.4.3 Modelin oluşturulması ve simülasyon dili seçimi	33
2.4.3.1 Model geliştirme.....	33
2.4.3.2 Veri toplanması	35
2.4.3.3 Simülasyon dili seçimi	35

2.4.4 Niteleme, Doğrulama ve Geçerlilik.....	36
2.4.5 Simülasyon Deneilerinin Dizaynı.....	38
2.4.6 Sonuçların Analizi ve Dokümantasyon.....	38
2.4.7 Çıktı Bilgisi Analizi.....	39
2.4.8 Uygulama (Implementation).....	39
2.5 Performans İstatistiklerinin Ölçümü.....	39
2.5.1 Sistem Zamanı.....	40
2.5.2 Kuyruk Zamanı.....	40
2.5.3 Herhangi Bir Zamanda Kuyruktaki Ortalama Varlık Sayısı.....	40
2.5.4 Kaynak Verimliliği.....	40
3. MATERYAL VE YÖNTEM.....	42
3.1 Simul8 Yazılımı.....	42
3.2 Program Arayüzü.....	42
3.3 Program Yapı Taşları.....	43
3.4 Program Prosedürleri.....	46
3.4.1 Model Kurulumu.....	46
3.4.2 Menü Şeridi Açıklamaları.....	46
3.4.2.1 Home Sekmesi.....	46
3.4.2.2 Data & Rules Sekmesi.....	47
3.4.2.3 Insert Sekmesi.....	48
3.4.2.4 View Sekmesi.....	48
3.4.2.5 Visual Logic Sekmesi.....	49
3.4.2.6 Advanced Sekmesi.....	49
3.4.2.7 File Sekmesi.....	50
3.4.3 Çalışma saati.....	50
3.4.4 İşlem ve Varış Süreleri.....	50
3.4.5 Kaynaklar.....	52
3.4.6 Sonuçlar.....	53
3.4.7 Grafikler.....	56
3.5 Simülasyon Modelleri.....	57
3.6 Model Kurulum ve İş Akış Diyagramlarının Belirlenmesi.....	62
3.7 Alternatif İş Akış Diyagramlarının Oluşturulması.....	66
4. ARAŞTIRMA BULGULARI.....	75
4.1 Operasyonlara Ait Dağılımların Belirlenmesi.....	75
4.2 Modelin Simul8 Yazılımına Aktarılması.....	79
4.3 Model Simul8 Yazılımına Aktarılırken Kabul Edilen Varsayımlar.....	80
4.3.1 Makine Arıza Yüzdeleri.....	81
4.3.2 Kalite Kontrolde Dönüş Yüzdeleri.....	82
4.3.3 Modeli en uygun besleme zamanının belirlenmesi.....	83
4.4 Model Geçerliliği.....	87
4.5 Simülasyon Yazılımında Geliştirilen Algoritma.....	91
5. TARTIŞMA VE SONUÇ.....	96
6. KAYNAKLAR.....	105
ÖZGEÇMİŞ.....	107

ÇİZELGE DİZİNİ

	<u>Sayfa</u>
Çizelge 1.1: Modele ait üretim süreçleri	2
Çizelge 2.1: Kesikli olay modellemedeki bazı kavramlar	29
Çizelge 3.1: Model 1 sağ omuz çatma operasyonu değerleri	66
Çizelge 4.1: Model 1 sağ omuz çatma operasyonu değerleri	75
Çizelge 4.2: Model 1'e ait operasyonların istatistikî dağılımı.....	76
Çizelge 4.3: Model 2'ye ait operasyonların istatistikî dağılımı.....	77
Çizelge 4.4: Model 3'e ait operasyonların istatistikî dağılımı.....	78
Çizelge 4.5: Makine sayıları.....	81
Çizelge 4.6: Makine türlerine göre arıza süreleri (dk).....	81
Çizelge 4.7: Örnek model besleme süreleri ve çıktı adedi ortalamaları	84
Çizelge 4.8: Model 1 Alternatif akışlar algoritma sonuçları	93
Çizelge 4.9: Model 2 alternatif akışlar algoritma sonuçları	94
Çizelge 4.10: Model 3 alternatif akışlar algoritma sonuçları	95
Çizelge 5.1: Algoritmalar ve Simul8 hat dengeleme verimlilikleri.....	96
Çizelge 5.2: Model kalp algoritması sonuçları.....	98
Çizelge 5.3: Model 1 alternatif akışlar kalp algoritması sonuçları.....	99
Çizelge 5.4: Model 2 alternatif akışlar kalp algoritması sonuçları.....	100
Çizelge 5.5: Model 3 alternatif akışlar kalp algoritması sonuçları.....	101
Çizelge 5.6: Model1 operasyonları için önerilen çalışan sayısı.....	102
Çizelge 5.7: Model2 operasyonları için önerilen çalışan sayısı.....	103
Çizelge 5.8: Model3 operasyonları için önerilen çalışan sayısı.....	104

ŞEKİL DİZİNİ

	<u>Sayfa</u>
Şekil 1.1: Kritik yol üzerinde yer alan süreçler	2
Şekil 2.1: Sistem kavramında girdi ve çıktı yaklaşımı	18
Şekil 2.2: Durum değişkeninin kesikli sistemde değişimi.....	19
Şekil 2.3: Durum değişkeninin sürekli sistemdeki değişimi	20
Şekil 2.4: Basit simülasyon modeli unsurları	20
Şekil 2.5: Modelin karmaşıklığı ile modelleme süresinin karşılaştırılması	23
Şekil 2.6: Sistem çalışma yöntemleri	24
Şekil 2.7: Modelleme işlemi.....	26
Şekil 2.8: Simülasyon modellerinde kullanılan değişkenler	27
Şekil 2.9: Olay programlama yaklaşımı	30
Şekil 2.10: Faaliyet arama yaklaşımı.....	30
Şekil 2.11: Simülasyon çalışmasındaki adımlar.....	32
Şekil 2.12: Simülasyon model gelişim işlemi	37
Şekil 3.1: Simul8 arayüzü.....	43
Şekil 3.2: Başlangıç noktası özellikleri penceresi	44
Şekil 3.3: Kuyruk özellikleri penceresi	44
Şekil 3.4: Aktivite özellikleri penceresi	45
Şekil 3.5: İş çıkış noktası özellikleri penceresi.....	45
Şekil 3.6: Kaynak özellikleri penceresi	46
Şekil 3.7: Örnek model.....	46
Şekil 3.8: Home sekmesi görünümü.....	47
Şekil 3.9: Data & Rules sekmesi görünümü.....	47
Şekil 3.10: Insert sekmesi görünümü	48
Şekil 3.11: View sekmesi görünümü.....	49
Şekil 3.12: Visual Logic sekmesi görünümü.....	49
Şekil 3.13: Advanced sekmesi görünümü	50
Şekil 3.14: Çalışma saati ayarı	50
Şekil 3.15: İşlem süresi ayarı.....	51
Şekil 3.16: Varış süresi ayarı.....	52

Şekil 3.17: Kaynak ayarları	53
Şekil 3.18: İş giriş sonuçları	53
Şekil 3.19: Depolama haznesi sonuçları	54
Şekil 3.20: Kuyruk grafiği örneği	54
Şekil 3.21: İş merkezi sonuçları	55
Şekil 3.22: İş merkezi sonuçları	55
Şekil 3.23: Kaynak sonuçları	56
Şekil 3.24: Sonuçlar özeti	56
Şekil 3.25: Grafik ayarları	57
Şekil 3.26: Basic T-shirt (Model 1)	57
Şekil 3.27: Elbise (Model 2)	58
Şekil 3.28: Tayt (Model 3)	58
Şekil 3.29: Model 1 (Basic t-shirt) üretim bandı	59
Şekil 3.30: Model 2 (Elbise) üretim bandı	60
Şekil 3.31: Model 3 (Tayt) üretim bandı	61
Şekil 3.32: Model 1 iş akış diyagramı	63
Şekil 3.33: Model 2 iş akış diyagramı	64
Şekil 3.34: Model 3 iş akış diyagramı	65
Şekil 3.35: M1A1 iş akış diyagramı	67
Şekil 3.36: M1A2 iş akış diyagramı	68
Şekil 3.37: M1A3 iş akış diyagramı	69
Şekil 3.38: M1A4 iş akış diyagramı	70
Şekil 3.39: M2A1 iş akış diyagramı	71
Şekil 3.40: M2A2 iş akış diyagramı	72
Şekil 3.41: M2A3 alternatif iş akış diyagramı	73
Şekil 3.42: M3A1 Alternatif iş akış diyagramı	74
Şekil 4.1: Stat-Fit modülü görüntüsü	79
Şekil 4.2: Örnek model	79
Şekil 4.3: SIMUL8 programında oluşturulan model arayüzü	80
Şekil 4.4: Aktivite özellikleri penceresi	82
Şekil 4.5: Aktivite özellikleri tamir oranları tanımlama	83
Şekil 4.6: Excel'de örnek model besleme kritik süresi hesaplama	85
Şekil 4.7: Giriş besleme zamanının Simul8 yazılımına aktarılması	85
Şekil 4.8: Giriş besleme zamanının sabitlenmesi	86

Şekil 4.9: Giriş besleme adedinin sabitlenmesi	86
Şekil 4.10: Hipotez testleri	87
Şekil 4.11: Kolmogrov-Smirnov (KS) testi	88
Şekil 4.12: Sistem 1 ve model 1 geçerlilik sonuçları	89
Şekil 4.13: Sistem 2 ve model 2 geçerlilik sonuçları	89
Şekil 4.14: Sistem 3 ve model 3 geçerlilik sonuçları	89
Şekil 4.15: Simülasyon yazılımında geliştirilen algoritma.....	91

ÖNSÖZ

Tez konumun belirlenmesinden sonuçların analizine dek çalışmam süresince görüş ve önerileriyle beni yönlendirerek desteğini esirgemeyen değerli danışman hocam Doç. Dr. Can ÜNAL' a,

Çalışmam sırasında veri toplama ve uygulama konusunda işletmelerinde imkân sağlayan Aster Tekstil Genel Müdürü İsmail KOÇALI 'ye,

Her daim yanımda olan ve manevi desteklerini hiçbir zaman esirgemeyen annem, babam ve kardeşime en içten saygılarım ile teşekkür ederim.

MART 2015

SELİN BİLGET

1. GİRİŞ

Günümüzün rekabetçi ve genişleyen tekstil pazarında, işletmelerin organizasyonel problemlerini aşarak, zamanında üretim yapmaları büyük önem taşımaktadır. Artan müşteri talepleri, sık değişen teknoloji ve gittikçe sertleşen global rekabet, firmaları ürettikleri ürünü veya verdikleri hizmeti, daha hızlı, daha etkili, daha kaliteli ve daha az maliyette nasıl üretebileceklerini bulmaya zorlamaktadır. Hatasız üretimin yanı sıra, siparişlerin teslim tarihlerine yetiştirilmeleri için işletmeler, güçlü matematiksel analizler yaparak mevcut sistemde yer alan darboğazları belirlemelidirler. Modaya yönelik, siparişe dayalı çalışan konfeksiyon işletmelerinde en büyük problem, organizasyon eksikliği ve darboğazların belirlenememesinde yatmaktadır. Siparişe dayalı çalışan konfeksiyon işletmeleri, genellikle teslim tarihlerine siparişlerin yetiştirilmesinde büyük sorunlar yaşamaktadırlar. Bu sorunun temel kaynağı, planlama ve organizasyonda yatmaktadır.

Bilindiği üzere, sistem üzerinde deneyler yapıp, sonuç almak riskli ve maliyetli bir iştir. Bu nedenle, sistemlerin belirli zaman periyodu içerisinde benzerlerinin yaratılması yoluyla, simülasyon modelleri elde edilmektedir. Bu sayede, elde edilen simülasyon modelleri üzerinde daha kolay bir şekilde denemeler yapılabilen ve sonuç almak çok daha basit olmaktadır. Elde edilen sonuçlar üzerinden değerlendirmeler yaparak, mevcut sistemde aksayan bölümler açık bir şekilde belirlenebilmekte ve bu darboğazlara alternatif sistemler sunulabilmektedir. Böylece, hem organizasyonel hem de kapasite problemlerini belirlemek kolaylaşmaktadır.

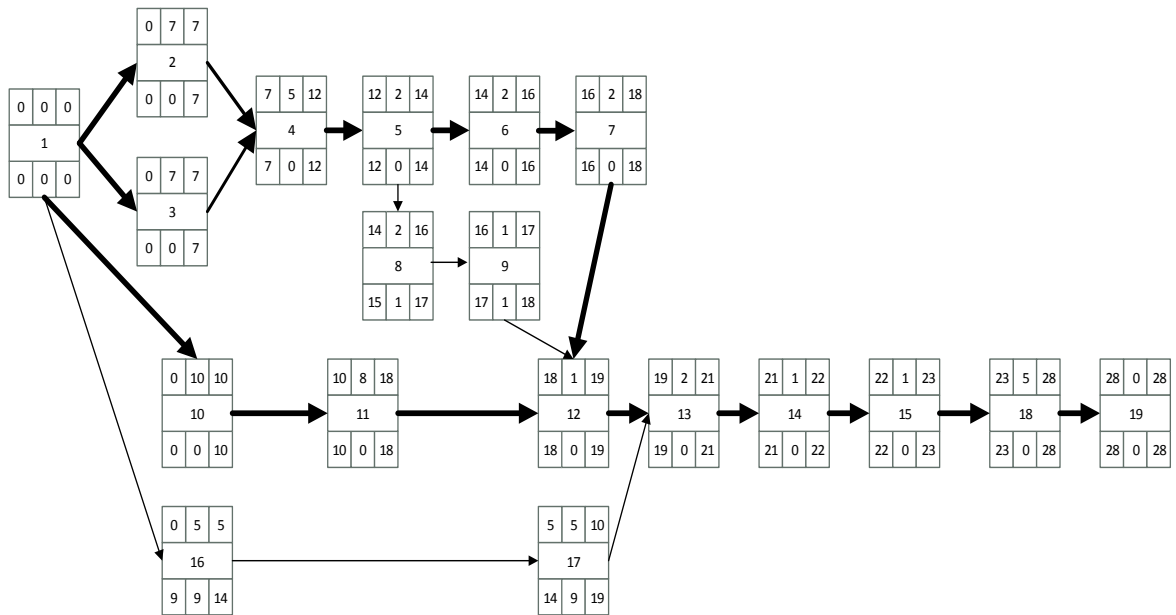
Konfeksiyon işletmelerinde, hammaddenin bitmiş ürüne dönmesi için farklı departmanların farklı süreçlerine ihtiyaç vardır. Simülasyon uygulamasını herhangi bir işletmenin tüm departmanlarında uygulamak mümkündür. Ancak çalışmanın uygulanması için gerçekçi bir zaman yönetimi gerektirdiğinden, dikimhane departmanı hedef olarak seçilmiştir. Bilindiği üzere dikimhane bir konfeksiyon işletmesinin kalbidir. Dikimhanede oluşan herhangi bir gecikmenin tüm işletmenin sipariş zamanını etkileyecek niteliktedir.

Çalışmanın gerçekleştirildiği işletmede, dikimhane departmanının önem düzeyini belirginleştirmek için “Kritik Yol Metodu - CPM” ile simülasyon uygulaması yapılması planlanan modeller için dikimhanenin kritik yol üzerinde yer alıp almadığı incelenmiştir. Çizelge 1.1’de verilen üretim aşamaları için erken başlangıç-erken bitiş, geç başlangıç-geç bitiş zamanları Şekil 1.1’de uygulanan CPM sonucunda belirlenmiş ve dikim işleminin kritik yol üzerinde yer almakta olduğu görülmüştür.

Çizelge 1.1 Modele ait üretim süreçleri

Operasyon Numarası	Operasyon Adı	Operasyon Numarası	Operasyon Adı
1	Sipariş Onayı	11	CS Onay
2	Ham Kumaş Geliş	12	Pilot Kesim
3	Lap Onay	13	Pilot Onay
4	İlk İmalat Kumaş Geliş	14	Kesim Başlangıç
5	İlk İmalat Kumaş Onay	15	Dikim Öncesi Nakış
6	Baskı Nakış Geliş	16	UPC
7	Baskı Nakış Onay	17	Dikim Paket Malzeme Geliş
8	Boyalı Aksesuar Geliş	18	Dikim
9	Boyalı Aksesuar Onay	19	Yükleme
10	Fit Onay		

Erken Başlangıç	Faaliyet Süresi	Erken Bitiş
Operasyon		
Geç Başlangıç	Boş Zaman	Geç Bitiş



Şekil 1.1 Kritik yol üzerinde yer alan süreçler

Dikim departmanında, herhangi bir ürün için üretim sürecinin kısaltılması, sipariş süresinin kısalması ve müşteriye daha hızlı geri dönüş yapılmasına sebep olacaktır. Bu nedenle firmalar yalın üretim felsefesini dikim sürecine entegre etmeye çalışmakta, bu amaç doğrultusunda çeşitli danışman firmalarla ortak projeler yürütmektedirler.

Yalın üretim, ürün ve hizmet yaratma sürecini israflardan arındırıp sadeleştirerek sunulan değeri mükemmelleştirmek ve bu yolla firma karlılığını arttırmak amacını taşıyan kavram, sistem ve teknikler bütünüdür. Yalın düşüncede israf, bilinen anlamının ötesinde müşteri açısından bir değer oluşturmayan, müşterinin fazladan bedel ödemeyi kabul etmeyeceği her şeydir. Tasarımdan sevkiyata tüm ürün/hizmet yaratma aşamalarındaki her türlü israfın (hatalar, aşırı üretim, stoklar, bekleme, gereksiz işler, gereksiz hareketler, gereksiz taşımalar) yok edilmesi ile maliyetlerin düşürülmesi, müşteri memnuniyetinin artırılması, piyasa koşullarına uyum esnekliğinin kazanılması, nakit akışının hızlandırılması hedeflenir (Womack 1997).

Yalın Düşünce'nin İlkeleri şu şekildedir;

- 1) Değer: Yalın düşüncenin başlangıç noktası “değer”dir. Değeri üretici yaratır ama değer ancak nihai müşteri tarafından tanımlanabilir. Değer tanımının anlamlı olabilmesi için müşterinin ihtiyaçlarını, belli bir zamanda ve belli bir fiyattan karşılayan belli bir ürün yada hizmet cinsinden ifade edilmesi gerekir.
- 2) Değer Akışı; Yalın Düşünce'nin ikinci adımı değer akışının tanımlanmasıdır. Değer akışı ham maddenin nihai ürüne dönüşme sürecindeki bir üreticiden diğer üreticiye ve son kullanıcıya kadar olan tüm aşamaları içerir
- 3) Sürekli Akış; Sürekli akış uygulandığında ürün geliştirme, sipariş alma, fiziksel üretim işleri çok kısa sürede tamamlanabilir hale gelecektir. Bu müşterinin gerçekten istediği şeyleri, tam istediği zamanda tasarlayabilme, planlayabilme ve üretebilme imkânını verdiği için tahmini yapmak, karmaşık planlama yazılımları kullanmak, stokta kalan ürünleri itmek için kampanyalar düzenlemek zorunluluklarını ortadan kaldırarak sadece istenen şeylerin daha iyi üretilmesine odaklanabilmeyi de sağlayacaktır (Womack 1997).
- 4) Çekme; Çekme ilkesi, nihai müşterinin belli bir ürün için yaptığı taleple başlar, ürün müşteriye ulaşana kadar geçen tüm aşamaları geriye doğru izleyip her aşamanın bir öncekinden talep etmesiyle üretimi başlatmak şeklinde uygulanır.
- 5) Mükemmellik; Yalın yaklaşım uygulandığında işgücü verimliliği, işin tamamlanma zamanı, stoklar, müşteriye ulaşan hatalı ürünler ile hurda oranları, ürünü pazara sunma süresi gibi parametrelerin hepsinde birden radikal iyileşmeler görülecek, çok küçük ilave maliyetlerle ürün çeşitliliği artırılabilir ve bunlar yeni teknoloji yatırımlarına gerek kalmadan, hatta mevcut bazı ekipmanlar satılarak negatif sermaye yatırımı ile ve birkaç yıllık bir süre içinde başarılabilecektir.

Çalışmanın amacı konfeksiyonda yalın üretim sistemlerini simülasyon uygulamasıyla incelemek olduğu için yalın üretim konusunda daha ayrıntılı bilgi verilmemiştir.

Çalışmada simülasyon, Simul8 yazılımının özellikleri, ana modeller ve alternatif modellerin oluşturulması, oluşturulan modellerin simul8 yazılımında geçerliliğinin sağlanması, geliştirilen algoritma ile kişi başı üretim adeti daha fazla olan alternatiflerin oluşturulması amaçlanmıştır.

2. KURAMSAL TEMELLER

2.1 Literatür Taraması

Çalışma başlangıcından bu zamana kadar, konuyla ilgili literatürdeki çalışmalar incelenmiştir. Ancak konfeksiyon üretimine yönelik simülasyon çalışmasının sayısı oldukça azdır. Literatür taramaları sırasında şu çalışmalara rastlanmıştır;

Cocks ve Harlock çalışmalarında herhangi bir konfeksiyon işletmesinin dikim bölümünün üretimini modellemek için Fortran 77’de bir simülasyon programı yazmıştır. Programda dikim bölümündeki işlemler, makineler ve yaptığı işler, çevrim zamanları, iş sıraları, sistemdeki iş miktarı tanımlanmıştır.

Fozzard ve arkadaşları, konfeksiyonda akış hatlarının benzetim modelini kurarken, operatörün boş kalma süresi, makine beklemeleri-bozulmaları, işlem süreleri gibi kriterlerin de göz önüne alınması gerektiğini, hat denetleyicisinin rolünün de modeli kurarken önemli olduğunu fakat bunun modele aktarılmasının çok karmaşık olduğundan bahsetmişlerdir.

Zielinski ve Czacherska, bir konfeksiyon işletmesinin dikim hattının optimizasyonunu sağlamak amacıyla boş zamanları en aza indirme kriterini ele almışlardır. Hattın verimliliğini maliyetler kısıdı altında Group Witness simülasyon programı kullanarak artırmaya çalışmışlardır.

Rajakumar ve arkadaşları, konfeksiyon işletmesinde az iş yükü olan işçilere yeni işler atayarak üretim hattını dengelemeye çalışmıştır. Bunun için C++’da bilgisayar simülasyon programı yazmış ve çizelgeleme stratejisi olarak rastgele, kısa işlem süresi önce ve uzun işlem süresi önce kombinasyonlarını kullanmışlardır.

Güner ve Ünal, bir konfeksiyon işletmesindeki t-shirt dikim hattını Arena simülasyon programı kullanarak simüle etmişlerdir. Yine Ünal ve arkadaşları, çalışmalarında konfeksiyon işletmelerindeki dikim bantlarını dengelemek için sezgisel bir algoritma önermişler, ve önerdikleri algoritmayı, bir pantolon dikim hattını U-tipi ve düz hat olarak Arena simülasyon programı ile simüle ederek sınamışlardır.

Kurşun ve Kalaoğlu çalışmalarında t-shirt, sweatshirt ve gömlek dikim hatlarını Enterprise Dynamics simülasyon programı kullanarak simüle etmiş ve farklı senaryolar ile hatları dengelemeye çalışarak şirketin yatırım kararları için öneriler sunmuşlardır. Yine Kurşun ve Kalaoğlu simülasyon metoduyla bant dengelemeyi hedeflemiş bunun için pantolon dikim bandında iş zaman etüdü yapmışlardır. Daha sonra veri dağılımına uygunluk için istatistiksel testler yaparak model geçerliliği gerçek sistem ile karşılaştırılarak sınanmıştır.

Kurşun, bir konfeksiyon işletmesinde gömlek dikim hattının darboğaz noktalarını belirleyip bu noktalara tezgah, işçi ekleme-çıkarma kararlarıyla hattın dengelenmesi ve alınacak yatırım kararlarına destek olacak önerileri sunmuştur.

Kalaoğlu ve Sarıcam, simülasyon paket programı kullanarak giyim endüstrisinde modüler sistem uygulamıştır. Modüler sistemde ürün süreci verimlilik, operatör ve makine etkinliği, mamül ve yarı mamül olarak 3 farklı prensip üzerine tasarlanmıştır.

Eryürük bir elbise modeli montaj hattı seçmiş ve Arena simülasyon programı ile modellemiştir. Çalışmanın amacı sabit bir çevrim süresinde optimum makine ve operatör kullanarak, maksimum hat verimliliğine ulaşmak üzere bir montaj hattı tasarlamaktır.

2.2 Simülasyon

2.2.1 Simülasyonun Tanımı

Simülasyon, teorik ya da gerçek fiziksel bir sisteme ait neden sonuç ilişkilerinin bir bilgisayar modeline yansıtılmasıyla, değişik koşullar altında gerçek sisteme ait davranışların bilgisayar modelinde izlenmesini sağlayan bir modelleme tekniğidir.

Simülasyon, gerçek bir süreç ya da sistemin belirli bir zaman periyodu içinde benzerinin yapılmasıdır (Banks 2000). Simülasyon ile sistemin geçmiş üretim bilgileri elde edilir ve bu geçmiş üretim bilgilerinin gözlemlerinden sonuç çıkarılarak, gerçek sistemin işlem özellikleri hakkında bilgi edinilir.

Simülasyon, sistemin davranışını analiz etme ve tanımlama amacı ile kullanılmaktadır. Simülasyon ile hem var olan, hem de henüz tasarım aşamasında olan sistemlerin modelleme imkânı bulunmaktadır (Örneğin; başlangıç ya da planlama aşamasındaki sistemlerin simülasyonu gibi).

Simülasyon tekniğiyle, modele sistemin davranışını anlamak ya da çeşitli yeni üretim politikalarını uygulamak amacıyla deneyler dâhil edilmektedir. Simülasyon hem modelin yapısını hem de problemin çözümü için modelin deneysel kullanımını içermektedir. Bu nedenle, simülasyon modelleme, deneysel ve uygulamalı bir analiz yöntemi olup, aşağıdaki işlemleri de yerine getirmektedir;

- Sistemlerin davranışını anlamak,
- İncelenen davranış için geliştirilen teori ya da hipotez kurmak,
- Modeli kullanarak, gelecek hakkında bilgi edinmek, sistemde meydana gelen değişimlerin etkisini incelemek (Pegden ve Shannon 1995).

Bir sistemin simülasyonu, bu sistemi temsil edebilecek bir model oluşturma işlemidir. Simülasyon, gerçek sistemin modelinin tasarlanması ve bu model ile sistemin işletilmesi amacıyla yönelik olarak, sistemin davranışını anlayabilmek veya değişik stratejileri değerlendirebilmek için deneyler yürütülmesi sürecidir.

Simülasyon, geliştirilen veya yeniden düzenlenen sürecin işlemlerini tamamlamada ve deneme çalışmalarını yürütmeye ve süreçlerin hata zamanlarını tahmin etmek için yapılan deneysel çalışmadır. Yeni sürecin değişikliklere gösterdiği olası reaksiyonları da anlaşılabilir.

Simülasyon örnekleme ve analizi, bir fiziksel sistemin bilgisayar matematiği ile yaratılması ve denenmesi metodudur (Chung 2004).

Simülasyon, gerçek bir sistemin modelini tasarlama süreci ve sistemin davranışını anlamak veya değişik stratejileri değerlendirmek amacı ile geliştirilen bu model üzerinde denemeler yapmaktır (Banks 2000).

Bir başka tanıma göre simülasyon, gerçek bir prosesin veya sistemin zamana bağlı olarak modelini tanımlayan matematiksel bir modeldir. Simülasyon ister elle, isterse bilgisayar ile yapılsın, bir sistemin yapay kayıtlarının oluşturulması ve gerçek sistemin işletim karakteristikleriyle ilgili sonuçlarının elde edilmesinde bu yapay kaydın incelenmesini kapsamaktadır (Banks ve Carson 1984).

Simülasyon, gerçek hayattaki olayların bilgisayar ortamına aktarılması işlemidir. Sanal ortamlar sağlayan yazılımlardır. Bir sistemin simülasyonu, bu sistemi temsil edebilecek bir model oluşturma işlemidir.

Simülasyonlar, genel tasarım formları içinde metin, test, canlandırma, seslendirme, alıştırmaya uygulama gibi pek çok tasarım seçeneğinin uygulanmasına olanak tanımaktadır. Yaparak, yaşayarak öğrenmeyi sağlamaktadır.

Eğitimsel simülasyon, bir olay veya aktivitenin etkileşim sonucu öğrenilmesini sağlayan modellemedir.

Simülasyon, önerilen veya gerçek dinamik bir sistemin modellenmesi ve zaman içindeki davranışın gözlenmesi işlemidir. Bir simülasyon çalışması, herhangi bir sistemin davranışının incelenmesi ve farklı parametrelerin çalışma durumuna etkilerinin araştırılması amacı ile yapılır.

Simülasyon çalışmalarında uygulanan iki adım; model tasarımı ve deneylerdir. Model tasarımı sistemin tüm önemli durumlarını temsil eden bir modelin kurulmasıdır. Geçerli bir model kurulduktan sonra deneyler kısmı başlar. Simülasyon genellikle mevcut olmayan veya pahalı ve zor gerçekleştirilebilecek sistemlerin denenmesine imkân sağlar.

2.2.2 Simülasyonun Tarihsel Gelişimi

Simülasyon tarihi “WEICH” şeklinde adlandırılan Çin Savaş Oyunlarından, 5000 yıl öncesinden gelir ve 1780'lere kadar devam eder. Prussian'ların bu oyunları ordularındaki trenlerde kullanmalarından beri, tüm askeri güçlerin başkanları, simüle edilmiş çevre koşulları altında askeri stratejileri test etmek için savaş oyunlarını kullanmışlardır.

2. Dünya Savaşı esnasında matematikçi Jhon Van Neumann tarafından bu teknik askeriyeden ve operasyonel oyunlardan yeni bir teknik olan “Monte Carlo” simülasyon tekniği geliştirilmiştir. Bir nicelik miktar tekniği olarak Los Alamos Scientific Laboratuvarında nötronlarla çalışılırken, Van Neumann simülasyonu, elle veya fiziksel modellerle analizi karmaşık ve pahalı olan fizik problemlerini çözmede kullanıldı. Nötronların rastsal yapısı ihtimallerle uğraşmada rulet tekeri kullanımını önerdi. Oyun yapısı nedeni ile Van Neumann kanunların değişimi çalışmasını “Monte Carlo” modeli olarak adlandırdı. 1950'lerde iş bilgisayarlarının gelişi ve birleşik kullanımı ile simülasyon bir yönetim aracı olarak geliştirdi (Chung 2004).

Simülasyon, 1950 ve 1960'lı yılların sonlarına doğru, genellikle sadece büyük sermaye yatırımları gerektiren şirketlerin kullandığı çok pahalı ve özel alanlarda kullanılan bir araç idi. Bu şirketler, Fortran gibi programlama dilleri ile büyük ve karmaşık simülasyon modellerini geliştirmek için doktorasını yapmış kişilerden oluşan çalışma gruplar kurmuşlardı. Geliştirilen modeller daha sonra büyük merkezi işlem birimlerinde çalıştırılıyorlardı. Bu makinelerin maliyeti, saatte bin dolara kadar çıkabiliyordu. Günümüzde ise, herkesin sahip olabileceği bir kişisel bilgisayar bile, bu makinelerden çok daha güçlü ve çok daha hızlıdır.

Simülasyonun bugün bildiğimiz kullanımı ise 1970'li yılların sonunda başlamıştır. Bilgisayarların maliyeti oldukça düşmüş ve çok daha hızlanmış ve simülasyonun değeri birçok alanda keşfedilmeye başlanmıştır. Aynı zamanda, bu süreç içerisinde simülasyon, üniversitelerde endüstri mühendisliği ve yöneylem araştırması dallarının standartlaşan bir bölümü haline gelmiştir. Simülasyonun endüstri alanındaki hızlı ilerleyişi, üniversiteleri, simülasyonu daha kapsamlı bir şekilde öğretmeye zorlamıştır.

Aynı zamanda gelişen taleple beraber bu konuda çalışan araştırmacı ve öğrencilerin sayısı da oldukça artmıştır. Son zamanlarda modern yönetim biliminde de önemli bir araç olarak simülasyonun kullanıldığı gözlenmektedir.

1980'li yılların sonunda kişisel bilgisayarların kapasitelerinin de artmasıyla simülasyon kullanımı iş dünyasına iyice yerleşmiştir. Simülasyon günümüzde halen, başarısız olmuş veya geliştirilmeye ihtiyaç duyulan sistemlerin analizinde kullanılsa da, birçok kurum artık planlanan sistem kurulmadan önce simülasyonunun yapılmasını koşul olarak getirmektedir. Aslında birçok durumda, simülasyona başvurulduğu anda, sistemin tasarımını değiştirmek için artık çok geçtir ancak yine de geriye kalan işlemlerde sistemi yönlendirebilmek için sistem yöneticisine ve sistem tasarımcısına bir şans tanınabilmektedir. 1980'li yılların sonlarına doğru birçok büyük firma simülasyonun değerini anlamıştır ve bunlardan birçoğu, büyük sermaye gerektiren yatırımları onaylamadan önce bir gereksinim olarak simülasyonlarını yapmıştır. Yine de bu yıllarda simülasyon çok fazla yaygın değildi ve çok az sayıda ciddi firmalar kullanıyordu.

1990'ların birçok kuruluş simülasyon araçlarını benimsemiş ve simülasyonu, en etkili olabileceği, projelerinin çok erken aşamalarında kullanmaya başlamışlardı. Çok iyi bir animasyon yeteneği, kullanım kolaylığı, bilgisayarların kapasitelerindeki gelişme, diğer paket programlarla kolay uyumu ve simülatörlerin gelişmesi, simülasyonu birçok firma için standart bir araç haline getirmiştir. Simülasyonunun uygulama şekli değişebilmektedir; sistemlerin tasarım aşamasında kullanılan simülasyon programları, yapılan herhangi bir değişiklikte sistemin farklı alanlarında kullanılabilir, böylelikle yaşayan bir simülasyon kullanımı sağlanabilmektedir (Chung 2004).

2.2.3 Simülasyonun Genel Özellikleri

Stok kontrol ve kaynak sistemleri modellemesi gibi simülasyon tekniği bakımından kesikli konum simülasyonu kapsamında incelenen modelleme çalışmalarının, bilgisayarda programlanmasında dikkati çeken temel özellikler Law ve Kelton (2000) tarafından belirtilmektedir;

- 0 ile 1 arasında uniform $U(0,1)$ dağılışından şans sayısı türetimi,
- Bilinen bir olasılık dağılışından şans değerlerinin türetimi,
- Simülasyon saatinin çalıştırılması,
- Uygun simülasyon bloklarına geçişte kontrol sisteminin kurulması,
- Simülasyon listesine kayıt ekleme, kayıt çıkarma olanakları,
- Uygun veri analiz yöntemlerinin kullanımı,
- Sonuçların yazdırılması,
- Hataların izlenmesi.

Bu özellikler, simülasyonda özel amaçlı simülasyon dillerinin kullanımını zorlamaktadır. Bu diller daha sonra simülasyon tekniklerinin kullanım alanının genişlemesine yol açmıştır; ancak buna rağmen özel amaçlı simülasyon dilleri ile genel amaçlı programlama dillerinin arasında simülasyon senaryolarının bilgisayarda programlanması açısından uzun zamandan beri avantaj ve dezavantaj tartışmaları süregelmektedir.

Simülasyonu kapsamında incelenen modelleme çalışmalarıda belirtilen özellikler ile birlikte simülasyon kullanılırken şunlara da dikkat edilmelidir:

Bir sistemi durdurarak çalıştıramazsınız: Eğer sistem çözümlenmek için uzun süre durdurulursa, bu sistemin doğal seyrini değiştirebilir. Bu nedenle uygun çalışma hareket halinde iken gerçekleştirilmelidir. Bilgisayar simülasyonları sistemin hareket halinde olmasına müsaade eder. Performansla ilgili verim ölçümleri sistem operasyonları devam ederken elde edilir. Bu, özellik bilgisayar simülasyonlarının mevcut sistem devam ederken çalışabilme yeteneklerini işaret eder.

Doğru sonuç için tekrarlamalar gereklidir: İkinci özellik, sistemlerin tek bir simülasyon yürütülmesi ile sonuçlandırılmayacağını vurgulamaktadır. Nitekim analiz için simülasyonun kullanıldığı pek çok sistem için tek bir simülasyon yürütülmesi ile elde edilecek olan sonuçlar, mevcut problem için ciddi çareler bulmada yeterli değildir.

Görsel niteliklerinden dolayı kullanıcı adayları tarafından bir oyun olarak algılanması önlenmelidir: Animasyonla örneklemenin tehlikesi, eğitimsiz ve bilgisiz bireylerin simülasyonun karmaşık yapısını takdir edemeyecek olmasıdır, bu bireyler simülasyonun sadece bu animasyonlarını düşünürler. Bu nedenle böyle düşünen bireyleri, modelin önemli çareler ve politik işlem kararları almada kullanılır olduğuna inandırmak güçleşir. Yani model, sadece matematiksel olmamalı aynı zamanda bilgisi olmayanları da etkileyip inandırmalıdır (Chung 2004).

2.2.4 Simülasyonun Kullanım Amaçları

Özel amaçlı simülasyon dilleri, düşük operasyon maliyetleri için yüksek hesaplama kabiliyetleri ve simülasyon metodolojisindeki gelişmeler, simülasyonu yöneylem araştırmasında ve sistem analizinde en çok kullanılan ve kabul edilen bir metod yapmıştır. Simülasyonun hangi şartlar altında kullanılması gerektiği birçok yazar tarafından

incelenmiştir. Bunları genel olarak sınıflandırırsak, simülasyon aşağıdaki amaçlar için kullanılabilir (Banks ve Carson 1984);

1. Simülasyon, karmaşık bir sistemin içyapısını veya karmaşık bir sistemdeki alt sistemi incelemek için kullanılabilir.
2. Bilgi, organizasyonel ve çevresel değişiklikler simüle edilebilir ve modelin davranışı üzerinde bu değişikliklerin etkileri incelenebilir.
3. Bir simülasyon modelinin tasarımından elde edilen bilgiler, incelenen sistemin geliştirilmesine büyük ölçüde katkıda bulunmaktadır.
4. Simülasyon girdilerini değiştirerek ve sonuçları inceleyerek, hangi değişkenlerin daha önemli olduğu ve değişkenlerin birbirlerini nasıl etkiledikleri hakkında bilgi edinilebilir.
5. Simülasyon, analitik çözüm metodolojisini destekleyen bir bilgi verici araç olarak kullanılabilir.
6. Simülasyon, analitik sonuçları test etmek için kullanılmaktadır.
7. Güncel sistemi değiştirmeden bilgi artışı sağlamak için kullanılmaktadır: Simülasyon modelleri, hareket ettiremeyen sistemlerin denenmesi için olası tek serbest metottür. Bazı sistemler oldukça hassas veya kritik olabilirler. Bu yüzden bu sistemleri analiz etmek için operasyon tipini veya strateji kaynağını değiştirmek imkânsız olabilir. Bu tip bir sisteme klasik bir örnek olarak ticari bir hava alanındaki güvenlik kontrol noktaları verilebilir. Stratejik operasyonların yürütülmesi veya deneme düzeyleri kaynakları, operasyonel kabiliyet veya sistemin güvenlik etkinliği üzerinde ciddi bir etkiye sahiptir.
8. Simülasyon modeli seçilen donanımın yapılanışını yerine getirmeden önce hataların ayıklanmasına yardımcı olmaktadır.
9. Simülasyon modeli amaçlanan sistemin ne kadar iyi performans göstereceği hakkında bir fikir verebilir. Yeni bir sistem modelinin maliyeti, herhangi bir kayda değer imalat prosesinin kurulması için gerekli olan ana yatırımın yanında çok küçük kalabilir; farklı düzeylerde ve farklı fiyatlardaki donanımların etkileri simülasyon kullanılarak değerlendirilmektedir (Chung 2004).
10. Simülasyon, sistemin davranışını analiz etme ve tanımlama amacı ile kullanılmaktadır. Simülasyon ile hem var olan, hem de henüz tasarım aşamasında olan sistemlerin modelleme imkânı bulunmaktadır (Örneğin; başlangıç ya da planlama aşamasındaki sistemlerin simülasyonu gibi) (Gürkan 2004).

Simülasyon temel olarak, sistemin operasyonel yönlerini ortaya çıkarmayı hedeflemektedir, yani “ne?”, “ne zaman?”, “nerde?” ve “nasıl?” sorgulamalarının yapıldığı görev alanları üzerinde yoğunlaşmaktadır.

Bilgisayar simülasyonu, sistem dizaynı ve analizinde hızla popüler olan bir araçtır. Simülasyon, mühendis ve planlamacılara sistemin dizaynı ve işletimiyle ilgili zamanında ve zekice kararlar vermeleri için yardımcı olur. Simülasyon tek başına problemleri çözemez fakat problemi açıkça tanımlar ve sayısal olarak alternatif çözümleri değerlendirir. Koşul (what-if) analizi yapabilen bir araç olan simülasyon önerilen herhangi bir çözüm için sayısal ölçüm ve analiz yapabilir ve kısa zamanda en iyi alternatif çözümü bulmaya yardımcı olur. Yeni bir sistemi kurmadan veya işletme politikalarını test etmeden önce bilgisayarda sistemi modelleyerek, sistem ilk çalıştırıldığında karşılaşılabilecek birçok tuzağı önceden görmemize yardımcı olur.

Devreye alma aşamasında iyi ürün elde etmek için aylar belki de yıllar süren çalışmalar simülasyonla günlere hatta saatlere sıkıştırılmış olur.

Simülasyonun çok fazla sayıda ve çok fazla özellikli değişkeni tek bir modelde toplayabilme özelliği, bugünkü kompleks sistemlerin dizaynı için vazgeçilmez bir araç olmasını sağlamaktadır. Bir üretim sisteminde, iş parçalarının, aletlerin, paletlerin, taşıma araçlarının, taşıma yollarının, işlemlerin vs. mümkün olan kombinasyonları, permütasyonları ve bunların sonucundaki performans değerlendirmeleri neredeyse sonsuzdur. Pratik sistemleri dizayn etmek için bilgisayar sistemi bir gereklilik olmuştur.

Servis sistemleri için müşteri akışını planlama, personel yönetimi, kaynak yönetimi ve bilgi akışının simülasyonunu yapmak da üretim sistemleri kadar önemlidir.

2.2.5 Simülasyonun Uygulama Alanları

Simülasyon, en iyi analiz araçlarından biri olması ve çok yönlülüğü nedeniyle, birçok araştırma çalışmasında işlem araştırma tekniklerinin kullanım oranlarının tespitinde kullanılmaktadır (Shannon 1981). Simülasyon bugün hemen hemen her sektörde uygulanabilir konuma gelmiştir. Aşağıda simülasyonun uygulama alanları kısa bir liste halinde verilmiştir:

Bilgisayar Sistemleri: Donanım bileşenleri, yazılım sistemleri, donanım ağları, bilgi tabanlı yapı ve yönetim, bilgi işlem, donanım ve yazılım emniyeti.

Üretim: Materyal kavrama sistemleri, otomatik üretim kolaylıkları, otomatik stoklama kolaylıkları, kayıt-kontrol sistemleri, güvenlik ve bakım çalışmaları, fabrika düzeni, makine dizaynı.

İş: Stok ve alım-satım analizleri, fiyat politikası, pazarlama stratejileri, kazanç çalışmaları, çek-akış analizleri, ulaşım alternatifleri, insan gücü planlanması, raporlama.

Hükümet: Askeri silahlar ve kullanımları, askeri taktikler, nüfus raporları, arazi kullanımı, sağlık-bakım dağıtımı, yangın koruma, polis servisleri.

Ekoloji ve Çevre: Su kirliliği ve temizlenmesi, atık kontrolü, hava durumu, deprem ve fırtına analizleri, maden araştırması ve çıkarılması, güneş-enerji sistemleri, ürün üretimleri.

Toplum ve Davranış: Yiyecek/nüfus analizleri, organizasyon yapıları, sosyal sistem analizleri, refah (bolluk) sistemleri, üniversite yönetimleri.

Bio Bilimler: Spor performans analizleri, hastalık kontrolleri, biyolojik yaşam döngüleri, tıbbi çalışmalar.

Hillier ve Lieberman, bu tekniğin geniş uygulama alanlarını belirtmek için aşağıdaki örnekleri vermişlerdir:

1. İşletme politikaları ve uygulamalarındaki (bakım kapasitesi, tesislerin, yedek uçakların vb.) değişiklikleri test etmek için bir havayolu şirketi tarafından büyük bir havaalanındaki operasyonların simülasyonu,
2. En iyi trafik akışını belirlemek için, trafik ışıklarının simülasyonu,
3. Optimal tamir personeli sayısını belirlemek için bakım operasyonu simülasyonu,
4. Bir radyasyon kalkanına yansıyan radyasyonun yoğunluğunu belirlemek için, bakım operasyonu simülasyonu,
5. Bir radyasyon kalkanına yansıyan radyasyonun yoğunluğunu belirlemek için, kalkandaki yüksüz parçacıkların akış simülasyonu,
6. Uygulama, kapasite ve tesislerin şekillerindeki değişiklikleri değerlendirmek için, çelik üretim operasyonunun simülasyonu,
7. Ekonomik politika kararlarının etkilerini tahmin etmek için ekonomi simülasyonu,
8. Savunma ve saldırı silah sistemlerini değerlendirmek için büyük çaplı askeri savaşların simülasyonu,
9. Büyük çaplı dağıtım ve envanter kontrol sistemlerinin tasarımını geliştirmek için bu sistemlerin simülasyonu,
10. Firmanın politikaları ve operasyonlarındaki değişiklikleri değerlendirmek için tüm firmanın genel operasyonlarının simülasyonu,
11. En ekonomik düzeyde, tatmin edici servis sağlamak için, gerekli parça kapasitesini belirlemek amacıyla bir telefon iletişim sisteminin simülasyonu,
12. En ideal baraj, elektrik santrali ve sulama işlerinin şeklini belirlemek için, ırmak havza operasyonlarının simülasyonu (Bank ve Carson 1984).

2.2.6 Üretimde Simülasyon

Modern dünyadaki rekabet nedeniyle, sanayiciler üretim politikalarını ve prosedürlerini tekrar gözden geçirerek, pahalı olan fabrika otomasyonuna geçmek zorunda kalmıştır. Yüksek otomasyona sahip bilgisayar kontrollü sistemlerin en dikkatli şekilde planının yapılması bile bazen analizeciyi büyük hata yapmaktan alıkoyamamaktadır. Bu yapılan hatalara makine kapasitelerinin yanlış eşleştirilmesi, beklemedeki parçalar için yetersiz alanın ayrılması, çok

kalabalık makine gidiş gelişlerinde otomatik araçların yığılmaları örnek olarak verilebilir. Karmaşık sistemlere sahip üretim sistemlerinin geleneksel tasarım ve analiz metotlarıyla çalışıldığında yeterli sonuç alınmadığı ispatlanmıştır. Organizasyonlar dinamik analizler için uygulamaya geçmeden önce simülasyonu bir analiz aracı olarak kullanmaktadırlar. Aksi takdirde, daha çok desteğe ihtiyaç duyulmakta ve maliyetler artmaktadır.

Simülasyon, sistem bileşenlerinin etkileşimlerini ve hareketlerini hesaplayarak karmaşık sistemlerin davranışını belirler. Makineler ve iş istasyonları arasındaki parça akışının değerlendirilmesiyle ve sınırlı kaynaklar için taleplerin uyuşması incelenerek, fiziksel düzen, ekipman seçimi ve işletme prosedürü değerlendirilir. Simülasyon sayesinde henüz var olmayan sistemlerin modelleri üzerinde deney yapma imkânı doğmaktadır. Bunun yanı sıra gerçek sisteme herhangi bir uygulama yapılmadan, yeni tasarımların ve değişimlerin sistemde ne gibi değişikliklere yol açacağı incelenebilir.

Otomasyona sahip üretim sistemlerinin simüle edilmesinde, kaynakların hangisinin performansı etkilediği ile ilgilenilmektedir. Sistemlerin modellenmesinde birçok problem ile karşılaşmaktadır. Bunlar, performansı en çok etkileyen kaynakların belirlenmesi (ve özellikleri), bu kaynakları ve ilişkilerini ifade eden bir model ya da tanımın formüle edilmesi, verilen senaryolarda ilgilenilen performans ölçüm değerlerinin saptanmasıdır.

Üretim sistemlerinin planlanması, tasarımı, yerleşimi ve işletim kararları aşağıda yer alan maddelere göre belirlenir:

- Zor-sistem konfigürasyon kararları, örneğin, modelde yer alacak ekipman tipleri ve numaraların seçimi ile sistemin yeteneğinin ayarlanması, fiziksel plan konfigürasyonu ve sistemde işlenecek bölümlerin ayarlanması,
- Kolay-sistem konfigürasyon kararları; örneğin, planlama, çizelgeleme, belirli zaman periyodu için bölümlerin, araçların ve işçilerin art arda düzenlenmesi,
- Gerçek-zaman kontrolü; örneğin, sistemdeki iş akışının kontrolü ve rastlantıların yanıtlanması (araç hataları, ekipman bozulmaları v.b.)

Üretim simülasyonu ile birçok alanı kapsayan çalışmalar yapılabilir. Genel uygulama alanlarına bakılırsa:

- Kapasite analizi ve planlanması,
- Ekipman ve personel planlanması,
- Kaynak ihtiyaç analizi ve planlanması,
- Darboğaz ve kısıt analizi,
- Üretim planlama,
- Çizelge optimizasyonu,
- Envanter yönetimi,

- Lojistik planlama,
- Yerleşim optimizasyonu,
- Bakım ve koruma düzenlemesi,
- Detaylı ve karmaşık kaynak modellemesi,
- Teslimat performansı analizi,
- Mühendis ve teknisyen işbaşı ve süreç eğitimi,
- Yeni operatör eğitimi,

2.2.7 Simülasyonun Avantaj ve Dezavantajları

2.2.7.1 Simülasyonun avantajları

Simülasyonun sistemlerin analizinde analitik ya da matematiksel modellere göre çok fazla avantajları bulunmaktadır. İlk olarak, simülasyonun analitik yöntemlere göre içeriğinin daha kolay kavramasından dolayı müşterileri ve yöneticileri memnun etmesi daha kolaydır. Dahası, simülasyon modelleri gerçek sistemler ile karşılaştırıldıklarından daha güvenilirler ve incelenen sistemin doğru özelliklerinin fazlası modele yansıtıldığından varsayımlar daha azdır (Hiller ve Lieberman 1980). Diğer avantajlar şunları içermektedir (Pegden, Shannon and Sadowski 1995).

- Modelin bir bilgisayarda simüle edilmesinden dolayı deneme simülasyonunun işletilmesi sıkıştırılmış zamanlarda yapılabilmektedir. Simülasyonun başlıca avantajı budur çünkü bazı süreçleri tamamlamak aylar hatta yıllar alabilir.

Uzun süreli süreç sistemleri çok sağlıklı analizleri zorlaştırabilir ve hatta işlenmesini imkânsızlaştırabilir. Bir bilgisayar modeli ile operasyon ve zaman alıcı süreçlerin etkileşimi ikinci olarak simüle edilebilir. Her simülasyon yürütmesinin çok yönlü eşlemesi, aynı zamanda analizlerin istatistiksel güvenilirliğinin de kolayca artırılabilmesi anlamına gelir. Böylelikle, önceden çok sağlıklı analiz edilmesi imkânsız olan sistemlerle artık çalışılabilir (Chung 2004).

- Yeni tasarımlar ve planlar herhangi bir kaynağın bu uygulamaya tahsis edilmesine gerek duyulmaksızın test edilebilmektedir.
- Personel için yeni politikalar, çalışma yöntemleri, kararların alınması, örgüt yapıları, bilgi akışı vb. gibi işlemler devam eden işlemlerde herhangi bir rahatsızlık vermeksizin simülasyon ile araştırılabilmektedir.
- Simülasyon, materyalde, bilgide, ürün akışında ve artan akış oranları için deneme seçeneklerindeki darboğazların tanımlanmasında kullanılmaktadır.
- Simülasyon, sistem içindeki kesin fenomenlerin nasıl ve neden oluştuğu hakkındaki varsayımların test edilmesinde kullanılmaktadır.

- Simülasyon ile zamanın kontrolü söz konusudur. Bu sayede, birkaç saniye içerisinde birçok ay ya da yıl için sistemin çalıştırılması gerçekleştirilebilmektedir. Piyasada birkaç dönemi yaşayarak elde edilecek bilgiler simülasyon modelleri ile birkaç saat veya günde öğrenilebilmektedir. Dolayısıyla küçük pazar testlerini beklemeksizin uygulama başlatılabilmektedir.
- Simülasyon ile bir modelin gerçekte nasıl çalıştığı ve performansı en çok etkileyen değişkenlerin ne olduğu kolaylıkla anlaşılabilir.
- Simülasyon en önemli avantajlarından biri de yeni ve alışılmadık durumların denenmesidir.
- Simülasyon metodu ile mevcut olan teorik ya da fiziksel bir sistemin bilgisayar ortamında modellendikten sonra farklı koşullar altında vereceği sonuçları gerçek sistemle karşılaştırma, alternatif senaryolar geliştirerek üretimde mükemmelliği yakalayabilme imkânı tanımaktadır.
- Stokastik elemanlar içeren, çok karmaşık yapıya sahip gerçek sistemler, analitik matematiksel model aracılığıyla tanımlanmasında güçlüklerle karşılaşıldığında, simülasyon tekniği uygulanabilmektedir.
- Simülasyon, sistem verilerinin detaylı olmadığı durumlarda kullanılabilir.
- Simülasyon modeli üzerinde daha sonra yapılacak analiz için veri, çoğu kez gerçek hayatta olduğundan daha ucuz elde edilmektedir.
- Simülasyon, analistleri daha genel düşünmeye zorlamaktadır.
- Simülasyon, analitik çözümlerin doğruluğunu gerçeklemek üzere kullanılabilir (Kuş 2003).
- Belirlenen bir stratejinin rakiplerin olası uygulamaları karşısındaki tutarlılığının test edilmesi için de simülasyon kullanılabilir. Böylelikle risk profili daha düşük stratejilerin geliştirilmesi sağlanabilmektedir.
- Aktif bir öğrenme yolu olan vaka çalışmaları ise hem hatırlama sürelerini uzatmakta, hem de hatırlama oranını %30'lara çıkarmaktadır. Ancak, yaşam süremiz içinde en kolay öğrendiğimiz çocukluk döneminden ders almamız gerekirse, en etkin öğrenme oyun yoluyla gerçekleşmektedir. İşte bu nedenle, yönetici eğitim programlarında oyun niteliği taşıyan simülasyon modelleri ilgi odağı olmaktadır.
- En etkin öğrenme hatalardan ders alarak gerçekleşmektedir; ancak stratejik hata maliyeti şirketleri batmaya sürükleyebilecek kadar yüksek olabilmektedir. Simülasyon

modellerinin bir başka faydası da bu maliyetlere katlanmaksızın üst yönetimin değişik alternatifleri değerlendirmesine fırsat tanınmasıdır.

- Simülasyon, yöneticilerin verdikleri kararların şirketin iş sonuçları ve değeri üzerindeki etkilerini daha net bir şekilde algılamalarına da yardımcı olmaktadır. Model bir kez kurulduktan sonra şirketin değişik bölümlerinde çalışanların da kendi konularıyla ilgili verdikleri kararların işin diğer bölümlerine ve sonuçlara nasıl yansıdığı konusunda eğitilmeleri kolaylaşmaktadır. Böylelikle şirkette karar alma sürecinin kalitesi iyileştirilmektedir.
- Ancak belki de simülasyonun strateji konusunda sağladığı en önemli avantajı, simülasyon modelleri ile yapılan çalışmaların yöneticileri rakiplerin bakış açısı ile bakmaya zorlamasıdır. Rakiplerini daha iyi tanıyan yöneticilerin onların stratejik hareketlerini daha hızlı ve daha iyi değerlendirme fırsatı olmaktadır (Argüden 2006).
- Simülasyon özel yazılım paketlerinin gelişimi, çalışanların normalde kullandığı birçok karmaşık hesaplama ve bilgisayar programları gereksiniminin azaltılmasına (izole edilmesine) yardımcı olmaktadır. Bu analitik gereksinimlerin azaltılması, daha fazla çalışana geniş varyasyonlu çevre ile daha farklı sistemleri analiz etme olanağı sağlamaktadır,
- Simülasyonun sunumu sırasında kullanılan animasyon, modelin güvenilirliğini/inanılabilirliğini sağlanmasına yardımcı olmaktadır. Ayrıca animasyon, sistem süreçleri arasındaki etkileşimlerin ve operasyonların eş zamanlı olarak tanımlanmasında kullanılabilir. Bu içerik sistem modelinin farklı durumları nasıl ele aldığı dinamik bir kanıtıdır (Chung 2004).

2.2.7.2 Simülasyonun dezavantajları

Simülasyonun çok sayıda avantajının olmasına rağmen, bunun yanı sıra dezavantajları da bulunmaktadır (Pegden, Shannon, Sadowski 1995).

- Simülasyon modelleme özel bir eğitim gerektiren bir işlemdir. Bu nedenle pratisyenlerin beceri seviyeleri geniş bir oranda değişiklik göstermektedir. Çalışmanın yararı modelin kalitesine ve modelistin becerisine bağlıdır.
- Simülasyon modelleri girdi-çıkı modelleridir. Örneğin; verilen bir girdi sisteminde sistemin muhtemel çıktıları oluştururlar. Bu nedenle çözümden çok modelin çalıştırılmasına yöneliktir. Optimum bir çözüm oluşturmamakla birlikte uzman tarafından özel durumlarda bir sistemin davranışının analizi için kullanılırlar.

- Stokastik simülasyon modelinin her çalıştırılması sonucunda, belirli bir grup giriş parametreleri için modelin doğru karakteristiklerinin bir tahmini yapılır. Çalışılan her bir giriş parametre grubu için modelin birçok kere bağımsız çalıştırılması gerekecektir. Eğer sistemin bir analitik modeli kurulabiliyor ve geliştirilebiliyorsa bu yöntem, simülasyon modeli kullanımına tercih edilmelidir.
- Simülasyon çalışmasının sonucunda birçok sayının üretilmesi ve gerçekçi animasyonun yarattığı ikna edici etki genellikle, çalışmanın sonuçlarına ilişkin büyük bir güvenin oluşmasına meyil yaratmaktadır. Eğer model, çalışılan sistemin uygun bir gösterimi değilse, simülasyon sonuçları, gerçek sistem hakkında çok az faydalanılabilecek bilgi sağlayacaktır (Kuş 2003).
- Simülasyon bilgisayar verileri hatalı olduğunda tam olarak doğru sonuçları veremez. Model tam olarak doğru bilgisayar verilerini içermiyorsa, çalışanların doğru üretim verilerini (çıktı) sağlamak adına makul beklentileri olamaz. Maalesef bilgisayar sürecinin en zor kısmının veri toplama olduğu düşünülür. Bu yaygın inanışa rağmen, bu süreç için normalde ayrılan zaman çok kısadır. Birçok çalışanın günlük/sıradan verileri toplamaktan çok bir simülasyon modelini geliştirmeyi tercih etmesi bu problemi daha da zorlaştırmaktadır.
- Simülasyon problemleri kendiliğinden çözememektedir: Bazı yöneticiler, simülasyon modeli yönetiminin ve proje analizlerinin problemleri çözeceğine inanırlar. Tek başına simülasyon probleme çözüm getirememektedir. Yöneticilere problemler için potansiyel çözümler temin eder. Amaçlanan değişiklikleri yerine getirmede yöneticilere bireysel sorumluluklar yükler.
- Simülasyon karmaşık problemlere kolay cevaplar sağlayamaz: Bazı analistler, simülasyon analizlerinin karmaşık problemlere basit cevaplar geliştireceğine inanabilirler. Aslında, bu daha çok karmaşık problemlerin karmaşık cevapları olduğu anlamına gelir. Eğer sistem analizi birçok unsur ve etkileşimi içeriyorsa, en iyi operasyon ve strateji kaynakları alternatifi sistemin her unsurunun düşünülmesidir. Basitleştirilmiş varsayımları yapmak uygun bir zamanda uygun bir modelin geliştirilmesi ile mümkündür. Ancak, eğer sistemin kritik unsurları dikkate alınmazsa o operasyon veya strateji kaynaklarının etkinliği azalır.

2.3 Sistem ve Modelleme

2.3.1 Sistem

Sistem literatürde amaca veya amaçlara göre değişik şekillerde tanımlanmaktadır.

Bunlardan bazıları:

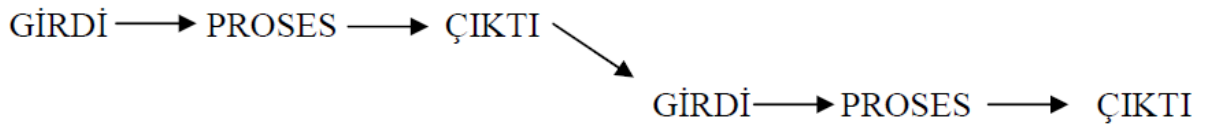
Sistem, mantıksal bir sonuca ulaşmada etkide bulunan ve birbirleriyle etkileşimi olan varlıklar (insan ya da makine) topluluğu olarak tanımlanmaktadır (Law ve Kelton 2000).

Belirli girdileri alan ve bunlara uygun olarak işleyiş göstererek belirli çıktılar arasındaki ilişkiyi gösteren bir fonksiyonu en büyüklemeyi amaçlayan varlıklar veya elemanlar topluluğudur.

Bu tanımlar ışığında sistem, birbiriyle ilişkili parçaların meydana getirdiği her şeydir.

Ancak bu parçaların arasındaki dinamik ilişkiler ve organizasyonel bağıntılar dikkate alınmadığında hiç bir anlam ifade edilemez. Yani tam olarak statik sistemler sistem sayılmazlar.

Sistemin temel yapısı girdi ve çıktılarından oluşmaktadır (Şekil 2.1).



Şekil 2.1 Sistem kavramında girdi ve çıktı yaklaşımı

Görülen sistemin girdilerinden her birinin belki de bir başka sistemin çıktısı olabileceğidir. Bu durum da yaklaşımların daha geniş düşüncelerle tasarlanması gerektiğini ortaya koyar. Sisteme bu açıdan bakacak olursak şunları görürüz:

1. Sistemin amacı,
2. Sistemin çevresi,
3. Sistemin kaynakları,
4. Sistemin bileşenleri,
5. Sistemin iş yönetimi.

Yapılan açıklamalar doğrultusunda sistemin temel nitelikleri üç grupta toplanır:

1. Amaçların olması,
2. Sistem öğelerinin anlamlı bir biçimde ilişkilenmiş olması,
3. Bilgi enerji malzeme benzeri girdilerin bulunması.

2.3.2 Sistemin Bileşenleri

Bir sistemi anlamak ve analiz etmek için, çok sayıda terim tanımlanmıştır. Bu terimleri eleman, özellik, faaliyet, durum, olay, sistem içi, sistem dışı olarak sınıflandırabiliriz.

Eleman, sistemdeki ilgili unsurdur.

Özellik, bir elemanın niteliğidir.

Faaliyet, belirli uzunluktaki bir zaman periyodunu temsil eder. Bir banka örneği göz önüne alındığında, müşteriler elemanlardan biri, hesap bakiyeleri bir özellik ve para yatırma bir faaliyet olabilir(Law and Kelton 2000).

Bir çalışmadaki sistemi oluşturan elemanların kümesi, başka bir çalışmadaki tüm sistemin bir alt kümesi olabilir.

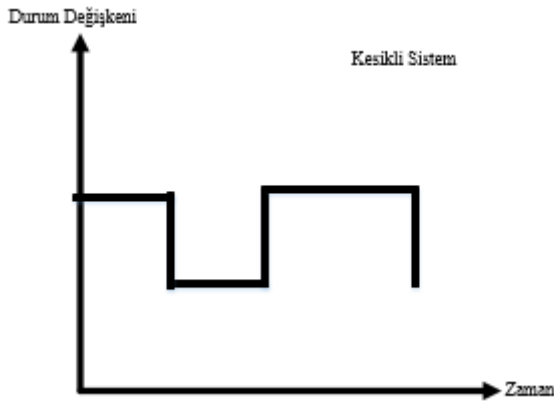
Sistem durumu, çalışmanın amacına bağlı olarak, herhangi bir anda sistemi tanımlamak için gerekli değişkenler kümesi şeklinde tanımlanabilir. Banka örneğindeki durum değişkenleri, meşgul veznedarların sayısı, hatta bekleyen veya servis gören müşterilerin sayısı ve bir sonraki müşterinin geliş zamanıdır.

Olay ise, sistemin durumunu değiştirebilen ani oluşumdur.

Sistem içi terimi, bir sistem içinde ortaya çıkan faaliyetleri ve olayları tanımlamak için; sistem dışı terimi ise, sistemi etkileyen çevredeki olayları ve faaliyetleri tanımlamak için kullanılmaktadır.

2.3.2.1 Sürekli ve süreksiz (kesikli) sistem

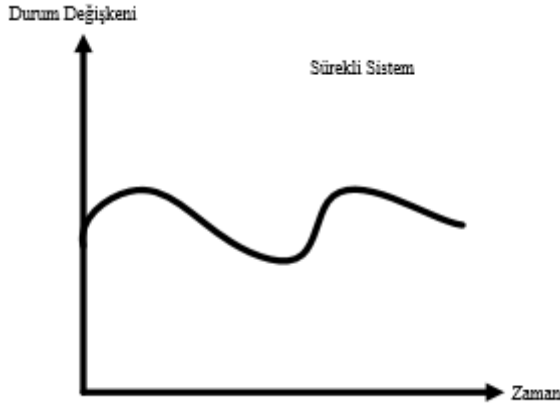
Sistemler, kesikli veya sürekli olarak sınıflandırılabilir. Pratikte çok az sayıda sistem tamamen kesikli veya sürekli; çünkü tek bir değişim tipi sistemde etkili olduğundan, bir sistemi kesikli veya sürekli olarak sınıflandırmak mümkündür (Law and Kelton 2000). Kesikli sistem, durum değişkeninin (değişkenlerinin) zamana göre sadece kesikli nokta kümesinde değiştiği bir sistemdir (Şekil 2.2). Banka, kesikli sisteme bir örnek olarak gösterilebilir, çünkü durum değişkeni (bankadaki müşteri sayısı), bir müşteri geldiğinde veya müşteriye verilen servis tamamlandığında değişmektedir.



Şekil 2.2 Durum değişkeninin kesikli sistemde değişimi

Sürekli sistem ise, durum değişkeninin (değişkenlerinin) zamana göre sürekli olarak değiştiği bir sistemdir. Sürekli sisteme örnek olarak, bir barajdaki su yüksekliği gösterilebilir. Yoğun yağmur yağışından sonra sular, barajın arkasındaki göle akmaktadır. Su baskısını kontrol etmek ve elektrik üretmek için sular barajdan çekilmelidir. Burada, buharlaşma da su düzeyini

azaltmaktadır. Böylece, durum değişkeni olan barajdaki su yüksekliğinde, bu etkiler altında sürekli bir değişim olmaktadır. Şekil 2.3'te sürekli durum değişkenlerinin bir zaman periyodunda nasıl değiştikleri görülmektedir.



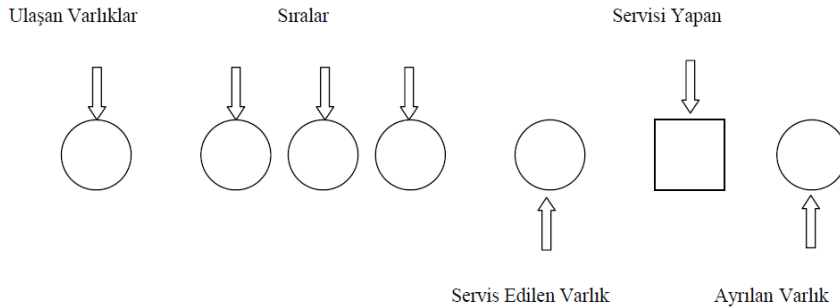
Şekil 2.3 Durum değişkeninin sürekli sistemdeki değişimi

2.3.2.2 Sistemin yapısal elemanları

Sistemi anlamak ve analiz etmek için, birçok sayıda terimin tanımlanması gerekmektedir. Varlık, sistemde ilgilenilen nesnedir. Nitelik ise, varlığın özelliğidir (Gordon 1978). Hareket ise, belirlenen uzunluktaki zaman periyodunu ifade eder.

Sistem durumu, herhangi bir zamanda sistemi tanımlamak için gerekli olan değişkenler topluluğu olarak tanımlanır ve çalışmanın amacına bağlı olarak değişmektedir. Olay ise zamanın belirli noktalarında sistemin durumunda değişikliğe neden olan durumdur (Ingalls 2001).

Varlıklar: Varlıklar, sistemin durumunu değiştiren şeylerdir. Bazı durumlarda, özellikle servis sistemlerini içerenlerde, varlık bir insan olabilir. Müşteri servis merkezlerinde, varlıklar müşterilerdir. Varlıkların insan olmaları ise zorunlu değildir, nesne de olabilirler. Şekil 2.4'te verilen fabrika örneğinde varlıklar, makine olmak üzere bekleyen parçalardır.



Şekil 2.4 Basit simülasyon modeli unsurları

1.Varlık takımları: Varlıkların numaralandırılmaları, aynı zamanda sisteme ulaşan takımların büyüklüğüne göre yapılır. Bazı sistemlerde, takım büyüklüğü her zaman birdir. Bazılarında

ise varlıklar deęişik büyüklükteki takımlar halinde olabilirler. Örneęin, bir tiyatroya giden aileler birer varlık takımıdır ve büyüklükleri birbirlerinden farklıdır. Takım büyüklüğü iki, üç, dört veya daha fazla olabilir.

2. Varlıkların sisteme ulaşma zamanları: Takımların gelişleri arasında geçen zaman “varış zamanı” olarak bilinir. Normal takım büyüklüğünün bir veya daha fazla olması fark etmez, burada dikkate aldığımız sadece son takımın sisteme ulaşmasından sonra gelecek olan takımın sisteme ulaşma süresidir. Önceki takım sadece bir varlığa sahipken, sonraki takım birden fazla varlığa sahip olabilir. Ayrıca varlıkların varış zamanı, varlıkların varış zamanlarının karşılıklı oranlamaları şeklinde de olabilir.

Takımların varış zamanlarını biriktirmek, varlıkların varış zamanlarını biriktirmekten daha kolaydır. Bununla beraber, eski bazı veriler varışların genel oranları şeklinde oluşturulmuş olabilir. Varış zamanları, uygulayıcının model için sağlayacağı bilgisayar verilerini de dikkate alır.

3. Varlık özellikleri: Varlıklar, özelliklere sahiptirler. Bunlar sistemdeki her bir varlık için deęişken deęerlere sahiptirler. Varlık özellięi, varlıklar için farklı deęerde olabileceęi gibi aynı deęerde de olabilir.

Varlıklar, sisteme ulaştıkça her bir varlığın özellięi olan varış zamanları biriktirilerek simülasyon sisteminin zamanı oluşturulacaktır. Bir varlık takımı ulaştığında ise takımdaki her varlık aynı varış zamanına yani aynı özellięe sahip olacaktır. Bazı özellikler aynı deęere sahip olabilirler.

Simülasyon programları global deęişkenlerden de faydalanırlar. Global deęişken nicelikler, varlık özellikleri ile karıştırılmamalıdır. Bu deęişken nicelikler tek bir deęere sahip olmaları yönünden varlık özelliklerinden farklıdırlar. Global deęişkenin simülasyon programlarındaki tipik kullanımı simülasyonun çalışma süresidir (Chung 2004).

Nitelikler: Varlıklar sadece kendilerine ait olan niteliklere sahiptirler. Bu nedenle nitelikler lokal deęerler olarak algılanmalıdır. Örneęin, kırmızı ve mavi parçaların üretimi yapılacaksa, böyle bir durumda nitelik parçaların rengi olacaktır. Eęer parçaların sistemdeki zamanları önemli ise bu durumda parçaların renkleri bir önem arz etmeyecektir. Bu örnekten de anlaşılabilen gibi bir ya da birden fazla varlık aynı özellięe sahip olabilir.

Kaynaklar: Kaynak, dinamik varlıklara servis veren bir varlıktır. Kaynak aynı anda bir ya da birden fazla varlığa hizmet verebilir. Dinamik bir varlık ise bir kaynağın bir ya da birden fazla birimine ihtiyaç duyabilir. Ya da varlık kuyrukta herhangi bir işlem için sıraya girebilir. Eęer varlık kaynakta işlenmeye başlarsa, işlem süresi kadar kaynağı meşgul eder ardından da kaynağı bırakıp gider. Kaynakların mümkün olan birçok durumları bulunmaktadır. Kısaca, bu durumlar boş ya da dolu olmasıdır. Fakat iptal, bozuk ya da bloke olmuş gibi dięer durumlar da söz konusudur.

Kuyruklar (Sıralar): Kuyruklar, varlığın başka bir hareketin başlaması için şartların deęişmesini bekledięi pasif durumlardır (Carrie 1988). Varlıklar genelde kendi işlemleri başlayana kadar kuyrukta beklerler. Beklemeler ise bazı sistem şartlarının

kombinasyonlarından kaynaklanan tanımsız gecikmedir. Bir varlık herhangi bir kaynakta işlem görmek amacıyla kuyrukta bekliyorsa, kuyrukta bekleme süresinin ne kadar olacağı başka olayların gerçekleşmesine bağlıdır. Basit sistemlerde genellikle “ilk giren ilk çıkar” (FIFO) (first-in-first-out) sıra üstünlüğü prensibi kullanılır. Basit sistemlerin bir diğer karakteristik özelliği de varlıklardan birinin sisteme girdiğinde, kuyruğa da girmiş olması gerektiği durumudur. Ayrıca varlıklardan biri kuyruğa girdinde, servise kabul edilmeden sistemden ayrılamazlar (Chung 2004).

2.3.2.3 Sistemin dinamik elemanları

Olaylar: Olaylar sistemdeki durum değişkeninin değiştiği durumlardır. Herhangi bir kaynağın bir varlığı işlemesi ile meşgul olması ve daha sonra işlem bittikten sonra ise boş kalması olaya örnek olarak verilebilir.

Faaliyetler: Her bir olayda başlanılan işlem ya da prosedürler faaliyet olarak adlandırılmaktadır. Faaliyetler sistem varlıklarının dönüşüm işleminden sorumludurlar.

Prosesler: Proses basitçe ardışık olaylar grubudur. Örneğin, bir hammadde iş istasyonuna gelir, burada işlenir ve ardından diğer iş istasyonuna gönderilir.

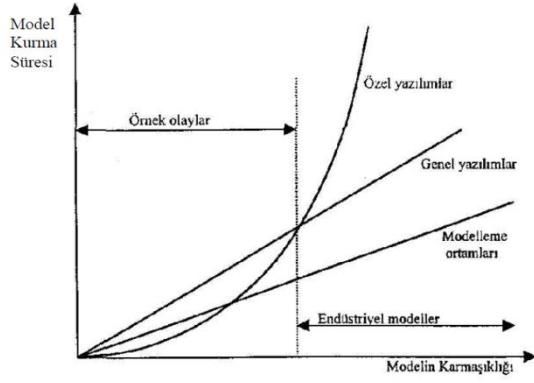
Simülasyon saati: Bu, simülasyon ilerledikçe, simülasyon süresinin o anki değerini tutan simülasyon model değişkenidir. Bu değişkenin değerine göre, uygun prosesler simülasyon işleminde sıfırlanırlar.

2.3.3 Model

“Model” ve “sistem” kelimeleri, simülasyonun tanımlanmasında kullanılan anahtar terimlerdir. “Model” terimi ile varlık formundan farklı olarak nesnelere ve fikirler grubu ifade edilmektedir. “Sistem” terimi ile belirli bir amacı başarmak için beraber çalışan ve birbirleri ile etkileşimi olan elemanlar grubu ya da topluluğu ifade edilir (Law ve Kelton 2000).

Sistemlerin zaman içerisindeki davranışını incelemek amacıyla simülasyon modelleri oluşturulur. Genellikle bu modeller, sistemin operasyonlarını içeren bazı varsayımlardan oluşur. Bu varsayımlar, sistemin varlıkları ya da ilgilenilen nesnelere arasındaki matematiksel, mantıksal veya sembolik ilişki formunda ifade edilirler. Bir model geliştirilip, doğruluğu ispatlandıktan sonra gerçek sistemin işleyişi tahminlenebilir. Sistemde olan potansiyel değişimlerin, sistem performansını ne ölçüde etkilediklerini görmek amacıyla simülasyon yapılabilir. Ayrıca herhangi bir sistemin oluşturulma işleminden önce, simülasyon işlemi yapılarak tasarım aşamasındaki sistemin gerçekte nasıl işleyeceği önceden tahminlenebilir. Böylece, simülasyon modelleme, var olan sistemlerde yapılan değişikliklerin etkilerinin ve değişen koşullar altında yeni sistemlerin performansının belirlenmesi amacıyla kullanılmaktadır.

Karmaşık sistemlerde ele alınan ana problemler, bunların boyutları, operasyonların anlaşılabilirliği, sistemin verimliliğinin artırılması ve sistemin performansının hesaplanması olarak karşımıza çıkmaktadır (Law ve Kelton 2000). Modellenen sistemin karmaşıklığına ve seçilen yazılımın tipine bağlı olarak model geliştirmenin beklenen süresinin bir diyagramı Şekil 2.5'te görülmektedir.



Şekil 2.5 Modelin karmaşıklığı ile modelleme süresinin karşılaştırılması (Law ve Kelton 2000)

2.3.3.1 Model Türleri

Modeller, “matematiksel” veya “fiziksel” modeller olarak sınıflandırılabilir. Bir matematiksel model, sistemi temsil etmek için sembolik notasyon ve matematiksel denklemleri kullanır. Simülasyon modeli, belirli tipte bir matematiksel sistem modelidir.

Simülasyon modelleri, statik veya dinamik, deterministik veya stokastik ve kesikli veya sürekli olarak sınıflandırılabilirler.

“Monte Carlo” simülasyonu olarak bilinen statik simülasyon modeli, zamanın belirli bir anındaki sistemi temsil etmektedir.

Dinamik simülasyon modelleri ise, zamana göre değişen sistemleri temsil etmektedir.

Rastsal değişken içermeyen modeller, deterministik modeller olarak sınıflandırılırlar. Deterministik modeller, tek bir çıktı kümesi veren girdi kümesine sahiptir. Deterministik modellere örnek olarak, tüm hastaların randevu saatlerine göre geldikleri bir dışı muayenehanesini gösterebiliriz.

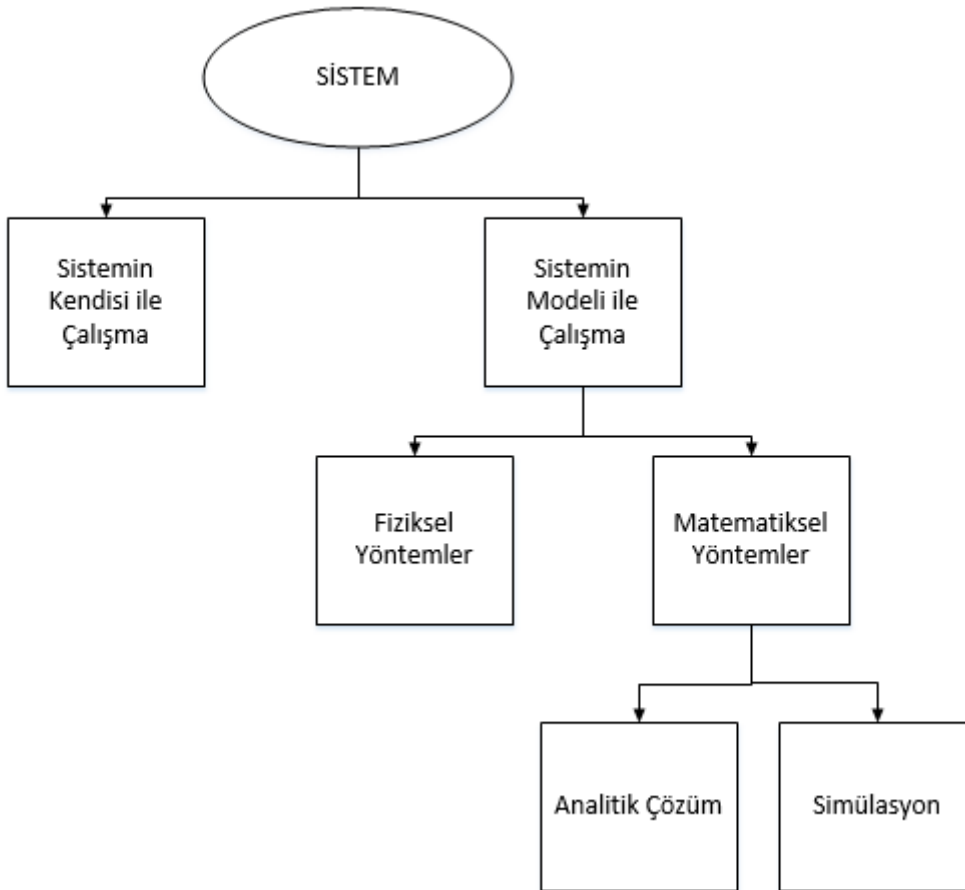
Stokastik simülasyon modeli ise, girdi olarak bir veya daha fazla rastsal değişkeni göz önüne almaktadır. Rastsal girdiler, rastsal çıktılar oluştururlar. Çıktılar rastsal olduğu için, bu çıktılar modelin gerçek karakteristiklerinin tahminleri olarak göz önüne alınabilir. Bir bankanın simülasyonu, genellikle rastsal gelişler arası süreleri ve rastsal servis sürelerini kapsamaktadır. Bu nedenle, stokastik bir simülasyonda, çıktı ölçütleri (bekleyen ortalama müşteri sayısı, bir müşterinin ortalama bekleme zamanı) sistemin gerçek karakteristiklerinin istatistiksel tahminleri olarak ele alınırlar.

Kesikli ve sürekli modeller, analog bir şekilde tanımlanmıştır. Ancak, kesikli bir simülasyon modeli kesikli bir sistemi modellemek için; sürekli bir simülasyon modeli de sürekli bir sistemi modellemek için her zaman kullanılmaz. Ayrıca simülasyon modelleri, kesikli ve sürekli şekilde karma modeller olabilirler. Kesikli veya sürekli (veya hem kesikli hem sürekli) simülasyon modelini kullanma seçimi, sistem karakteristiklerinin ve çalışma amacının bir fonksiyonudur. Bu nedenle, her mesajın karakteristiğinin ve hareketinin çok önemli olduğu bir iletişim kanalı kesikli olarak modellenebilir. Aynı şekilde, kanaldaki mesajların akışı

önemli olduğunda, sürekli simülasyon kullanarak sistem modelleme daha uygun olmaktadır (Banks ve Carson 1984).

2.3.4 Sistemlerin İncelenmesi ve Modelleme

Sistemi incelemek iki yolla mümkündür; birincisi gerçek sistemde deneyler yapmak, ikincisi ise sistemin modeli ile deney yapmaktır (Gürkan 2004). Bir sistemi incelemek için, sistemin kendisi ile deney yapmak her zaman geçerli değildir. Yeni bir sistem oluşmadan önce sistem, teorik formda veya tasarım safhası şeklindedir. Mevcut sistem ile deney yapmak pratik bir yöntem değildir. Örnek olarak, enflasyon altında çalışanların etkilerini belirlemek için işsiz kişilerin oranını iki katına çıkarmak mümkün değildir. Banka örneğinde, bekleme hattı uzunluğunun etkisini incelemek için veznedar sayısını azaltmak, müşterileri memnun etmez ve bu nedenle müşteriler, hesaplarını rakip bankaya yatırır. Sonuç olarak, sistemlerin incelenmesi sistemin bir modeli ile yapılmaktadır (Şekil 2.6).



Şekil 2.6 Sistem çalışma yöntemleri

Sistemin modelini incelemek ise fiziksel ya da matematiksel yöntemlerle mümkündür. Matematiksel yöntemler ise, analitik çözümler ya da simülasyon uygulamalarıdır. Simülasyon da deneysel bir yöntemdir. Gerçek sistem ile deney yapmak yerine deneyler simülasyon modeli üzerinde yapılır.

Deney yöntemi, en sağlıklı yöntemdir. Mümkün olduğu sürece kullanılmalıdır. Bu yöntemin kullanılmasına uygun olmayan durumlar:

- Çok tehlikeli alanlar (Kritik duruma varmış bir nükleer reaktör santralının davranışlarında, bir jet motorlu uçağın inişinde vs.)
- Çok pahalı alanlar (Hasara sebebiyet verecek tüm durumlarda, kiralık telefon hattı ağı üzerinden verilerin aktarımı için uzun deneysel çalışmalar vs.)
- Sistemin araştırılmasına imkan vermeyen durumlar (Dizayn seviyesinde mümkün olan birçok alternatiflerin değerlendirilmesi)

Analiz yöntemi, pratik yaşamda nadiren gerçek olan ve birçok kabule dayanan ve çoğu kez matematiksel olarak yapılan bir benzetim tipidir. Analitik yöntemlerin dezavantajları çok karmaşık cihazların kullanılması ve/veya hesaplamalarda çok uzun zaman harcanmasıdır. Queueing ağ analizi buna bir örnektir. Diğer taraftan formülün kullanılması genelde hızlı sonuç verir ve formüle basit farklı parametre değerleri ilave edilerek alternatif birçok durumlar test edilebilir. Deneysel yöntemler için çok daha fazla zaman harcandığı durumlar ortaya çıkabilir. Analizin diğer bir problemi, gerekli parametrelerin yokluğudur. Diğer benzetim sistemlerinin tahmin edilen verileri kullanması, sonuçların güvenilirliğini azaltmaktadır (Varol 1999).

Bazı durumlarda, matematiksel yöntemlerle çözümlenebilen modeller geliştirilebilir. Bu çözümlere, diferansiyel denklemlerin, olasılık teorisinin, cebir metotlarının ya da diğer matematiksel tekniklerin kullanımı ile ulaşılabılır. Genellikle çözümler sistem performans ölçüsü olarak adlandırılan, bir ya da birden fazla nümerik parametre içerir.

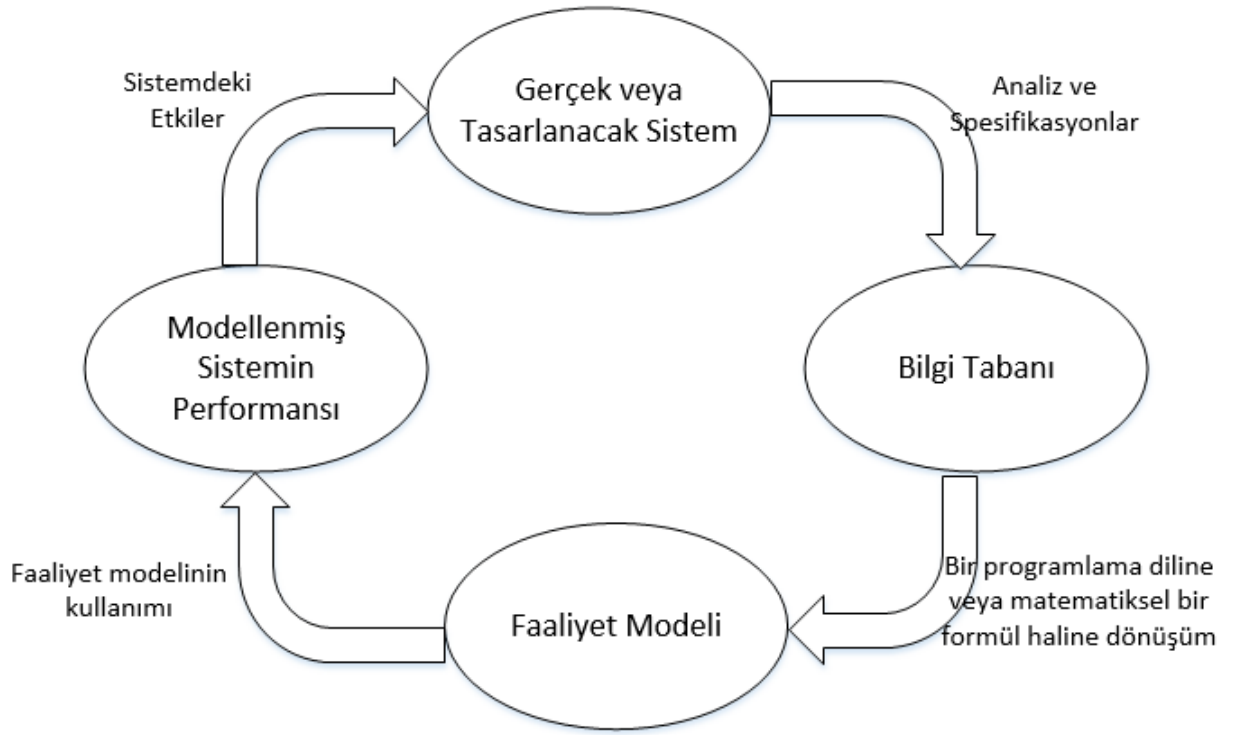
Gerçekte ise sistemlerin çok karmaşık yapıya sahip olmaları nedeniyle matematiksel yöntemlerle çözüme ulaşmak mümkün olmamaktadır. Bu gibi durumlarda ise, nümerik, bilgisayar tabanlı simülasyonun kullanımı ile sistemin zaman içerisindeki davranışı ifade edilir. Simülasyon işleminde incelenen gerçek sisteme ait bilgiler toplanır. Bu elde edilen bilgiler ise sistem performans ölçütlerinin tahminlenmesi için kullanılır.

Bir sistemin performansı ile ilgili bir çalışmada, öncelikle bu sistemin bir modelinin oluşturulması gereklidir. Bir model fikrinde birbiriyle bağlantılı üç kavram olduğunu ileri sürmüştür. Bu üç kavram ise şunlardır:

Bir model gerçek bir sistem ile benzeşmelidir,

Bir model gerçek bir sistemin basitleştirilmiş bir teşkili olmalıdır,

Bir model gerçek bir sistemin ideal bir hali olmalıdır.

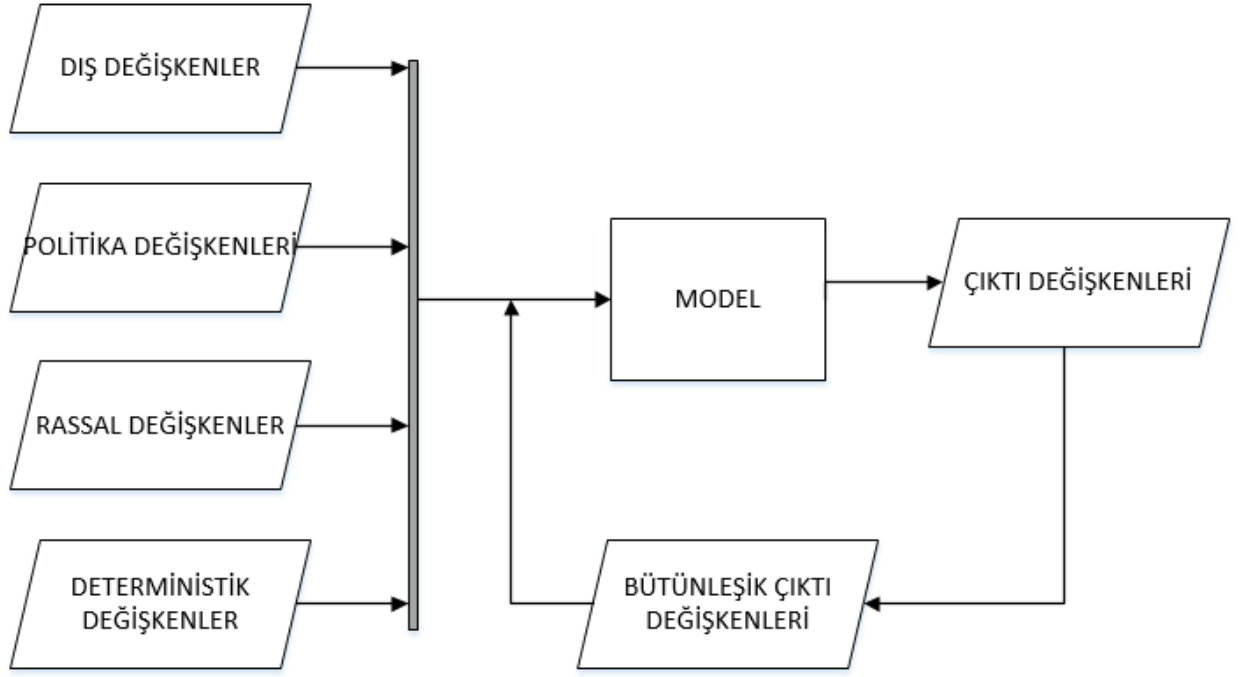


Şekil 2.7 Modelleme işlemi

Pratikte bir model, modellenecek olan gerçek bir sistemin gözlemleri baz alınarak oluşturulur. Bir bilgi modeli, faaliyet modeli ve modelleme işlemi arasındaki ilişkiler Şekil 2.7’de görülmektedir. Şekilde görülen iki kavramdan biri olan bir sistemin bilgi modeli, sistemin yapısının ve fonksiyonlarının doğal veya grafik bir dille formüle edilmesidir. Eğer bir sistem mevcutsa, bir bilgi modeli gözlem safhası esnasında elde edilmiş olan bilgilerin tümünü içermektedir. Eğer bir sistem mevcut değilse, o zaman bir bilgi modeli, topoloji ve tasarımcıların işlem spesifikasyonlarını kapsar. Faaliyet modeli ise, bilgi modelinin bir matematiksel formülasyona veya bir programlama diline dönüştürülmesidir.

2.3.5 Bir Simülasyon Modelindeki Değişkenler

Simülasyonda kullanılan değişkenler çıktı, bütünlük çıktı, dış politika değişkenleri veya rastsal değişkenler olarak sınıflandırılabilirler. Birçok simülasyon modeli probabilistik olmadığından, deterministik değişkenleri de göz önüne almak gerekmektedir. Şekil 2.8 simülasyon modellerindeki bu değişkenlerin rolünü göstermektedir (Watson ve Blackstone 1989).



Şekil 2.8. Simülasyon modellerinde kullanılan değişkenler

2.3.5.1 Çıktı değişkenleri

Çıktı değişkenleri, analiste ve/veya yönetime gerekli ara sonuçları veya son bilgileri sağlamaktadır. Bir üretim modelinde çıktı değişkenine örnek olarak, satılan mamullerin maliyeti ve envanterdeki yarı mamulle ilgili bilgiler gösterilebilir.

Bütünleşik çıktı değişkenleri

Daha önce de belirtildiği gibi, simülasyon modelleri dinamiktir ve zamana göre sistemin davranışını tanımlamaktadır. Sonuçta, zaman sonunda sistemin durumu, bir sonraki zaman periyodu boyunca sistemin analizinde gerekli bir girdi olmaktadır.

Bütünleşik çıktı değişkenleri, bir zaman periyodundan diğerine sistemin durumu hakkında bilgi vermektedir. Bütünleşik çıktı değişkenlerine (seri ilişkili değişkenler) ait örnekler üretim, finans ve pazarlama modellerinde ortaya çıkmaktadır. Periyot sonu bitmiş mamul envanteri, gelecek periyodun gelecek mamul envanterinin belirlenmesinde gerekli bir girdidir.

Dış değişkenler

Sistemin davranışını etkileyen bu değişkenler, dış değişkenler olarak belirtilmektedir. Dış değişkenlerin yapışma örnek olarak şu önermeyi kullanabiliriz: “X, Y’den etkilenir; ancak Y, X’den etkilenmez”. Dış değişkenler çoğunlukla sistem dışı, çevresel veya kontrol edilemeyen değişkenler olarak adlandırılırlar. Bir üretim modelinde, grevler ve hammadde eksikliği, dış değişkenler olarak göz önüne alınabilir.

Politika deęişkenleri

Birçok sistem, yönetimin incelediđi kontrol faktörlerini içermektedir, örnek olarak bir organizasyonda yönetim, hangi şartlar altında hangi makinenin alınacağına karar verebilir. Bu şekildeki faktörler, simülasyon modellerindeki politika deęişkenleri olarak göz önüne alınırlar. Bu faktörler, yönetimin kararlarına ve kontrolüne göre deęiştii için, politika deęişkenleri bazen karar veya kontrol edilebilir deęişkenler olarak da adlandırılırlar. Birçok simülasyon modelinde politika deęişkenlerinin hangi sistem politikasında daha etkin olduđu araştırılır.

Rastsal deęişkenler

Bazı sistemlerin davranışını belirlemek için, bunların rastsal veya probabilistik yapısını anlamak gerekir. Rastsal deęişkenler, bu role hizmet etmektedir.

Deterministik deęişkenler

Birçok sistem probabilistik olmasına rağmen analist, deterministik bir modelin kullanıcının amaçları için yeterli bilgileri sağlayacağını bilir. Bu durumda, sadece tek deęerli tahminler gerektiren deterministik deęişkenler kullanılır. Sistemdeki varyasyon miktarının çok az olduđu probabilistik modellerde, bazen deterministik deęişkenler de kullanılmaktadır.

2.3.5.2 Süreksiz Sistemlerde Simülasyon

Daha önce belirtildiđi gibi süreksiz bir sistemde durum zamana bađlı olarak düzensiz bir şekilde deęişmektedir. Sistem; çeşitli öğelerden, öğelerin birbiri ile olan ilişkilerinden ve sistemin çevre ile olan sınırlarından oluşur. Sistemin sınırları dışında kalan öğeler ve faaliyetler sistem çevresini oluştururlar. Bu çevre boş ise sistemin “kapalı” olduđu söylenir. Sistemin öğeleri, bunların nitelikleri, olaylar, faaliyetler ve bunlar arasındaki ilişkiler sistem modelinde tanımlanır.

Her simülasyon çalışmasında “simülasyon saati” ya da “iç saat” denen bir durum deęişkeni kullanılır. Simülasyon zamanı iki yaklaşım kullanılarak ilerletilir. Birincisi, sabit artımlı ilerleme; ikincisi, bir sonraki olay zamanıdır:

Sabit artımlı ilerleme: Zaman daima eşit aralıklarla ilerletilerek inceleme yapılır. Bu aralıkta meydana gelen olaylar sanki aralığın sonunda meydana gelmiş gibi düşünülür. Bu yöntem, sabit uzunluktaki aralarla meydana gelen olayların olduđu sistemler için faydalıdır.

Bir sonraki olay zamanı: Bir sonraki olay zamanı ilerlemesi daha yaygın kullanılan bir yaklaşımdır. Zaman mevcut olay ile bir sonra planlanan olay arasında ilerler. Böylece simülasyon, faaliyet olmayan süreleri atlamış olur. Burada kronolojik bir sırayla düzenlenmiş olaylar ve zamanlar takvimi vardır.

Süreksiz olay sistem simülasyonunun geliştirilmesinde kullanılan kavramlar

Bir sistemin kesikli olay modelini geliştirmek için bazı kavramlar kullanılmaktadır. Bu ana kavramlar Çizelge 2.1'de kısaca tanımlanmıştır: Kümeler bazı durumlarda listeler, kuyruklar veya zincirlerdir. Bir faaliyet deterministik veya olasılıđa dayalı olabilir veya herhangi bir

matematiksel fonksiyondur. Bir müşterinin kuyrukta beklerken harcadığı zaman, gecikmeye tipik bir örnektir. Sistemler zaman boyunca değiştiğinden dinamik bir yapıdadır. Bu yüzden zaman boyunca sürekli olarak değişen sistem durumu, gezen birimlerin nitelikleri ve aktif gezen birimlerin sayısı, kümelerin içerikleri, faaliyet ve içeriklerin tümü zamanın bir fonksiyonudur. Bütün bu zamanların hepsi, bir saat değişkeniyle temsil edilmektedir.

Çizelge 2.1. Kesikli olay modellemedeki bazı kavramlar

Kavram	Açıklama
Sistem	Bir zaman dilimi boyunca bir veya birden çok amacı başarmak için birbiriyle etkileşim halindeki varlıkların (gezen birimler, yani insan, makineler vb.) tümü.
Sistem Durumu	Herhangi bir zamanda bir sistemi tanımlamayı gerektiren tüm bilgileri kapsayan değişkenlerin tümü
Model	Sistemin özet bir temsilidir ve çoğunlukla sistem durumu, gezen birimler ve onların nitelikleri, kümeler, olaylar, faaliyetler ve gecikmeler terimleriyle tanımlanan bir sistemdeki mantıksal ve/veya matematiksel ilişkileri kapsamaktadır.
Gezen Birim (Varlık)	Bir model ile temsil edilen bir sisteme ihtiyaç gösteren sistem içindeki herhangi bir nesne veya bileşen (bir servisçi, bir müşteri, bir makine).
Nitelikler	Verilen bir gezen birimin özelliği
Küme	Bazı mantıksal davranışlar ile düzenlenmiş birbiriyle benzeşen gezen birimlerin tümü.
Olay	Bir sistemin durumunda aniden meydana gelen değişim (sisteme yeni bir müşterinin gelmesi gibi).
Faaliyet	Belirli uzunluktaki bir zaman süresi.
Gecikme	Belirsiz uzunluktaki bir zaman süresi.

2.3.5.3 Süreksiz sistemlerde ortam yaklaşımları

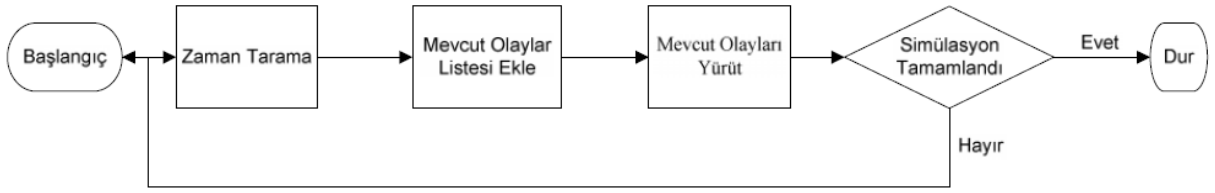
Süreksiz sistemler simülasyonunda 3 temel yaklaşım vardır. Simülasyon dilleri bu üç temel yaklaşımdan birini ya da birkaçını kullanırlar. Bu yaklaşımlar;

- Olay programlama,
- Faaliyet (aktivite) arama,
- Proses etkileşimleri.

Tüm simülasyon programlarında, olayları sıraya koyan ve simülasyonu çalıştıran bir “yürütücü” bölüm vardır. Bu “yürütücü” bölüm bir sonraki olayı getirir, simülasyon zamanını ilerletir, uygun çalışma programlarına transferleri kontrol eder. Yukarıda sözü geçen yaklaşımlar arasındaki fark, bir sonraki olayın seçimindeki stratejiden kaynaklanmaktadır. Aşağıda bu yaklaşımlar açıklanmaktadır:

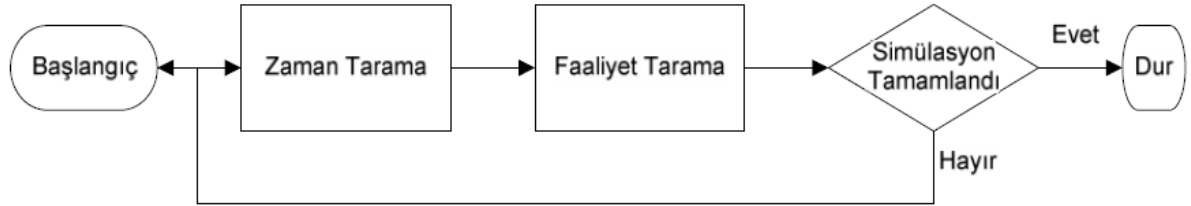
Olay programlama: Olay programlama yaklaşımı; bir simülasyon dili ya da olayların ve bunların sistem durumları üzerindeki etkilerinin incelendiği bir yaklaşımdır. “Yürütücü” program içinde, bir olay meydana geldiğinde bu olayı temsil eden bir program bulunmaktadır.

Şekil 2.9’da bir şema olarak olay programlama görülmektedir.



Şekil 2.9 Olay programlama yaklaşımı

Faaliyet arama: Burada, yürütücü program, takvimden bir sonraki olayı seçmede, planlama zamanı ve durum testini kullanır. Modelin temeli, belli bir süresi olan faaliyetleri aramadır. Bir faaliyet genellikle, biri faaliyetin başlangıcını diğeri sonunu temsil eden iki olay ile temsil edilir. Her faaliyet test koşullarını da içerir. Koşullar doğru ise faaliyeti gerçekleştirmek üzere nelerin yapılması gerektiği belirlenmiştir. Burada da yürütücü program faaliyetleri, test koşullarını yerine getirip getiremediği açısından ve zamana uygunluk yönünden incelenir. Sonra en uygun faaliyet yerine getirilir. Bir aktivitenin yürütülmesi tamamlandıktan sonra tarama tekrar başlar. Bu işlem simülasyon sona erene kadar devam eder (Şekil 2.10).



Şekil 2.10 Faaliyet arama yaklaşımı

Proses etkileşimi: Bu yaklaşım, olay programlama ve faaliyet arama yaklaşımının özelliklerini birleştirir. Simülasyon; olaylar ya da faaliyetlerden daha çok bir dizi işlemlerden oluşmaktadır. İşlem, bir elemanın var olduğu süre içinde yer alacağı operasyonları tanımlayan ifadeler dizisidir.

Proses etkileşimi yaklaşımını kullanan bir model, bir elemanın işlem süresince hareketlerini gösteren bir akış seması ile temsil edilebilir. Bir öge, işlem esnasında çeşitli noktalarda, bazı nedenle bloke edilebilir veya ertelenebilir. Bir şarta bağlı olmayan ertelemeler, belli bir zaman geçtiğinde meydana gelir. Bir şarta bağlı olan itelemelerde ise devamlılık sistemin durumuna bağlıdır. Örneğin bir eleman; uygun şartlar oluşuna veya planlı bir ertelemenin sonunda yeniden faal hale getirene kadar bir erteleme noktasına tutulabilir.

Proses etkileşim stratejisini uygulamak için birçok yol vardır. En yaygın kullanılan yaklaşım GPSS dedir ve Zeigler tarafından "süreç etkileşim prototipi" adlı çalışmasında kullanılmıştır. Bu yaklaşım; gelecekte uygulanmak üzere planlanmış bir gelecek olaylar listesi (GOL-FEL) ve o anda uygulanabilecek tüm olaylar için, mevcut olaylar listesi (MOL-CEL) kullanılır. Simülasyon zamanı güncelleştirildiğinde, gelecek için planlanmış olaylar, gelecek olaylar mevcut olaylar listesine aktarılır. MOL taranarak elemana herhangi bir işlem uygulanıp

uygulanmayacağı belirlenir. Eğer elemana uygulanacak bir işlem var ise gereken yapılır. Elemana artık bir işlem uygulanacaksa, MOL'dan yeni bir giriş yapılarak ilerletilir. Eğer MOL'de girişi yapılacak yeni bir eleman yoksa simülasyon zamanı güncelleştirilir ve gelecek olaylar listesi tekrar taranır. Simülasyon dilleri yukarıda açıklanan Ortam Yaklaşımlarına göre sınıflandırılabilir.

SLAM ve SIMAN gibi bazı diller bir kaç sınıfa dâhil edilebilir.

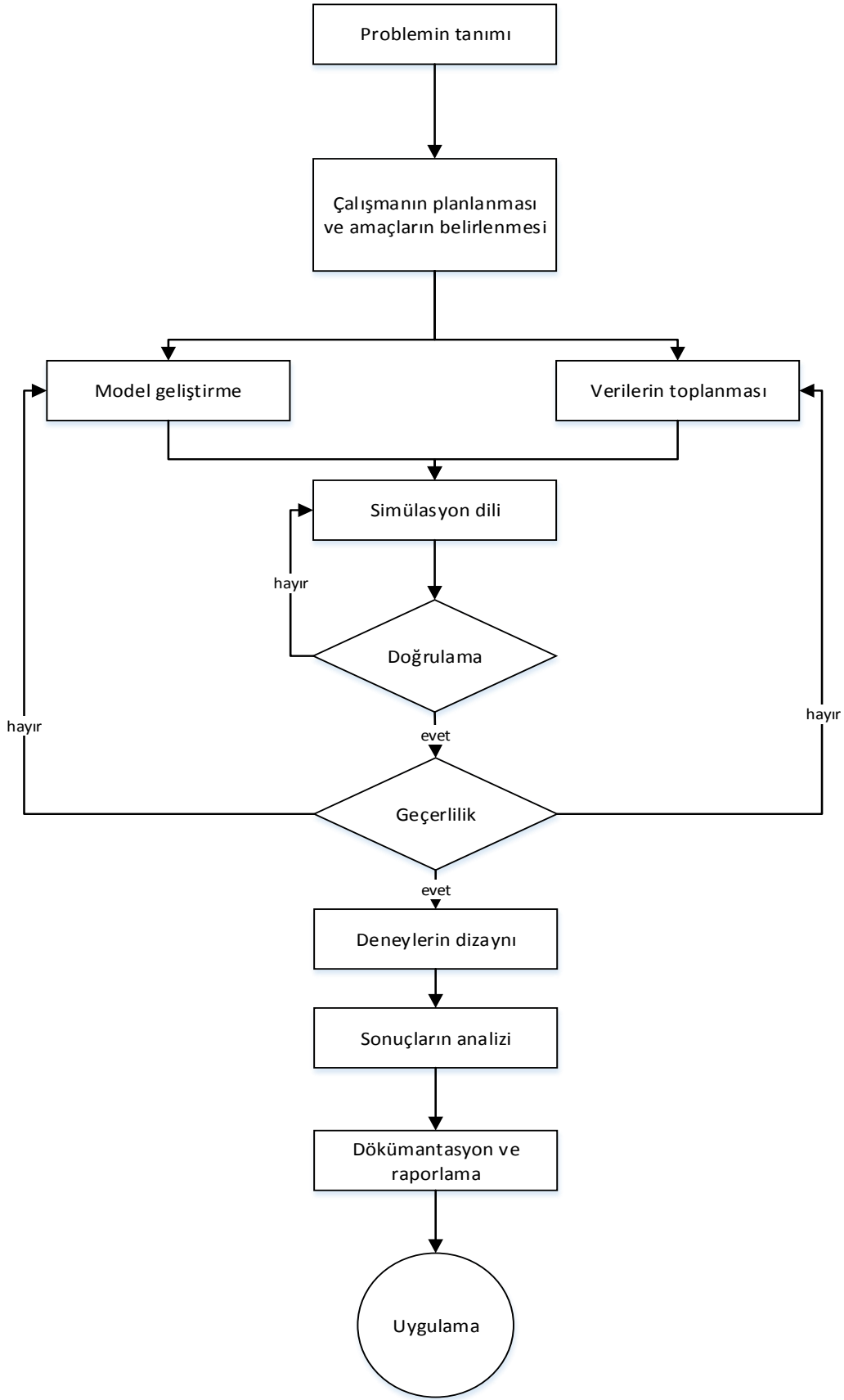
2.3.5.4 Sürekli (Continuous) Simülasyon Yöntemi

Bazı sistemlerde olaylar tüm zaman sürecinde sürekli değişebilir ve bu değişme bazı kesikli olayların aynı zamanlarında olmayabilir. Örneğin bir su deposundaki su seviyesi, giren ve çıkan akışkan debisine bağlı olarak bütün zaman içinde değişebilir. Böyle bir olay için sürekli benzetim yapmak daha uygundur. Buna rağmen kesikli benzetim yöntemi de burada bir yaklaşım yöntemi olarak kullanılabilir.

Sürekli simülasyon dili, 1950'li yılların sonunda analog bilgisayarların simülatörü olarak geliştirilmiştir. Analog bilgisayarlarda simülasyon, davranışları araştırılan sistem olarak, aynı matematik model tarafından tanımlanan (diferansiyel denklemler takımı) analog elektronik sistemin oluşturulması esasına dayanır. Elektronik sistem genelde integratörler, toplayıcılar ve diğer fonksiyonel birimler olarak davranmak için değiştirilmiş işlemsel yükselticilerine dayanan standart blokların iç bağlantısıyla oluşturulur. Kullanıcı sonra belirli çıkış noktalarında (osiloskop, çizici) uygun voltaj girişlerinin uygulanması ve voltajların kaydedilmesiyle ve bu elektronik sistem yardımıyla deneyleri yapar. Voltajın değişmesi, fiziksel davranışları tümünden farklı olabilecek (mekanik yer değiştirme, sıcaklık, vs.) orijinal sistem içerisindeki değişimleri tanımlayan fonksiyonla aynı olan zaman fonksiyonunu gösterir. Analog bilgisayarların ana problemi, multiplikasyon, bazı fonksiyonların türetilmesi, gecikmelerin veya diğerlerin türetilmesi gibi belirli işlemlerin analog olarak yürütülmesidir. Dijital bilgisayarlar bütün bu fonksiyonları çok basit olarak yapar ve bugün sürekli benzetim sadece dijital bilgisayarlar üzerinde yapılır. Analog bilgisayarların iyi olduğu tek alan integrasyondur. Dijital bilgisayarlar sayısal integrasyon için kullanır. Analog integratör kullanılarak yapılan integrasyonla kıyaslandığında, dijital bilgisayarlar genelde daha yavaş ve daha az hassastır.

2.4 Simülasyon Çalışmasındaki Adımlar

Aşağıda her simülasyon çalışmasında izlenmesi gereken adımlar yer almaktadır (Şekil 2.11).



Şekil 2.11 Simülasyon çalışmasındaki adımlar (Shannon 1975, Gordon 1978, Law and Kelton 2000)

2.4.1 Problemi Tanımlama ve Sistemi Anlama

Bir çalışma, mevcut bir ihtiyacı karşılayacak ya da bir problemi çözecek şekilde hazırlanmamışsa, detaylı ve eksiksiz olması bir anlam ifade etmez. Etkili bir çalışma yapabilmek için, potansiyel problemleri olan sistem parçalarının incelenmesi ve çalışmanın buna göre hazırlanması gerekir. İyi bir model, gelecek ihtiyaçları da göz önüne alarak, sistemin diğer parçalarını da kolayca içine alacak şekilde tasarlanmış olmalıdır. Fakat içinde fazlalık ve gereksiz bilgilerin bulunduğu bir model bilgisayar üzerinde diğer modellere göre daha yavaş çalışabilir ve maliyeti daha yüksek olabilir.

Her simülasyon çalışması bir problemin belirlenmesi ile başlamaktadır. Eğer problem açık bir şekilde ifade edilmemiş ise, bu durumda analizci problemin açık bir şekilde anlaşılması için daha fazla önem göstermelidir. Eğer problem, analizci tarafından formüle edilmiş ise, müşterinin gerçekleştirilen formülü kabul etmesi ve anlaması önemlidir. Bütün bu uyarılara rağmen, simülasyon çalışmasının yürütülmesi esnasında problemin formülasyonunda değişiklikler söz konusu olabilir. Simülasyon çalışmalarında ilk adım, analizci ya da bir grup analizcinin problemi belirlemesini ve bu probleme çözüm bulmasını içerir. Analize yeterli bilgi toplayarak, problem ve çalışılacak sistemin anlaşılması için gereken bilgiler sağlanarak başlanır.

Üzerinde çalışılacak problemin kesin ve öz bir tanımının yapılabilmesi, beklenenden zor olabilir. Çalışma sonucuyla ilgilenen tek kişi, genellikle sadece modeli oluşturan kişi değildir. Mühendisler, yöneticiler, operatörler ve birçok çalışanın, oluşturulan model ve yapılan 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. Tipik bir proje, analizcinin problemi tanımlaması ile başlamaktadır. Problem, analizcinin alt yapı ve deneyimine bağlı olarak, kar-zarar, beklentiler, darboğazlar ya da diğer ilgili işletme bilgileri terimleriyle ifade edilmektedir. Analizcinin problem tanımı belirtiler dâhilinde incelenerek teşhis konulmaktadır. Bu nedenle, belirtiler teşhis edilerek, problem tanımlanır ve son olarak model formüle edilir.

2.4.2 Çalışmanın planlanması ve amaçların belirlenmesi

Amaçlar, simülasyon çalışmasında cevaplanması gereken soruları temsil etmektedir. Proje planı, incelenecek çeşitli senaryoları içermelidir. Amaçlar, genellikle ele alınan problem tarafından belirlenir, çünkü model kurulduktan sonra, model üzerinde yapılacak çalışmaların problemi çözmesi hedeflenmektedir. Potansiyel sistem iyileştirme metodlarının değerlendirilmesinin, çalışma hedeflerinin belirlenmesinde rolü büyüktür, ancak bu metodlar, simülasyon çalışmasında ortaya çıkabilecek yeni alternatif metodları önleyecek şekilde dar olarak tanımlanmamalıdır. Bunlara ek olarak çalışma planında kullanılacak zaman terimleri, personel, donanım ve yazılım ihtiyaçları belirgin bir şekilde ifade edilmelidir.

2.4.3 Modelin oluşturulması ve simülasyon dili seçimi

2.4.3.1 Model geliştirme

Modelleme, genellikle sistemin soyut bir ortamının oluşturulmasıyla başlar ve gittikçe daha detaylı bilgilerin eklenmesiyle devam eder. Bu soyut model, sistemin mantıksal bir modelidir

ve sistemdeki olaylar arasındaki ilişkileri tanımlar. Bilgisayar üzerinde kurulan bu modelin gerçekleştirilebilmesi için, modeli kuran kişinin gerçek sistemin yapısını soyut olarak düşünebilmesi gerekir. Verilerin toplanması modelin kurulması esnasında da yapılabilir. Modellemenin özü, soyutlama ve basitleştirmedir. Öncelikle çalışmanın amaçlarına hizmet eden yeterli derecedeki sistem özellikleri tanımlanmaya çalışılır. Daha sonra, modelin yapısını oluşturacak amaç ya da hedef belirlenir ve ardından uygun bileşenler saptanmaya başlanır. Bu işlem, sistem verimliliğine ya da verimsizliğine etkisi olan bileşenlerin maddelenmesini gerektirir. Tam bir liste elde ettikten sonra, modelde her bir bileşenin olup olmayacağına karar verilir; fakat bu aşamada ana amaç açısından bileşenlerin önemi açık bir şekilde anlaşılmadığından, karar verme oldukça zor bir iştir. Cevaplanması gereken anahtar sorulardan biri de, modele girdi olarak sunulan özel bir bileşeni modelin parçası olarak mı, yoksa dış çevrenin bir bölümü olarak mı düşünüleceğidir (Morris 1967).

İlgilenilen sistemin uygun elemanlarında girdi bilgisinin eldesi, başarılı girdi modellemede ilk basamaktır. Öyle bir model tasarlanmalıdır ki, hem gerçek sistemi yansıtsın hem de çok fazla ayrıntı içermeyerek fazla hantal ve pahalı olmasın. En önemli tehlike, model çok fazla ayrıntılı olduğunda ve problemi anlamada az ya da hiç faydası olmayan elementleri içerdiğinde gerçekleşmektedir. Deneyimsiz kişiler, gerçek sistemdeki ayrıntılı zorlukları modele aktarmaya çalışırlar ve bilgisayardan bir şekilde problemi çözmesini umarlar. Modele basit bir şekilde başlanması, daha sonra projenin ilerlemesi ile birlikte modelin de gerçek sistemin karmaşıklığını yansıtmaya amacıyla kompleks bir forma dönüştürülmesinde yarar vardır. Her zaman gerçek sistemin simüle edilmesinden çok, ilgili sorulara cevap alabilmek için model tasarlanmaktadır. Pareto kuralına göre, varlıkların her grup ya da topluluğunda, azı önemli çoğu önemsiz bileşenler bulunur. Gerçekte, sistem davranışının %80'i, sistemi oluşturan bileşenlerin %20'sinin hareketiyle açıklanabilmektedir. Simülasyon model tasarımında problem, bu az olan önemli bileşenlerin belirlenip, modele dâhil edilmesidir.

Simülasyon hedeflerinin ve problemin belirlenmesinden sonra, modeli kuracak olan kişi modelin kavramsal iskeletini oluşturabilir. Bu iskelet, modelde ele alınacak ana olayları ve elemanları içerir. Ele alınacak sistemin bir taslağının ya da yerleşim düzenine ait bir çizimin kullanılması, çalışmaya çeşitli faydalar sağlar. İlk olarak, bu çizimler modeli kurmak için gerekli tüm detayların belirlenmesi ve çalışma sırasında sürekli hatırlanmasını sağlar. İkinci olarak, grafiksel gösterimler, her bir kaynakta ilgili verilerin sistematik olarak toplanmasını sağlamak için kullanılabilir. Ayrıca, sistemi anlamayı kolaylaştırmak için, sistemdeki akışlar ve etkileşimler (flows and interactions) bu çizimler üzerinde gösterilebilir. Sistemdeki operatörlerin, malzeme taşıyıcıların izledikleri güzergâh ve kullandıkları yollar da bu çizimler üzerine aktarılabilir. Toplanan verilerin doğruluğunun, elde edilen sonuç üzerinde etkisi büyüktür.

Yapılan ilk plan içerisinde; gerekli olan verilerin, bilgi kaynaklarının ve bu bilgilerin nasıl elde edileceği belirlenmelidir. İlk olarak, çalışma hedefleriyle ilgili olan bu bilgilerin çıkartılması gerekir. Tecrübeli bir model kurucu, çalışmada yer alan diğer kişilere hangi verilerin gerekli hangilerinin gereksiz olduğu konusunda yardım etmelidir. Son kullanıcılar ile model kurucu arasındaki sürekli iletişimin model kurma aşamasındaki önemi göz ardı edilmemelidir. Detaylarla ilgili ortak çalışma, projenin amaçlarından sapmasını önleyeceği

gibi, önerilen değişikliklerin güvenilirlik temellerini oluşturur (Gürkan 2004). Modele güvenirliliğin sağlanmasında iki önemli aşama doğrulama ve değerlendirmedir. Sistemin bire bir kopyasını çıkarmaya yönelik harcanan çaba genellikle gereksizdir. Detayların, gerekli olduğu zaman eklenmesi, çalışmanın hedeflerine ulaşması açısından takip edilmesi gereken en uygun yoldur. Teknik karmaşıklıklar, model ile modelin kurulma amacı arasındaki ilişkiden daha az öneme sahiptir (Kuş 2003).

2.4.3.2 Veri toplanması

Organizasyonlar, sistem içerisindeki bazı operasyonları için (makine arıza sıklıkları, belirli süreçler için işlem süreleri gibi) detaylı bilgiye sahipken bazı işlemler için kabataslak bilgiye sahip olabilirler (Shannon 1975). Yetersiz veya eksik veri bulunması durumunda modeli kuracak olan kişinin yapabileceği üç şey vardır:

- Sisteme en hâkim kişilerden yardım alabilir,
- Verileri kendisi toplayabilir,
- Verilerle ilgili tahminler yapabilir.

Modelde tahmini verilerin kullanılması durumunda, daha sonra yapılacak “duyarlılık analizi”nde bu verilerin sistem üzerindeki etkilerini anlamak için, değişik değerler kullanılmalı ve verilerin uç değerleri, toleransları çok iyi analiz edilmelidir. Bu tür bir çalışma, daha detaylı verilerin toplanmasının daha uygun olacağını gösterebilir.

Önce makro veri olarak adlandırılan, sistemle ilgili temel bilgilerin ve istatistiklerin toplanması gerekir. Bu makro verilerin amacı, modelin giriş parametreleriyle ve daha sonraki çalışmalarda kullanılacak olan verilerin toplanmasıyla ilgili detayları içeren parametrelerin temelini oluşturmaktadır. Bu durum, modeli kuran kişinin, projenin daha ileriki aşamalarında kullanılacak olan detaylı bilgileri daha kolay bulmasını sağlayacaktır.

Veri toplanması sürekli olarak yapılması gereken bir işlemdir. Simülasyon çalışması ilerledikçe ve makro veriler modele girildikçe, mikro verilerin toplanması önem kazanır. Birçok durumda, model kurucu, proje sırasında daha doğru ve güncellenmiş veriye ulaşabilir. Bir simülasyon modeline, yeni ve güncellenmiş verilerin kolaylıkla girilebilmesi bir avantajdır. Bu yüzden, çoğu model kurucu, daha kesin ve sağlıklı verilerin girilebilmesine imkân tanıyan daha esnek modeller kurmayı tercih ederler.

2.4.3.3 Simülasyon dili seçimi

Simülasyon işlemi gerçekleştirilirken dil seçimi konusunda iki alternatif bulunmaktadır. Bunlardan ilki FORTRAN, C, BASIC gibi algoritmik dilleri kullanarak simülasyon işlemi gerçekleştirmektir. İkincisi ise hazır olarak yazılmış simülasyon dillerinden birini kullanmaktır. Hazır paket programlardan simülasyon işleminin gerçekleştirilmesi ise genel uygulamalı sistemlerin ya da özel uygulamalı sistemlerin kullanımı ile gerçekleştirilir. Bütün genel uygulamalı sistemler Fortran temelli simülasyon dilleridir. Bu tip simülasyon dilleri her türlü modellerin geliştirilmesinde kullanılabilirler. Bu tiplere örnek olarak; IBM tarafından geliştirilen GPSS (General Purpose Simulation System), P-E Incubacon tarafından

geliştirilen HOCUS (Hand or Computer Universal System), System Modeling Co. tarafından geliştirilen SIMAN/ARENA verilebilir. Özel uygulamalı sistemler ise modelleme ve programlama zamanını azaltmaktadırlar. Bunlara örnek olarak; CMS Research tarafından geliştirilen MAST (Manufacturing System Design Tool), SIMFACTORY, WITNESS II verilebilir.

Aşağıda algoritmik bir dil yerine simülasyon dilinde simülasyon modeli programlamanın avantajları sıralanmaktadır:

- Kavramsal kılavuzluk,
- Model değişiminde esneklik fazladır,
- Daha az programlama hatası,
- İstatistiksel sonuçların otomatik eldesi.

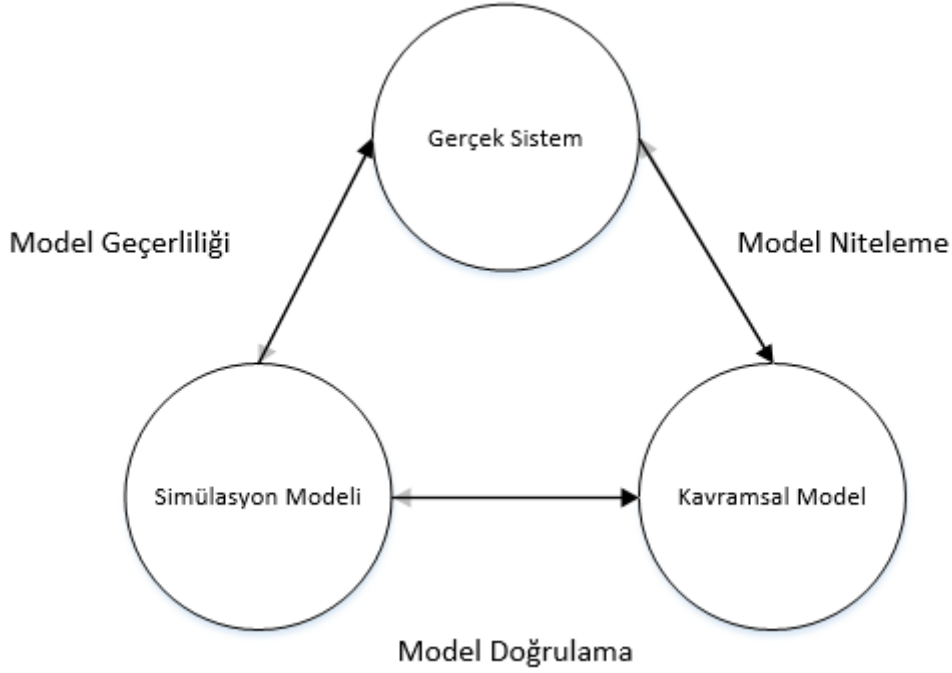
Bütün programlama dilleri, kullanıcıların model kavramı ve modeldeki ilişkileri nasıl ifade edebilecekleri arasındaki boşluğu doldurmaya çalışmaktadır.

Her ne kadar simülasyon dillerini kullanmanın avantajları olsa da algoritmik dillerinde avantajları bulunmaktadır:

- Simülasyon dilleri ile oluşturulan simülasyon modeli ancak belirli tip sistemlere cevap vermektedir.
- Eğer model ile ilgili olarak gelecekte herhangi bir değişim ya da yenilik yapılması düşünülüyorsa bu durumda algoritmik dili tercih etmek avantajlı olacaktır.
- Şirket için simülasyon dilini kullanmak pahalıya mal olacaktır.
- Algoritmik dillerin her bilgisayarda bulunması söz konusu iken, simülasyon dillerinin bulunma imkanı ancak kullanıcı isterse olacaktır.

2.4.4 Niteleme, Doğrulama ve Geçerlilik

Model niteleme, gerçek sistemle kabul edilebilir uygunluk seviyesini sağlamak için gerekli olan model bilgisini, elementlerini ve yapısını belirleme yöntemidir. Model doğrulama ise, model uygulamasının tam anlamıyla modeli geliştiren kimsenin kavramlarını ve tariflerini temsil edip-etmediğine karar verme işlemidir; geçerlilik, modelin, amaçlanan kullanımlar doğrultusunda gerçek sistemi ne dereceye kadar temsil ettiğine karar verme işlemidir (Hu, San and Wang 2001). Şekil 2.12’de simülasyon model gelişim işlemi verilmiştir:



Şekil 2.12 Simülasyon model gelişim işlemi

Model niteleme basamağı, bilgi toplama, bilgi açıklama ve parametre hesaplama, sistem yapısı ve analizi faaliyetlerini içermektedir. Modelin dayandırılacağı yeterli bilginin eksikliği durumunda model yapılandırma ilk basamak kapsamlı bilgi toplanmasıdır. Model yapılandırma bilgi toplama işlemi emek-yoğun olup, maliyetli bir işlemdir. Model yapılandırmanın ilk basamağı olan model nitelendirme basamağı son modelin kalitesini garantiye büyük rol oynamaktadır. Modele girilecek bilgiler gerçeği yansıtmalıdır.

Model doğrulama, yapılmış olan bilgisayar programının beklenen ve istenilen doğrultuda çalıştığını araştırır, bu sayede modelin doğru mantıksal temsili elde edilmiş olur (Kuş 2003).

Doğrulama işlemi, model ve model ihtiyaçlarının önemli ölçüde eksiklerini gidermektedir. Buna ek olarak, model gelişimindeki hataların giderilmesinde, doğrulama yapay-kodların aynı zamanda kavramsal modelin içeriğinde bulunan tanımları yansıtmayı yansıtmadığını belirler. Doğrulamanın amaçlarından biri, modelin her bir bölümünün birbirinden bağımsız ve beraber olarak çalıştığını, aynı zamanda doğru bilgileri doğru zamanda kullandığını göstermektir. Modelin geçerlilik işlemi, sistem hakkında türetilen sonuçların doğru olduğunu göstererek, kullanıcı güvenini kabul edilebilir bir seviyeye yükseltmektedir. Geçerlilik üç ana soru ile ilgilenmektedir:

- Model, gerçek sistemi yeterli seviyede yansıtabiliyor mu?
- Modelin davranış bilgi özellikleri, gerçek sistemin davranış bilgi özellikleri ile benzerlik gösteriyor mu?
- Simülasyon model kullanıcısı modelden elde ettiği sonuçlara güveniyor mu?

Sonuç olarak, testler üç gruba ayrılmaktadır: Model yapısının testleri, model davranışının testleri, modelin stratejik anlamlarının testleri.

Model özel bir amaç için yapılandırıldığı için, modelin yeterliliği ve geçerliliği sadece amaç dâhilinde değerlendirilebilir. Çalışılan sistem ya da sürecin problemlerini ve davranış özelliklerini yansıtan modeli oluşturmaya çalışılır. Geçerlilik işlemi modelin gelişimi ile ortaya çıkar ve çalışmanın başlamasıyla başlar, modelin makul bir şekilde çalışarak gerçek sistem hakkında doğru bilgi ve sonuçlar ürettiği görüldükçe devam eder.

2.4.5 Simülasyon Deneylerinin Dizaynı

Deney tasarımı simülasyon çalışmasının iki farklı basamağında gerçekleştirilir. Mümkün olduğunca erken yapılan deney tasarımında, hangi faktörlerin değişkenlik göstereceği, bu faktörlerin kaç farklı seviyesinin araştırılacağı ve verimi etkileyecek ölçümlerin seçimi yapılır. Model tasarımının erken basamağında yapılan deney planının ayrıntılı fikrinin elde edilmesi, istenilen bilginin verimli eldesinde model planlanması açısından daha kolay olmaktadır.

Model geliştirilip, doğruluğu ve geçerliliği kabul edildikten sonra, deneylerin yapılması için son stratejiler ve taktik planları düşünülür. Proje kısıtlamaları zamanında güncellenmeli ve bu kısıtlamalar tasarım üzerine aktarılmalıdır. Her ne kadar dikkatli bir plan yapılmış ve çalışmanın başından beri olası olaylar hesaplanmış da olsa, hangi kaynakların kaldığının ve kaynakların ne derece kullanıldığının dikkatli bir şekilde kontrol edilmesi gerekmektedir. Bu noktada kalan kaynakların hesaba katılması için deney tasarımı tekrar gözden geçirilmeli, aynı zamanda tasarım, model yapılandırma, doğrulama ve geçerlilik işlemleri esnasında elde edilen bilgiler de hesaba katılmalıdır.

Bilgisayar simülasyon deney tasarımı, temelde bilgi elde etme amaçlı kullanılan plandır. Tasarım, deneysel kaynakların verimli kullanımını iki nedenle etkilemektedir:

- Deney tasarımı, sonuçlara uygulanabilecek istatistiksel analiz formunu geniş ölçüde belirlemektedir.
- Soruların cevaplanmasında deneyin başarısı (zaman ve kaynak açısından verimli kullanım) doğru tasarımın seçimiyle yakından ilgilidir. Simülasyon deneyleri, zaman ve işgücü açısından pahalı olup, bazı durumlarda bilgisayar zamanı açısından da pahalılık arz etmektedir. Simülasyon çalışmalarının en basit amacı, mümkün olan en az maliyetle sistem davranışı hakkında bilgi edinmektir. Bu nedenle modelin dikkatli plan ve tasarımının yanı sıra, kullanımı da göz önünde bulundurulmalıdır.

2.4.6 Sonuçların Analizi ve Dokümantasyon

Her model konfigürasyonunun sonuçlarının mutlaka iyi bir dokümantasyonu yapılmalıdır. Normal raporlara ek olarak, yapılacak dikkatli bir dokümantasyon, modeli kuran kişinin hangi alternatifin en iyi sonucu verdiğini belirlemesine ilave olarak, yeni alternatifler doğuracak eğilimlerin de kolaylıkla ortaya çıkmasını sağlayacaktır. Bazı durumlarda, algılanan

iyileşmenin istatistiksel önemini belirlemek için ek tekrarlara ihtiyaç duyulabilir (Musselman 1998).

Genel olarak modeli kuran kişi, modellenen alternatiflerin, kullanılan varsayımların ve elde edilen sonuçların bir listesini oluşturur. Simülasyon yazılımları genellikle sonuçları istatistiksel olarak tablo formatında oluşturur. Buna ek olarak animasyon ve çıktı grafikleri, simülasyon sonuçlarının sunulması açısından oldukça önemli yardımcılarıdır. Simülasyonu yapılan modelin görsel etkisi de göz önüne alınmalıdır.

Sunuş bitirildikten sonra yapılacak daha fazla analiz yoksa modelin ispatlanmış olan önerileri yerine getirilmeye hazırdır. Eğer simülasyon iyi bir şekilde dokümanite edilmişse öneriyi gerçekleştirecek olan ekibe önemli bir kaynak sağlanmış olur.

2.4.7 Çıktı Bilgisi Analizi

Simülasyon ile kalite, kullanım oranı, esneklik (çalışan işçi, makine sayısı vs.), çıktı gibi miktarlar tahmini olarak hesaplanabilmektedir:

1. Bireysel değerlerin zaman serileri; örneğin, meşgul makine sayısı, işlemdeki envanter sayısı veya her bir birim için işleme süresi,
2. Zaman seri değerlerinin dağılımlarını özetleyen histogramlar ya da farklı faaliyetlerden geçen varlıkların dağılımını gösterme,
3. Zaman seri değerlerinin istatistiksel özetleri, özellikle ortalamaları, minimum ve maksimum değerleri,
4. Çıktı gibi tek değerler.

2.4.8 Uygulama (Implementation)

Uygulama gerçekte, simülasyon projesi ile başlamaktadır. Büyük projelerde, önerilerin uygulanabilmesi, izlenen adımların uygunluğuna bağlıdır. Modeli kuran kişi ve diğer ilgili personel, simülasyon projesinin uygulanmasında rehberlik etmelidir.

Proje için bir bitiş zamanı belirlense de, iyi oluşturulmuş modeller proje bitiminden sonra rafa kaldırılmaz, genellikle sistemin başka parçalarını da içine alacak şekilde geliştirilirler ya da başka modellerle entegre edilip sürekli iyileştirme çalışmalarında kullanılırlar. Bu aşamada model kurucu model mantığı ve varsayımlara ilgili dokümantasyonu sürekli iyileştirme çalışmaları için saklamalı ve yapılan çalışmalara göre güncellenmelidir (Pritsker 1995).

2.5 Performans İstatistiklerinin Ölçümü

Performans ölçümünde sık kullanılan 4 yöntem vardır. Bunlar;

- Sistem zamanı,
- Kuyruk zamanı,
- Herhangi bir zamanda kuyruktaki ortalama varlık sayısı,
- Verimliliğidir.

2.5.1 Sistem Zamanı

Sistem zamanı, gözleme dayalı bir üretim ölçümüdür. Bu, varlığın sistem içerisinde geçirdiği zamanı içermektedir. Sistem zamanı, varlığın sisteme varmasıyla ve kuyruğa girmesiyle başlar ve varlığın iş zamanı tamamlanıp, sistemden çıkmasıyla sona erer.

Kullanıcılar için tüm varlıkların ortalama zamanı çok önemlidir.

T_i =Her varlık için sistem zamanı(varış zamanı-ayrılma zamanı)

n =Sisteme giren, proseslerde yer alan varlıkların sayısını göstermektedir.

$$\frac{\sum_{i=1}^n T_i}{n}$$

2.5.2 Kuyruk Zamanı

Kuyruk zamanı, gözleme dayanan bir ölçümdür. Sistem zamanıyla benzerdir; ancak kuyruk zamanında varlıkların sadece kuyruқта geçirdikleri zaman hesaplanmaktadır.

D_i =Her varlık için kuyruk zamanı(varış zamanı-ayrılma zamanı)

n =Kuyruğa giren varlıkların sayısıdır.

$$\frac{\sum_{i=1}^n D_i}{n}$$

2.5.3 Herhangi Bir Zamanda Kuyruktaki Ortalama Varlık Sayısı

Herhangi bir zamanda kuyruktaki ortalama varlık sayısı, zamana bağlı istatistiklerdir ve direkt olarak kuyruktaki varlıkların sayısının bir fonksiyonu değildir. Herhangi bir zamanda kuyruқта görülmesi beklenmeyen varlıkların ortalama sayısıdır. Kuyruktaki varlıkların sayısı farklı olacaktır. Herhangi bir zamanda kuyruktaki ortalama varlık sayısının ortalama değeri çok küçüktür. Az dolu kuyruklarda ortalama değer 1'den küçüktür.

Q = Kuyruқта verilen zaman aralığı için sayı değeri

dt = Q 'nun gözlemlendiği süre

T = Simülasyon için toplam süredir.

$$\frac{\int_0^T Q dt}{T}$$

2.5.4 Kaynak Verimliliği

Kaynak verimliliği zamana bağlı bir istatistiktir. Herhangi bir zamanda kaynak boş veya dolu olabilir. Boş durumda kaynak verimliliği “sıfır”a tekabül ederken, dolu durumda ise “bir”e

karşılık gelmektedir. Kaynağın “sıfır” ya da “bir” seviyesinde olduğu zaman aralığında varlıkların fonksiyonu sisteme girmektir (Chung 2004).

B = Boş için 0; dolu için 1

dt = B 'nin ölçüldüğü zaman uzunluğu

T = Simülasyonun toplam süresidir.

$$\frac{\int_0^T B dt}{T}$$

3. MATERYAL VE YÖNTEM

3.1 Simul8 Yazılımı

Çalışmanın tüm uygulamaları Simul8 programında gerçekleştirilmiştir. Simul8 programı Simul8 Corporation tarafından geliştirilmiş, kesikli olay simülasyonu programıdır. Bu yazılım sayesinde kullanıcılar inceledikleri sistemin görsel bir modelini kolayca oluşturabilmektedirler. Programda kullanılan tipik nesnelere iş parçaları, kuyruklar ve servis noktalarıdır. İş parçaları üretilen fiziksel parçalar olabildiği gibi operatörler tarafından hatta bekletilen telefon görüşmeleri gibi sanal parçalarda olabilir.

Programda sistem modeli kurulduktan sonra, iş parçalarının sistem içerisinde akışı animasyonla gösterilmekte, böylece sistem iyileştirmesi için yeni fırsatlar ortaya çıkmaktadır.

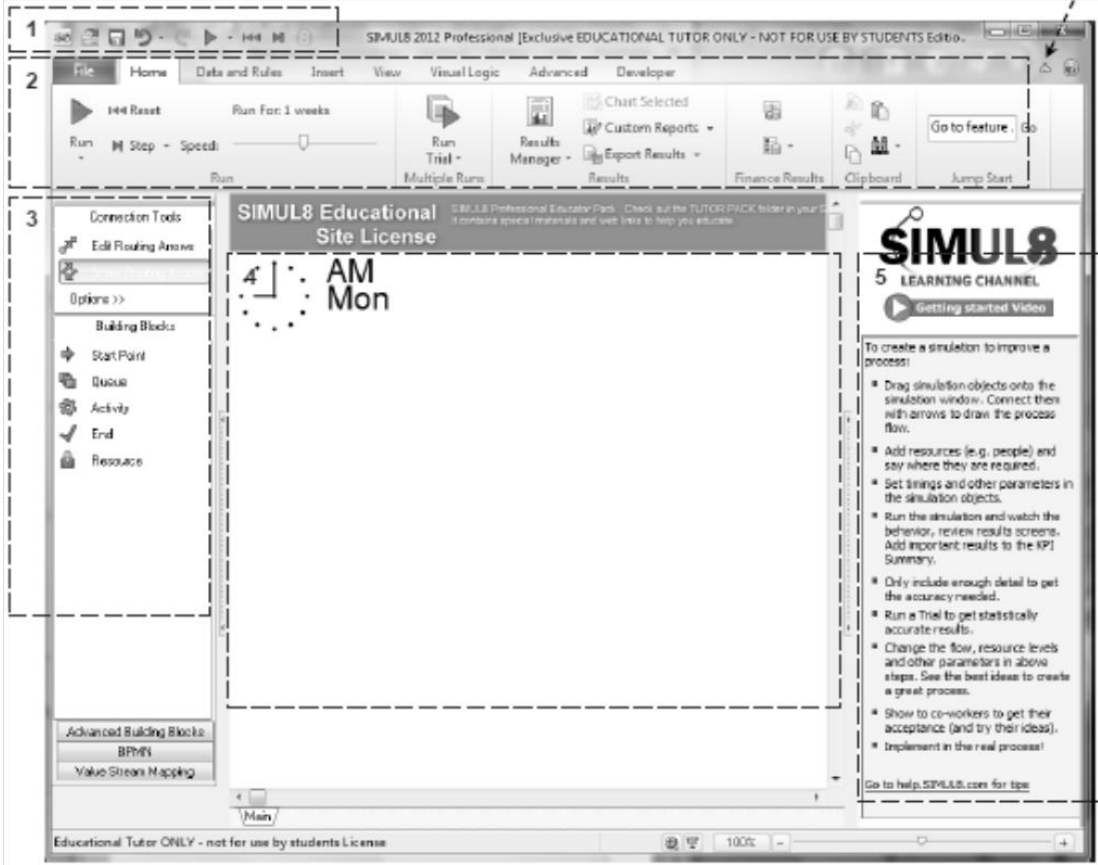
Simülasyon nesnelere, kapasite veya operatör hızı gibi terimlerle ifade edilebilmekte, model tekrar tekrar çalıştırılarak sistemin performansı istatistiksel olarak ifade edilebilmektedir. İstatistiksel ifadeler ortalama bekleme süreleri, iş merkezinin veya kaynağın verimliliğini göstermektedir.

Bunun yanı sıra Simul8 raporlamada çeşitli grafikler kullanmakta, böylece kullanıcının olup biteni daha iyi anlamasını sağlamaktadır.

3.2 Program Arayüzü

Simul8 yazılımı temelde 5 bölümden oluşmaktadır (Şekil 3.1);

1. Hızlı geçiş araçları
2. Menü şeridi
3. Simülasyon çizim ve bağlantı araçları
4. Simülasyon penceresi
5. Öğrenme paneli



Şekil 3.1 Simul8 arayüzü

3.3 Program Yapı Taşları

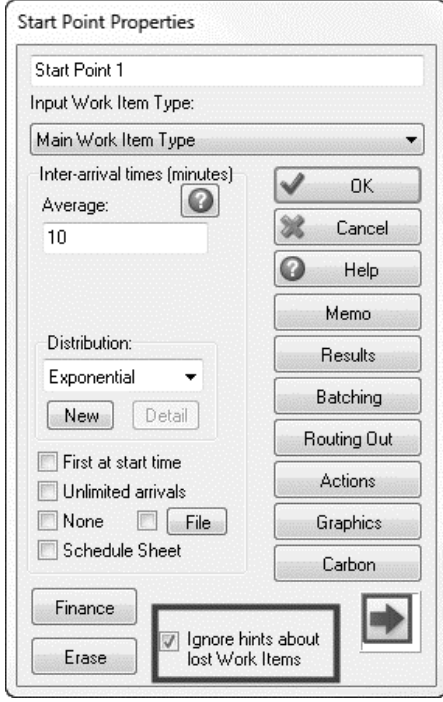
Simul8 yazılımının 5 tane temel yapı taşı vardır;

1. Başlangıç Noktası (Start Point)

Burası iş parçalarının[] sisteme girdiği noktadır.

Birden fazla ham materyale ihtiyaç varsa sistemin herhangi bir noktasından girmek üzere birden fazla iş giriş noktası olabilir.

Başlangıç noktası özellikleri Şekil 3.2'deki pencereden belirlenebilmektedir. Simülasyon analisti, parçaların sisteme giriş şablonunu bir çizelgeleme ya da istatistiksel dağılım ile kontrol edebilir. Ayrıca iş parçaları arası geliş sürelerini, geliş çizelgelemelerini, demet büyüklüklerini, çıkış noktalarını, etiket değişikliklerini, çeşitli görsel ayarları, enerji tüketimini, maliyetini, raporlama seçeneklerini değiştirmekte mümkündür.

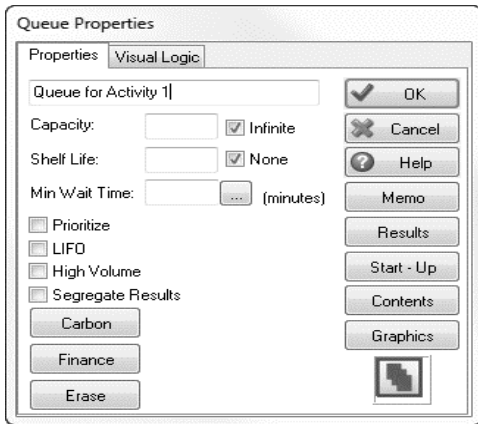


Şekil 3.2 Başlangıç noktası özellikleri penceresi

2.Kuyruk (Queue)

Burası iş parçalarının işlenmek üzere beklediği yerdir. Bunlar gerçek depolar olduğu gibi sanal kuyruklarda olabilir.

Kuyruk özellikleri Şekil 3.3'deki pencereden belirlenebilmektedir. Simülasyon analisti kuyrukların kapasitesini, kuyruktaki parçaların maksimum ve minimum bekleme süresini, başlangıçtaki kuyruk sayısını, kuyruk disiplini ve hangi sonuçların analizinin yapılması gerektiğini kontrol edebilir.

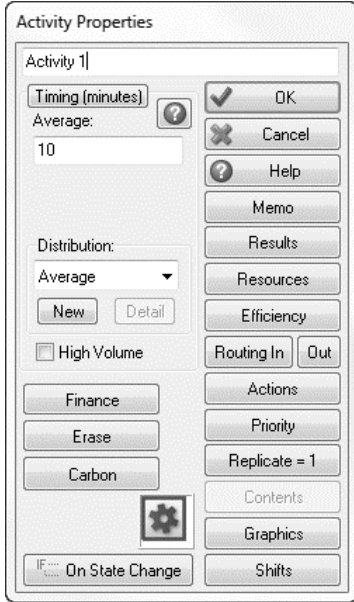


Şekil 3.3 Kuyruk özellikleri penceresi

3.Aktivite (Activity)

Burası işin bir makine ya da çalışan tarafından yapıldığı yerdir.

Aktivite özellikleri Şekil 3.4'deki pencereden belirlenebilmektedir. Simülasyon analisti, yapılacak işin süresinin hangi istatistiksel dağılım ile gerçekleştirileceğini, çalışılan ve boş zamanlardaki farklı sembollerini, işin hangi kaynak veya kaynaklarla gerçekleştirileceğini, aynı makineden kaç adet olacağını, maliyet değişkenlerini ve biten işin gideceği yeri kontrol edebilir.



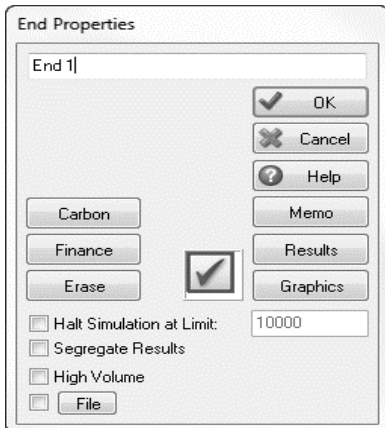
Şekil 3.4 Aktivite özellikleri penceresi

4. İş Çıkış Noktası (End Points) ✓

Burası işin sistemden çıktığı noktadır.

Birden fazla iş çıkış noktası olabilir. Mesela; bir üretim sonucunda ürün depoya ya da çöpe gidebilir veya bir arama merkezinde gelen aramalar “tamamlanmış” ya da “memnuniyetsiz müşteriler” olarak sistemden çıkabilir.

Çıkış noktası özellikleri Şekil 3.5'deki pencereden belirlenebilmektedir. Simülasyon analisti çeşitli görsel ayarları, enerji tüketimini, maliyetini, simülasyonu durdurma limitini, raporlama seçeneklerini kontrol edebilir.



Şekil 3.5 İş çıkış noktası özellikleri penceresi

5.Resource (Kaynak)

Bunlar, birkaç makinenin hazırlık işlemlerini yapan bir operatör örneğinde olduğu gibi süreçlerin kaynağa ihtiyaç duyduğu iş istasyonlarında gereklidirler.


Kaynak özellikleri Şekil 3.6'daki pencereden belirlenebilmektedir. Bir iş merkezine kaç kişinin bakacağı, vardiya dağılımı, bir işten diğerine gidiş süresi gibi çeşitli ayarlar bu pencereden yapılabilmektedir.

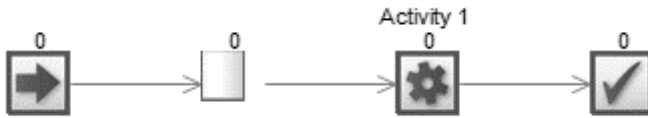


Şekil 3.6 Kaynak özellikleri penceresi

3.4 Program Prosedürleri

3.4.1 Model Kurulumu

Şekil 3.7'de görüldüğü gibi bir model oluşturmak için yapılacak ilk şey araç çubuklarından “Başlangıç Noktası”nı seçmek ve ekrana sürüklemektir. Daha sonra aynı şekilde “Kuyruk” ve “Aktivite” ekrana sürüklenir. Eğer şekiller birbirlerine otomatik bağlanmazlarsa, onları daha sonra oklar yardımıyla birbirine bağlamakta mümkündür. Model oluşuncaya kadar yukarıdaki işlemler tekrar edilir ve araç çubuklarındaki  ikona basılarak model çalıştırılır.



Şekil 3.7 Örnek model

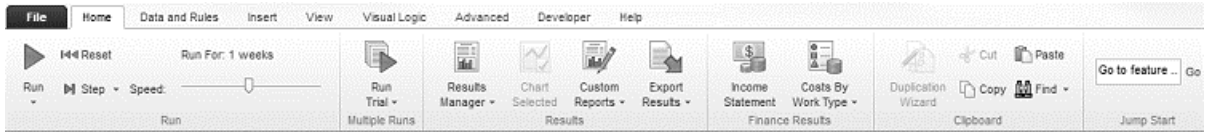
3.4.2 Menü Şeridi Açıklamaları

3.4.2.1 Home Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Modeli tek seferlik çalıştırmak,

- Modeli reset etmek,
- Modeli adım adım çalıştırmak,
- Sonuç toplama periyodunu belirlemek,
- Farklı karışık sayılarla modeli çoklu çalıştırmak,
- Elde edilecek sonuçları yönetmek,
- Sonuçların görsel gösterimini seçmek,
- Kişiyeye özel rapor yaratmak,
- Farklı formatlarda çıktı almak,
- Gelir gider raporları elde etmek,
- Kesmek, kopyalamak, yapıştırmak.



Şekil 3.8 Home sekmesi görünümü

3.4.2.2 Data & Rules Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Çalışma ve mola saatlerini ayarlamak,
- Çalışma ve tatil günlerini ayarlamak,
- Kaynaklar ve aktiviteler için vardiyaları ayarlamak,
- İş parçaları etiketlerini değiştirmek,
- Dağılımları belirlemek ve yeni dağılımlar yaratmak,
- Bilgi deposu yaratmak,
- Model parçalarını gruplandırmak ve grupları yönetmek,
- Kaynak matrisi oluşturmak,
- Girdi özeti ve iş matrisi oluşturmak,
- Simülasyon nesnelere arası bağlantıları değiştirmek,
- Sabit gider maaşları ayarlamak,
- Simülasyonu bir veri bankasına bağlamak,
- Excel bağlantıları oluşturmak.



Şekil 3.9 Data & Rules sekmesi görünümü

3.4.2.3 Insert Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Arka fon eklemek,
- Ekranı herhangi bir şekil çizmek,
- AutoCAD çizimini arkaya fon olarak eklemek,
- Herhangi bir şekli simülasyon penceresine eklemek,
- Simülasyon nesnelere için farklı katmanlar oluşturmak,
- Simülasyon penceresine bir metin eklemek,
- Modelin çalışması sırasında ekrana dinamik hatırlatmalar, tablolar, veri çubukları ve çizimler eklemek,
- İstenilen işlemi gerçekleştirmek üzere simülasyon ekranına aktif bir düğme atamak,
- Ekranı yeni bir menü eklemek



Şekil 3.10 Insert sekmesi görünümü

3.4.2.4 View Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Simülasyonu üç boyutlu göstermek,
- Simülasyon nesnelere durumunu göstermek,
- Görsel olarak doğrulama yapmak,
- Görsel olarak yoğun olan bağlantıları vurgulamak,
- Nesnelere arası mezüre ile ölçüm yapmak,
- Ekranı büyüteçle bakmak,
- Model için yapılan varsayımları yazmak,
- Simülasyonda kullanılan resimleri yönetmek,
- Alt süreçler oluşturmak ve yönetmek,
- Yatay ve dikey eksenler oluşturmak ve ayarlamak,
- Görüntüyü yakınlaştırmak, uzaklaştırmak.



Şekil 3.11 View sekmesi görünümü

3.4.2.5 Visual Logic Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Model çalışırken Visual Logic olaylarını zamana veya olayların oluş sırasına göre listelemek,
- Visual Logic çalışma hızını analiz etmek,
- İş parçalarının etiket değişimini izlemek,
- Visual Logic uyarılarını açmak veya kapamak,
- Kısacası Visul Logic kullanımı ile ilgili tüm ayarlamaları yapmak.



Şekil 3.12 Visual Logic sekmesi görünümü

3.4.2.6 Advanced Sekmesi

Bu sekme dâhilinde aşağıdaki işlemleri yapmak mümkündür;

- Yazılım veri toplamadan önce beklenmesi gereken ısınma süresini belirlemek,
- Aynı simülasyonu farklı senaryolarla çalıştırmak,
- Başarı kriterlerini gerçekleştirmek için en iyi girdileri optimize etmek,
- Stat Fit modülü ile verilerin oluşturduğu dağılımları hesaplamak,
- Duyarlılık analizi yapmak,
- İş parçası tiplerini yönetmek,
- Kullanılan karışık sayıları yönetmek,
- Görüntü birimi ile simülasyonda gerçekleşen aktiviteleri ayrıntılı olarak incelemek,
- Makro oluşturarak klavye ve mouse hareketlerini kaydetmek,
- İki farklı modeli birleştirmek,
- Farklı simülasyonları karşılaştırmak,
- Çizelgeleme yardımıyla simülasyonu geç başlatmak,



Şekil 3.13 Advanced sekmesi görünümü

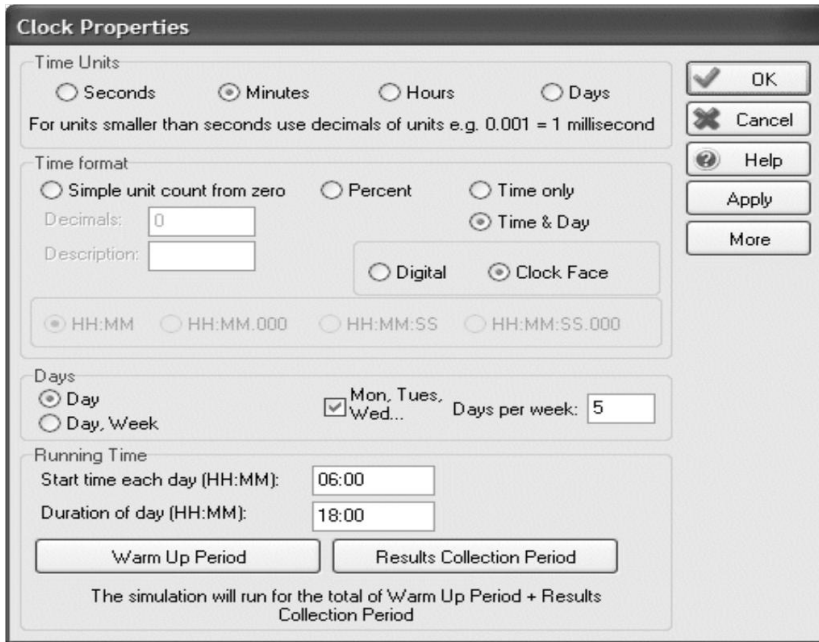
3.4.2.7 File Sekmesi

Bu sekme dâhilinde Yeni bir simülasyon oluşturmak, açmak, kaydetmek, çıktısını almak, yardım almak, seçenekleri ayarlamak gibi işlemleri yapmak mümkündür.

3.4.3 Çalışma saati

Çalışma saati Simul8'in kalbidir. Modeli çalıştırmadan önce kullanılmak istenen zaman birimine (dakika, saniye ve saat) ve simülasyonun ne kadar çalıştırılacağına karar vermek gerekmektedir.

Simul8 varsayılan çalışma saati sabah dokuzdan akşam beşe kadar haftalık 40 saattir. "Clock" menüsü içerisinde yer alan "Clock Properties" seçilerek saat ayarı yapmak mümkündür. Eğer başlama zamanı saat 06:00, bitiş zamanı saat 18:00 olarak seçilirse saat günlük 12 saat çalışacaktır (Şekil 3.14).

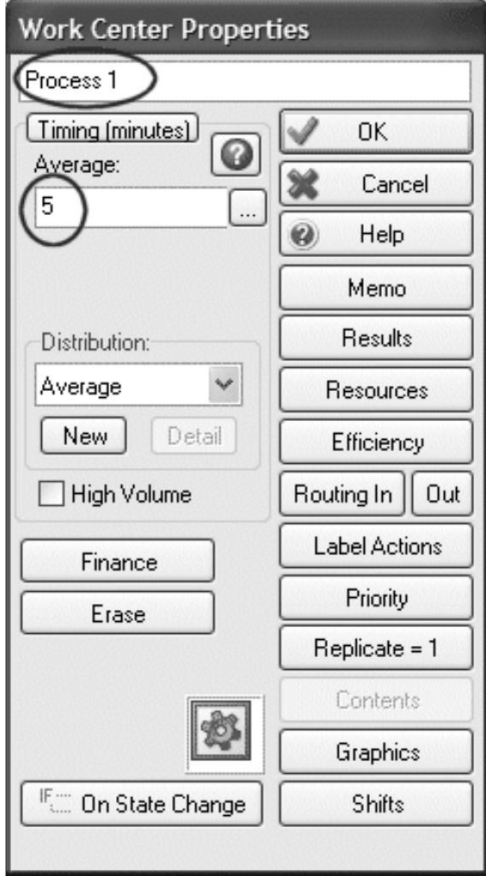


Şekil 3.14 Çalışma saati ayarı

3.4.4 İşlem ve Varış Süreleri

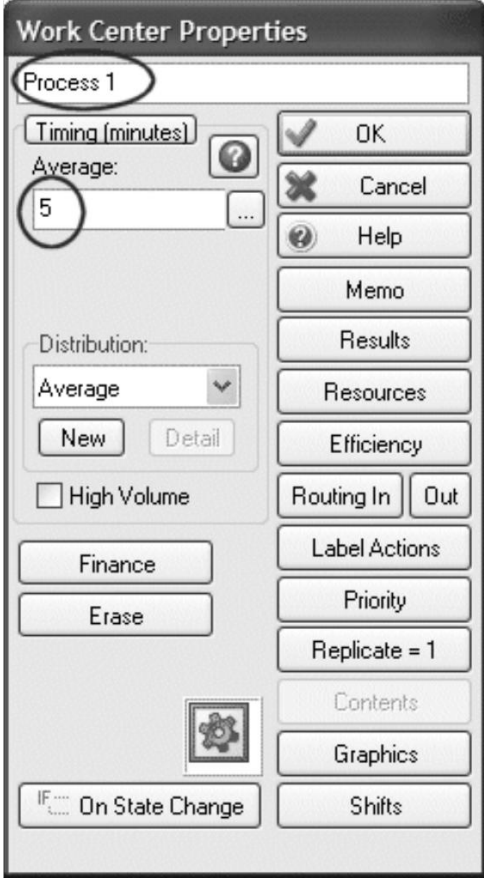
İşlem süresini değiştirmek için "Work Center"a tıklayarak istediğimiz değeri yazmak yeterlidir. "Work Center"a tıkladığımızda karşımıza "Work Center Properties" penceresi açılır. Simul8 programında varsayılan işlem süresi 10 dakikadır. "Average" kutusuna tıklayarak bunu kolayca 5'e çevirmek mümkündür (Şekil 3.15). Ayrıca iş merkezinin adı da

“Work Center Properties” yazısının altında yer alan kutuya tıklayarak değiştirilebilir. Bunun dışında parçaların işleme sürelerini ifade edilen dağılımlar değiştirilmek istenirse yapılması gereken şey “Distribution” yazının altında yer alan kutudan istediğimiz dağılım tipini seçmektir.




Şekil 3.15 İşlem süresi ayarı

Varış sürelerini değiştirmek için “Work Entry Point” ikonuna tıklamak gerekmektedir. Şekil 3.16’da görüldüğü üzere “Average” yazının altında yer alan 5 rakamı parçaların dakika cinsinden geliş aralıklarını ifade etmektedir. Diğer bir deyişle saatte 12 parça sistem girecektir. Burada yapılacak zamansal ve dağılımsal değişiklikleri sistemde oluşacak kuyrukları etkileyecektir.



Şekil 3.16 Varies süresi ayarı

3.4.5 Kaynaklar

Kaynaklar iş merkezlerinde işlerin yapılmasını sağlarlar. Simülasyona kaynakları eklemek için araç çubuklarında “Create Resources” düğmesine tıklamak gerekmektedir.  şeklinde görüntülenen ikon üzerinde yer alan 10 rakamının anlamı uygun olan kaynak sayısıdır.

Kaynak özellikleri “Work Center” a tıklanarak açılan “Work Center Properties” penceresinden “Resources” ın seçilmesiyle ayarlanır. Kaynak listesinden ihtiyaç duyulan kaynaklar eklenir. İhtiyaç duyulan kaynak sayısını değiştirmek için kaynağa tıklayarak “Resource Properties” penceresini açmak ve varsayılan değer olan 10’u değiştirmek yeterlidir (Şekil 3.17).

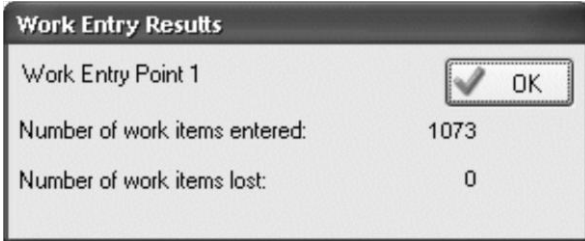


Şekil 3.17 Kaynak ayarları

3.4.6 Sonuçlar

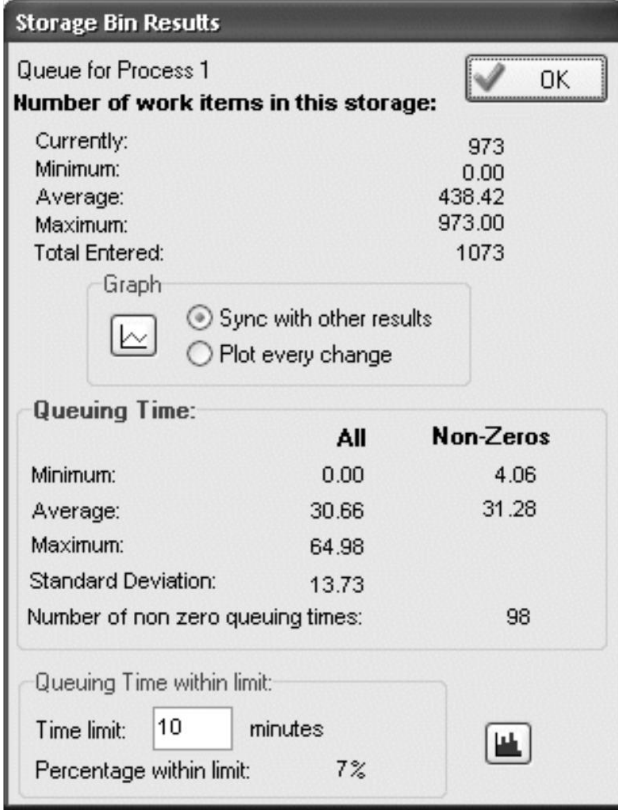
Ekrandan yaratılan ve yer alan tüm nesnelere sonuç almak mümkündür. Kısacası her iş giriş noktasının, depolama haznesinin, iş merkezinin ve iş çıkış noktasının kendine ait sonuç kümeleri vardır. Bunlar sonuçlar özetinde kolayca görülecek şekilde ayrıca verilmektedir.

İş giriş noktalarının sonucunu almak için “Work Entry Point”e tıklayarak, “Results” düğmesine basmak yeterlidir. Burası simülasyona kaç ürünün girdiğini veya kaçının kaybolduğunu gösterir (Şekil 3.18)



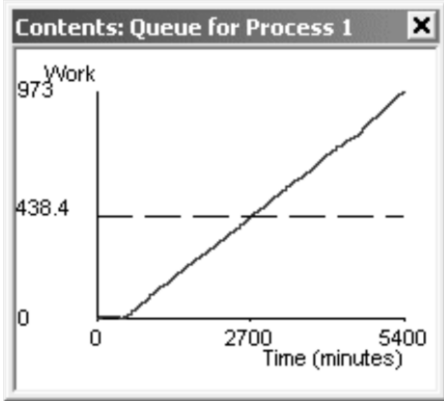
Şekil 3.18 İş giriş sonuçları

Depolama haznesi daha fazla sonuç ve ayrıca grafiksel çıktı vermektedir. Sonuçlar haznedeki kaç ürünün geçtiğini, simülasyon sonunda ne kadarının kaldığını, ortalama ürün sayısını vermektedir. Ayrıca bazı endüstri dalları için çok önemli olan kuyruk süresi de gösterilmektedir (Şekil 3.19).



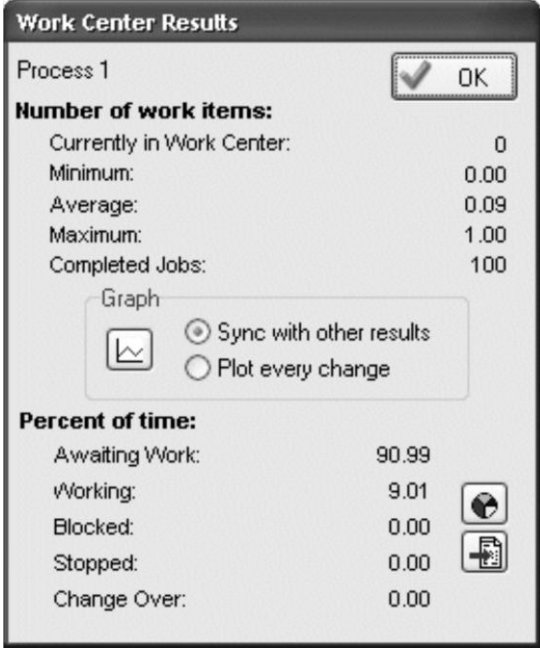
Şekil 3.19 Depolama haznesi sonuçları

Bunun dışında daha dinamik bir hikaye anlatan grafik çıktı almakta mümkündür. Şekil 3.20'de görüldüğü üzere stokta 100 ürün oluştuktan sonra kuyruk arttığı görülmektedir.



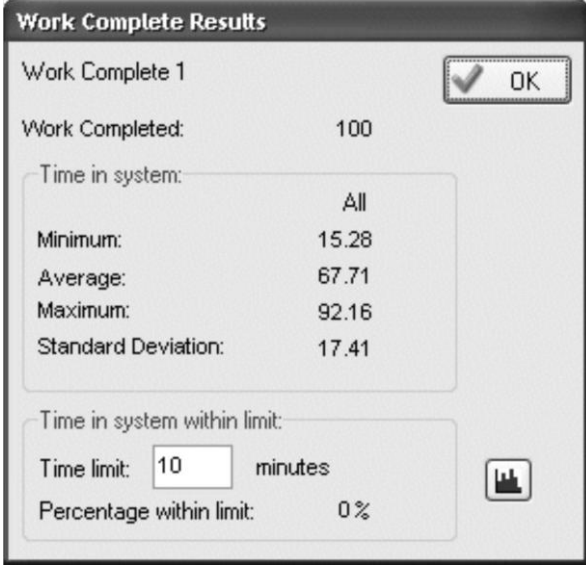
Şekil 3.20 Kuyruk grafiği örneği

İş merkezi sonuçları kaç ürünün işlendiğini ve zamanın yüzde kaçında iş merkezinin gerçekten çalıştığını, beklediğini, kilitlendiğini ve arıza yaptığını göstermektedir. Ayrıca grafiksel çıktı ve pasta grafiği almakta mümkündür (Şekil 3.21).



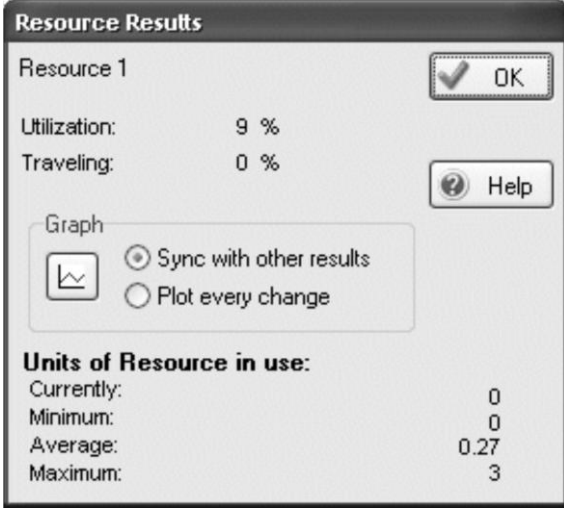
Şekil 3.21 İş merkezi sonuçları

İş çıktı noktası işlem gören ürün sayısını ve ürünün sistemde geçirdiği süreyi göstermektedir (Şekil 3.22).



Şekil 3.22 İş merkezi sonuçları

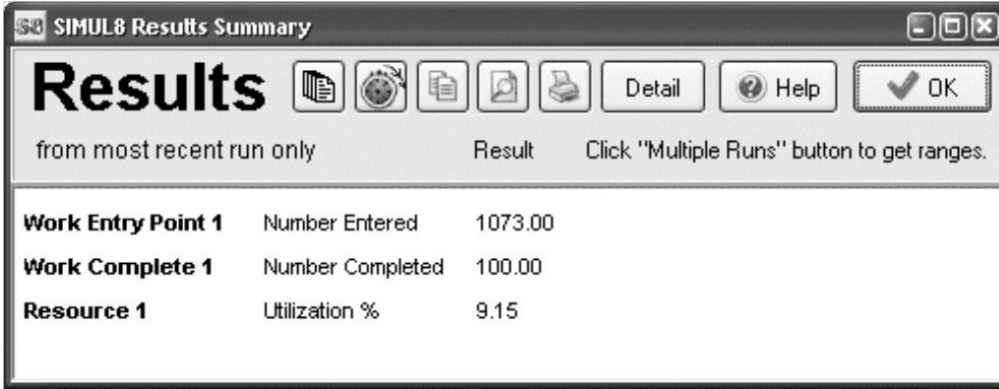
Kaynak sonuçları ise kısaca kaynağın verimliliğini göstermektedir. Bu oran çalışanın gerçekten çalışarak geçirdiği sürenin tüm çalışma süresine oranını vermektedir (Şekil 3.23).



Şekil 3.23 Kaynak sonuçları

Tüm bu sonuçlar, “sonuçlar özetine” eklenebilir. Bunun için “Results Summary” penceresini açıp sağ tuşla bilgi alınmak istenilen verileri eklemek mümkündür. Tek bir simülasyon çalışmasıyla elde edilen sonuçlar yanıltıcı olabileceği için eşzamanlı belirli sayıdaki tekrar eden çalışmalar yapmak gerekmektedir. Simul8 buna “Run a Trial” demektedir.

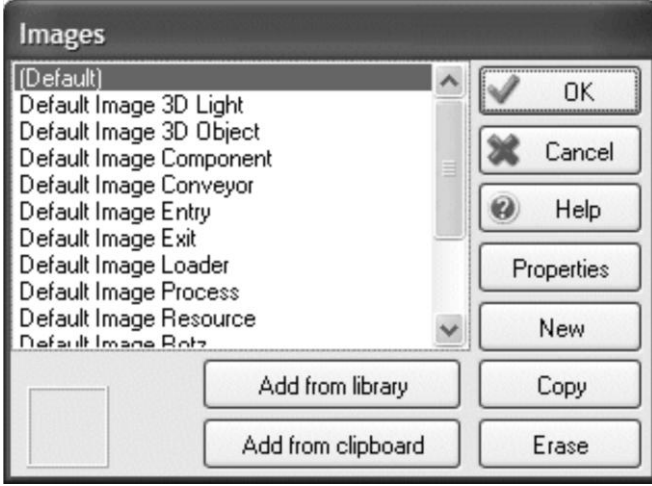
Sonuç özetleri otomatik olarak %95 alt ve üst güven aralıklarını vermektedir. Bu şekilde 100 sonucun 95’i tahminlenebilmektedir (Şekil 3.24).



Şekil 3.24 Sonuçlar özeti

3.4.7 Grafikler

Simülasyonda neler olup bittiğini anlamak ve görsel olarak anlatmak için grafikler hayati önem taşımaktadır. Simul8 programında herhangi bir nesne “bitmap” resmi olarak atanabilir. Program içinde 200’den fazla resim bulunmaktadır. İstenirse bunlara yenileri eklenebilir veya mevcut olanlar değiştirilebilir. Bir nesneye yeni bir resim atamak için, “Grafiks” düğmesi ve ardından “Images” seçilmelidir. “Add from library” düğmesine basılarak eklenmek istenen resim seçilir (Şekil 3.25).



Şekil 3.25 Grafik ayarları

3.5 Simülasyon Modelleri

Çalışmanın yürütüleceği (düz bant üretiminden yalın üretime geçen) firma ile görüşülmüş, çalışma planının ne şekilde yürütüleceği belirlenmiştir. Üretimi yapılan ve en çok tekrarı gelen basic t-shirt, elbise ve tayt olmak üzere 3 model belirlenmiştir. Özellikle sık tekrar edilen modellerin seçilme nedeni, simülasyon modeli oluşturulurken geçerlilik ve doğrulama aşamalarında, gerekli kontrolleri, tekrar edecek olan modellerde yeniden gözden geçirme olasılığının daha yüksek olmasıdır.

Çalışmanın gerçekleştirileceği modellerin numaraları şu şekildedir;

- Basic T-shirt (Model 1) (Şekil 3.26)
- Elbise (Model 2) (Şekil 3.27)
- Tayt (Model 3) (Şekil 3.28)



Şekil 3.26 Basic T-shirt (Model 1)

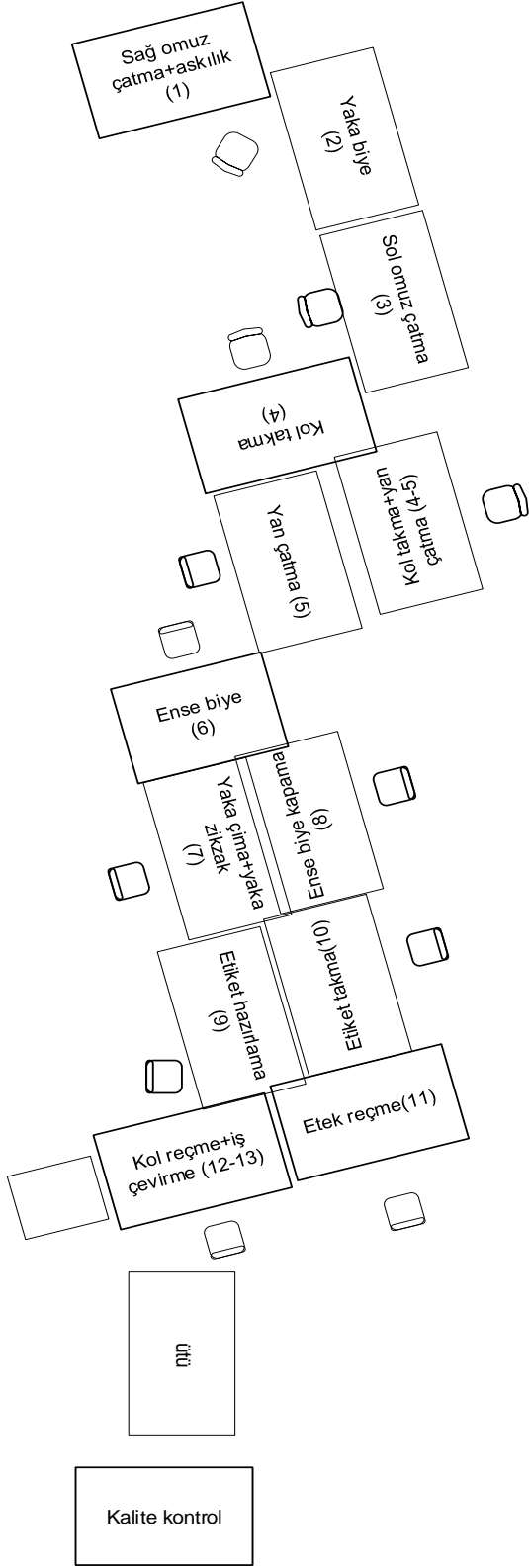


Şekil 3.27 Elbise (Model 2)

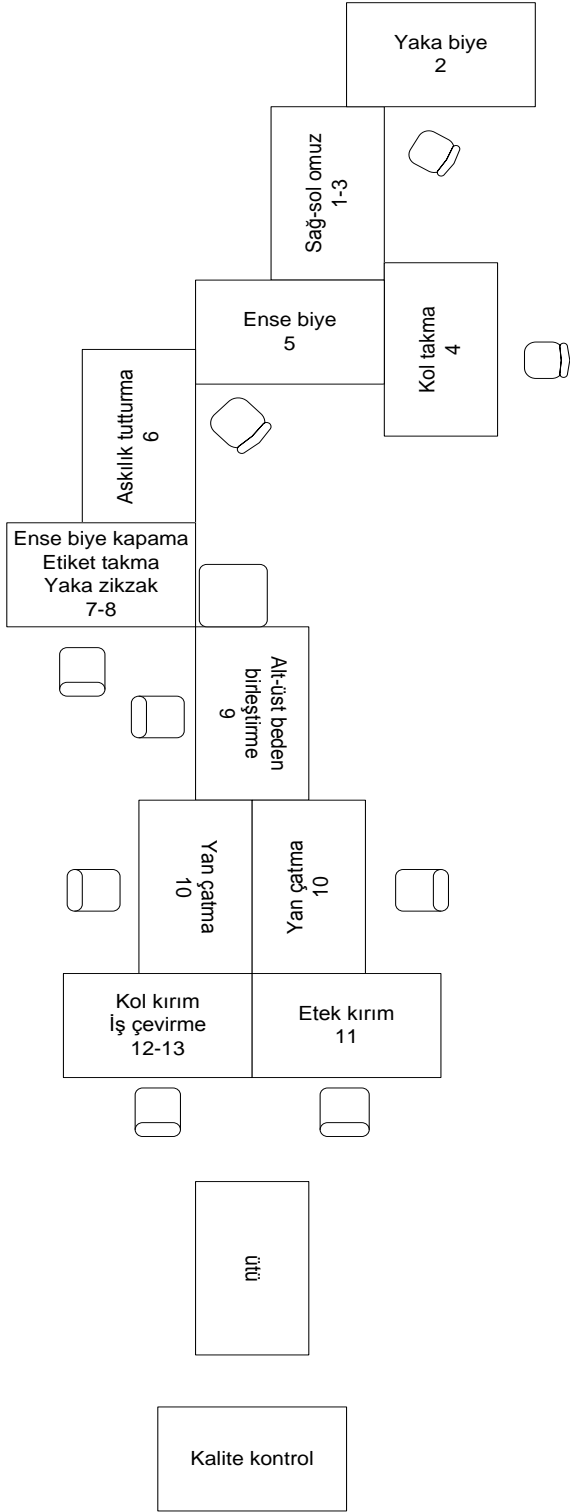


Şekil 3.28 Tayt (Model 3)

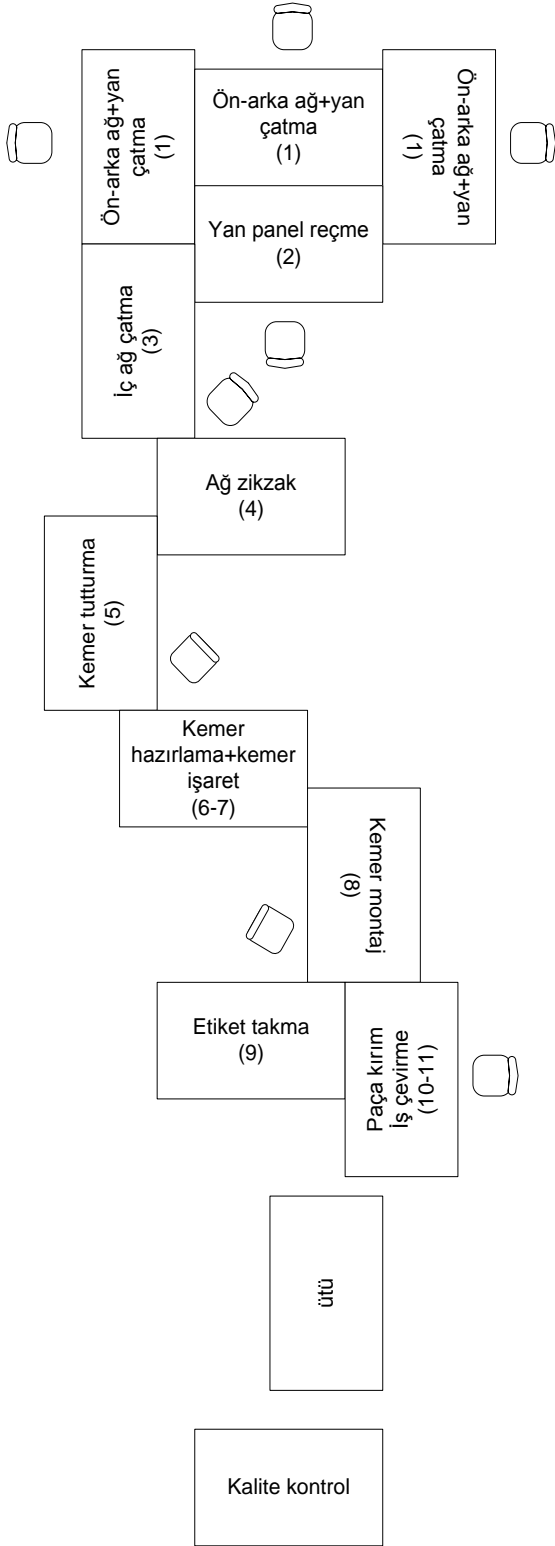
İlk olarak modeldeki aktiviteler arasında bağlantıları belirleyebilmek için üretim akışları ve bir operatörün mevcut sistemde kaç makineden sorumlu olduğunu belirlemek için yalın üretim bantlarının kuruluş biçimi çizilmiştir (Şekil 3.29, Şekil 3.30, Şekil 3.31).



Şekil 3.29 Model 1 (Basic t-shirt) üretim bandı



Şekil 3.30 Model 2 (Elbise) üretim bandı



Şekil 3.31 Model 3 (Tayt) üretim bandı

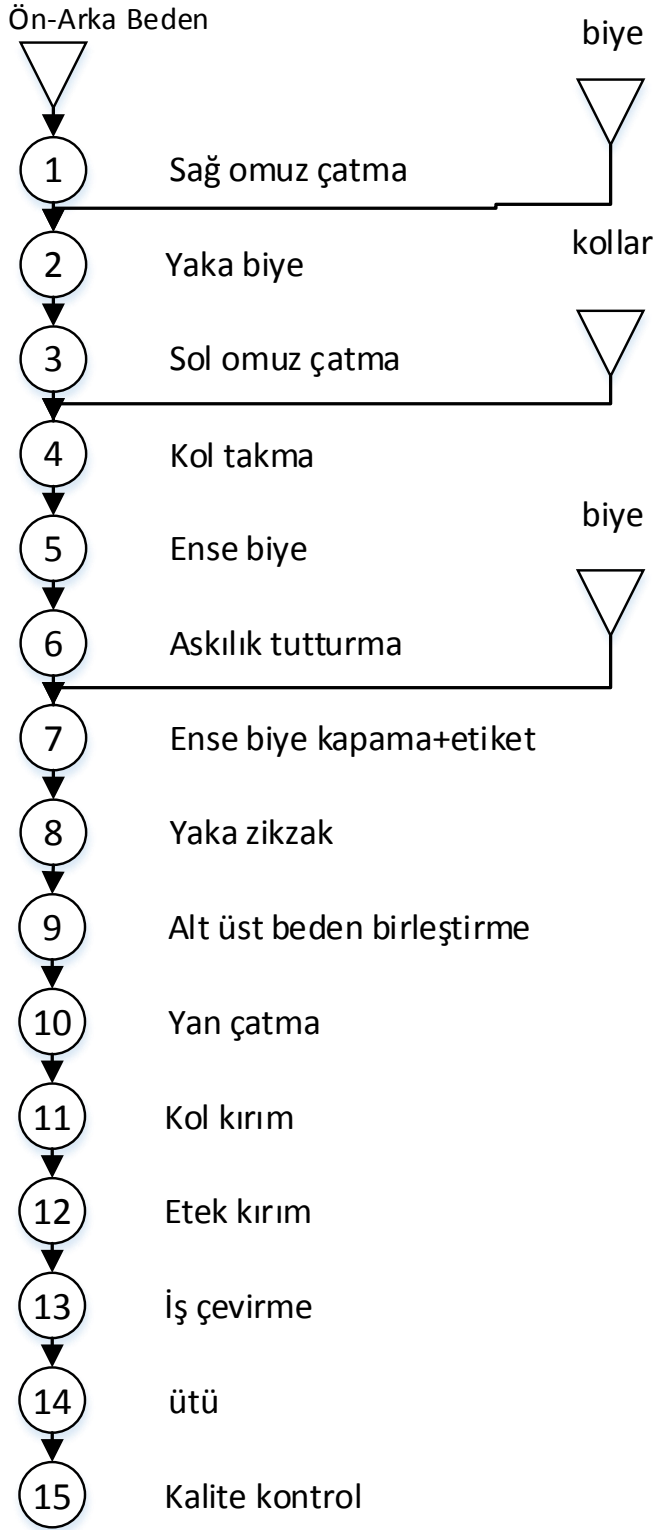
3.6 Model Kurulum ve İş Akış Diyagramlarının Belirlenmesi

Projede belirlenen modellerin SIMUL 8 programına aktarılması için gerekli iş akış diyagramları oluşturulmuştur.

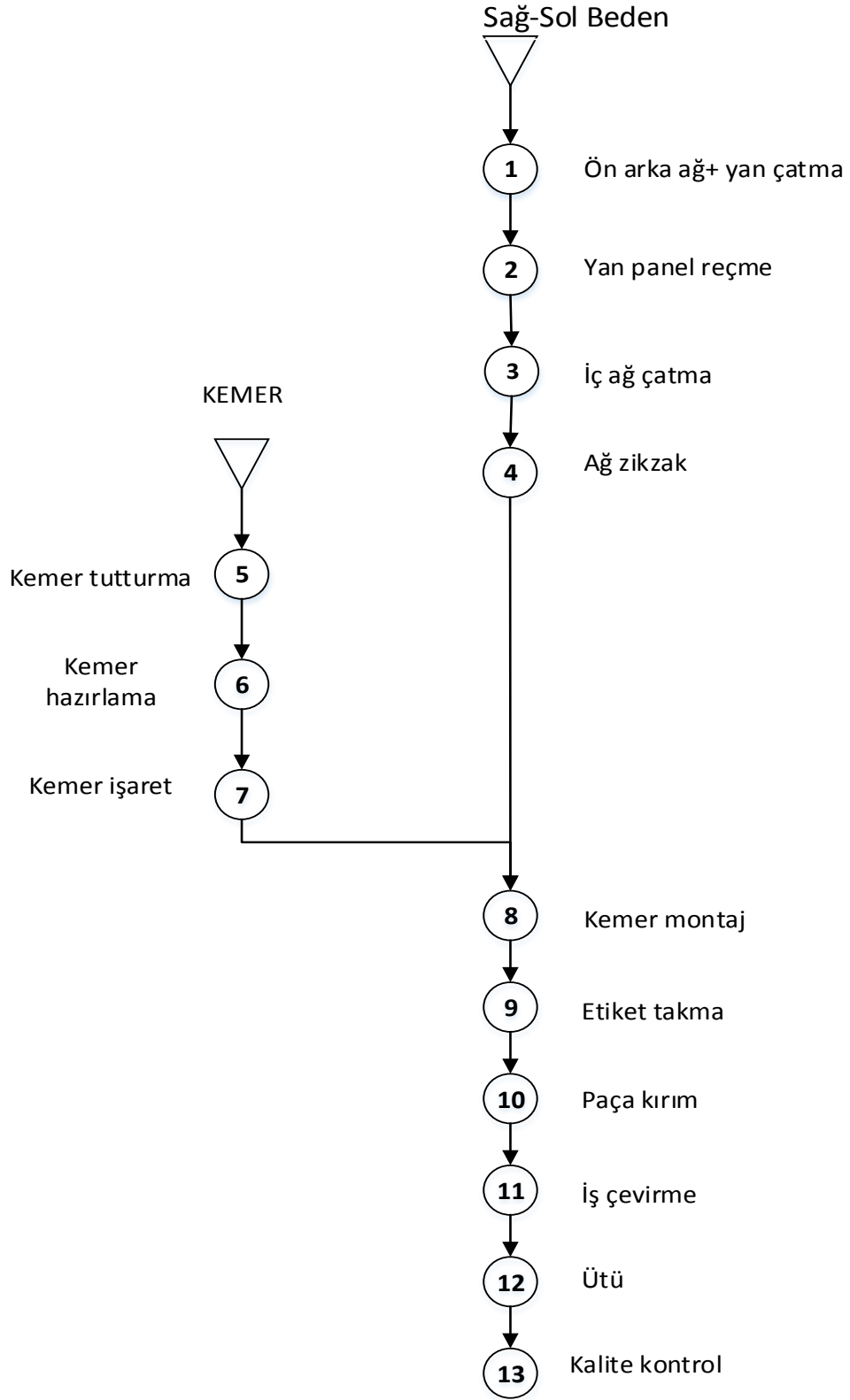
Ön-Arka Beden



Şekil 3.32 Model 1 iş akış diyagramı



Şekil 3.33 Model 2 iş akış diyagramı



Şekil 3.34 Model 3 iş akış diyagramı

Üretim akışları ve bantların kuruluş biçimi oluşturulduktan sonra her operasyonun gösterdiği istatistiksel dağılımı belirlemek amacıyla REFA Etüt formları örnek alınarak kronometreyle zaman tutulmuştur. Her operasyon için 60 adet ölçüm alınmıştır. Aşağıda Model 1'in "sağ omuz çatma" operasyonu için alınan zaman değerleri örnek olarak verilmektedir.

Çizelge 3.1 Model 1 sağ omuz çatma operasyonu değerleri

Sağ Omuz Çatma Değerleri (sn)									
13,1	13,1	12,9	16,07	11,5	13,1	12,5	13,3	12,6	13,1
13,3	12,5	12,7	1,7	14,03	13,9	27,6	17,2	15,2	20,6
13,2	14,8	12,9	13,2	11,2	13,1	12,2	10,8	12,5	12,3
17,7	14,9	17,2	15,6	15,7	16,5	21,4	18,2	14,9	16,1
11,7	12,4	14,1	12,30	12,5	13,2	11,01	15,2	12,7	11,1
13,9	15,4	24,5	18,90	20,2	18,1	15,9	19,9	16,4	15,9

3.7 Alternatif İş Akış Diyagramlarının Oluşturulması

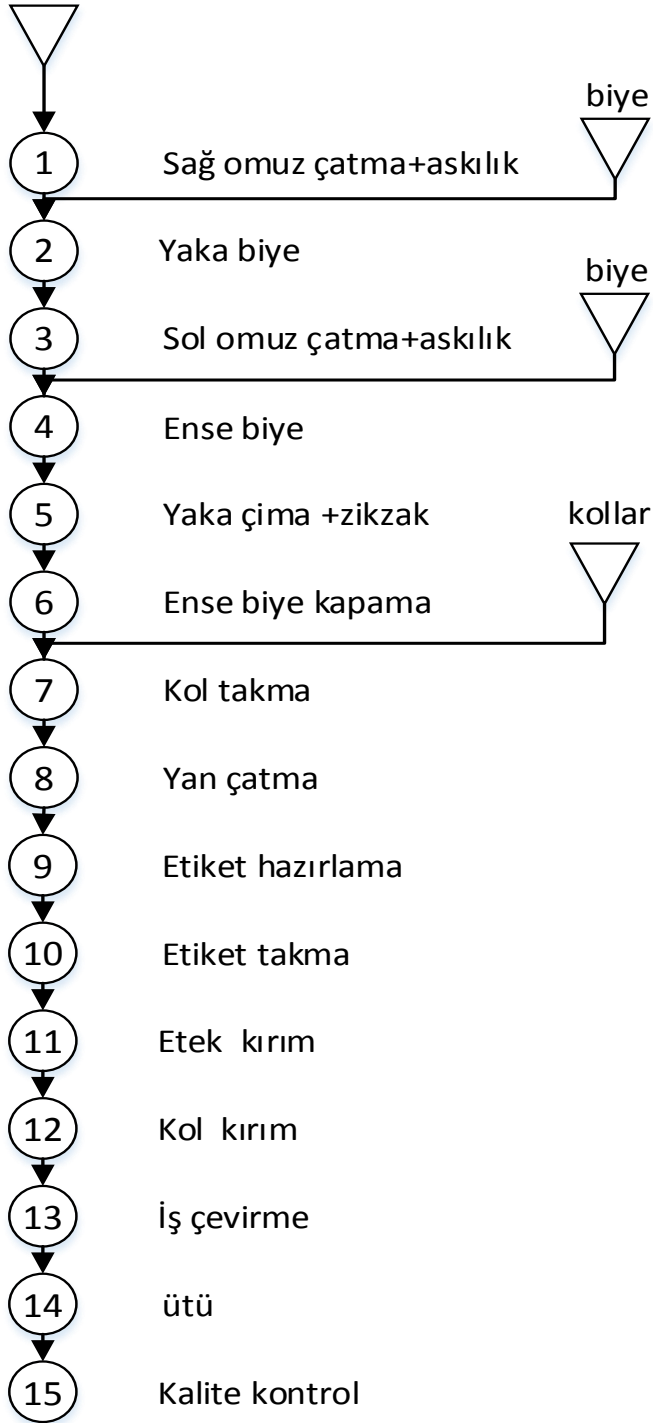
Projede belirlenen modellere alternatif akışlar oluşturularak maksimum hat verimliliği ve kişi başı üretim adedi hedeflenmiştir. Alternatif modeller kısa kodlamalar yapılarak isimlendirilmiştir. Örneğin; Model 1'e ait 1. Alternatif Model M1A1 ve bu alternatif için geliştirilen modeller M1A11, M1A12 olarak isimlendirilmeye devam ettirilmiştir. Alternatif akışlar her model için mümkün olan operasyonların sıralaması değiştirilerek oluşturulmuştur. Operasyon sıralamaları değiştirilerek daha verimli, kişi başı üretim adedi daha fazla simülasyon modelleri kurmak amaçlanmıştır. Örneğin; Model 1'de ense biye 6.operasyon iken M1A1 de ense biye 4. operasyon olarak yapılmıştır (Şekil 3.35). Model 2 kol takma operasyonu 4. sırada yapılırken alternatif M2A1 de ense biye 4. operasyon sonrasında ense biye kapama, askılık tutturma, yaka zikzak, alt-üst beden birleştirme ve 9. operasyon olarak kol takma yapılarak yeni bir alternatif oluşturulmuştur (Şekil 3.39). Bu şekilde operasyon sıralamaları değiştirilerek model 1 için 4 farklı alternatif (Şekil 3.35, Şekil 3.36, Şekil 3.37, Şekil 3.38), model 2 için 3 farklı alternatif (Şekil 3.39, Şekil 3.40, Şekil 3.41), Model 3 için 1 farklı alternatif (Şekil 3.42) oluşturulmuştur.

Ön-Arka Beden



Şekil 3.35 M1A1 iş akış diyagramı

Ön-Arka Beden



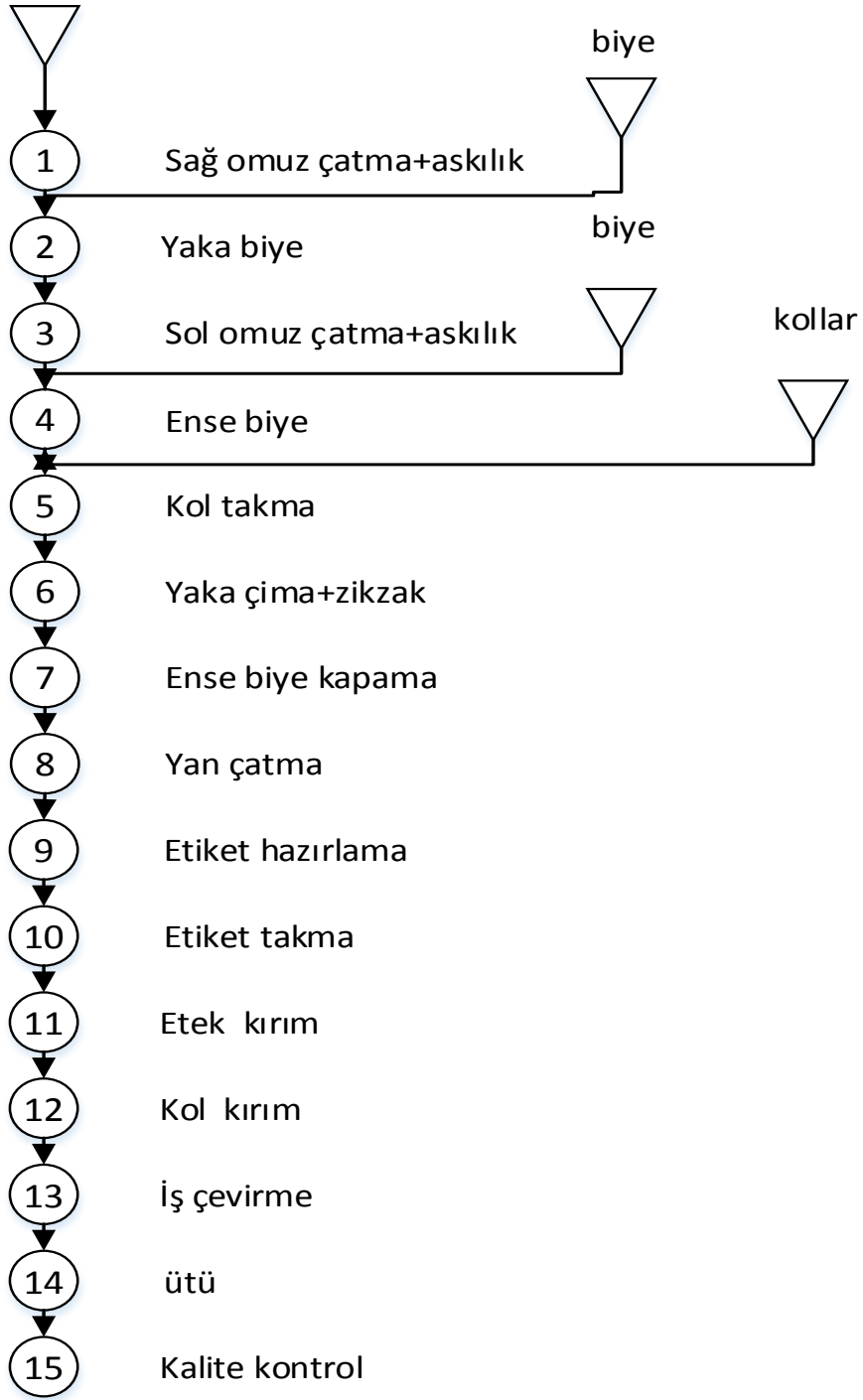
Şekil 3.36 M1A2 iş akış diyagramı

Ön-Arka Beden

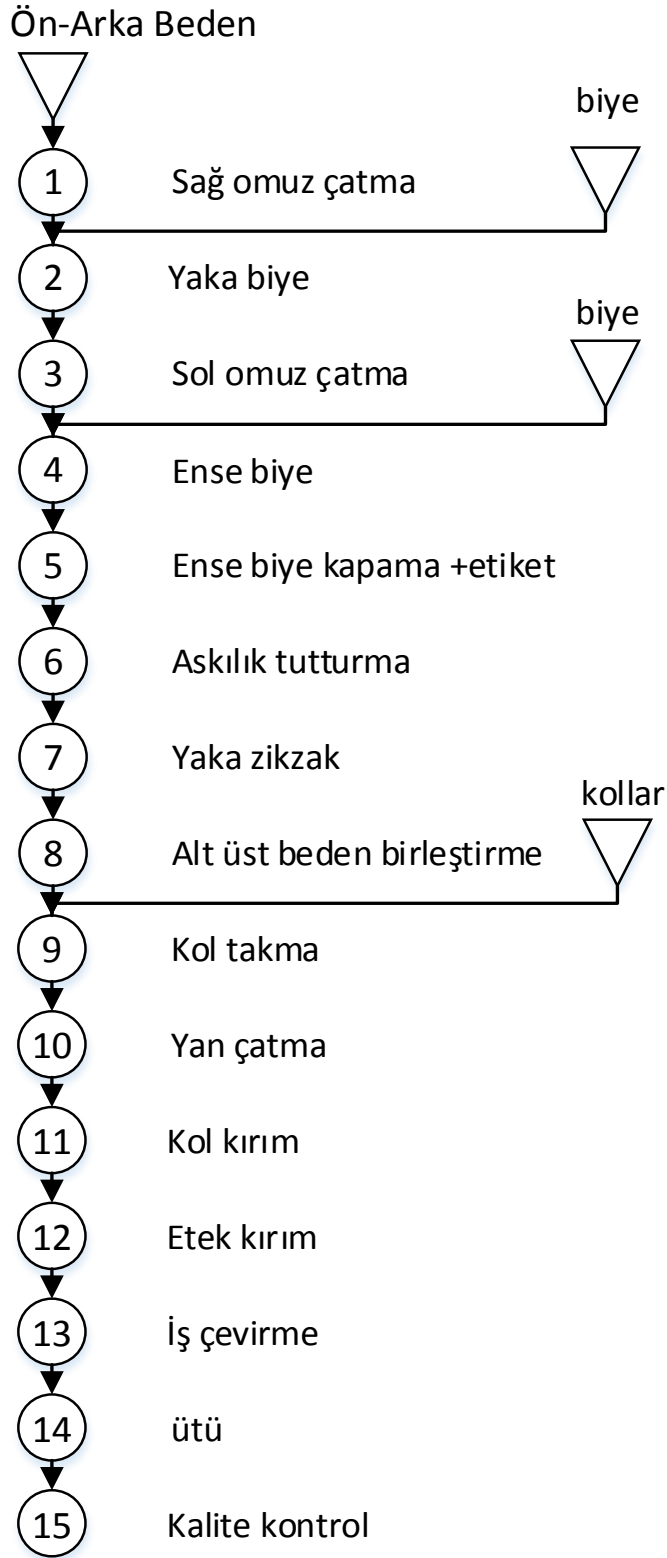


Şekil 3.37 M1A3 iş akış diyagramı

Ön-Arka Beden



Şekil 3.38 M1A4 iş akış diyagramı

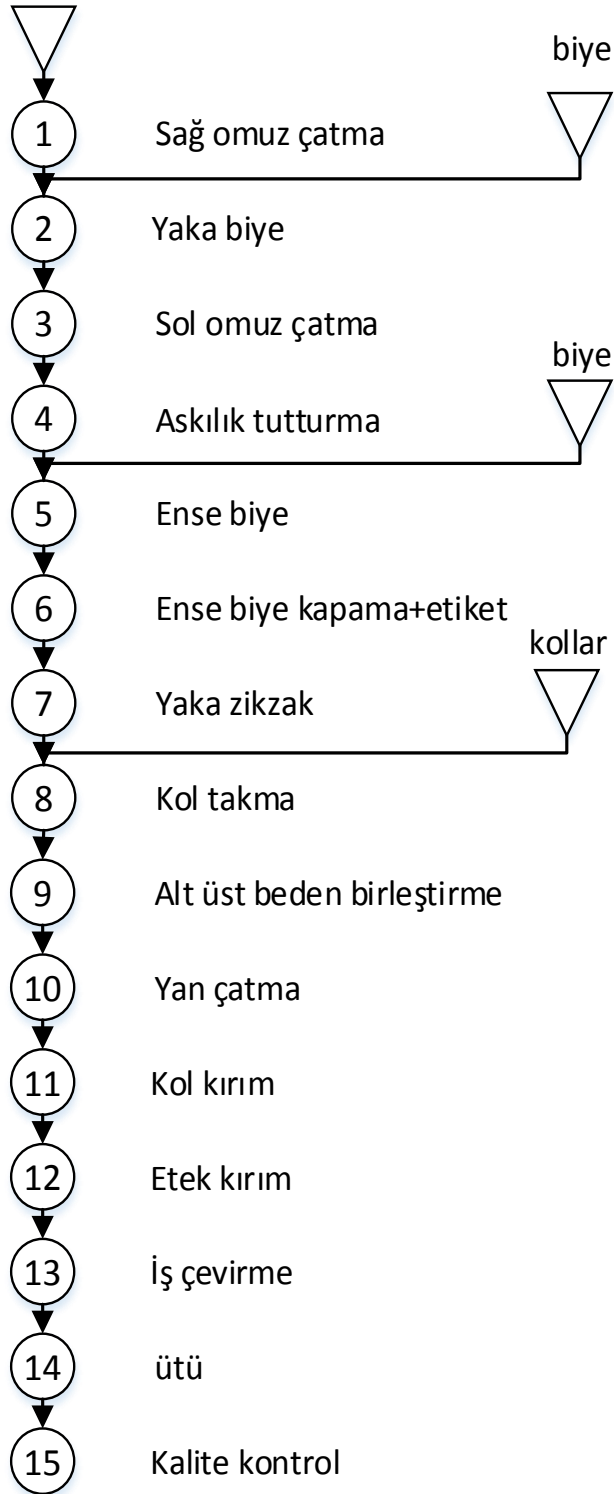


Şekil 3.39 M2A1 iş akış diyagramı

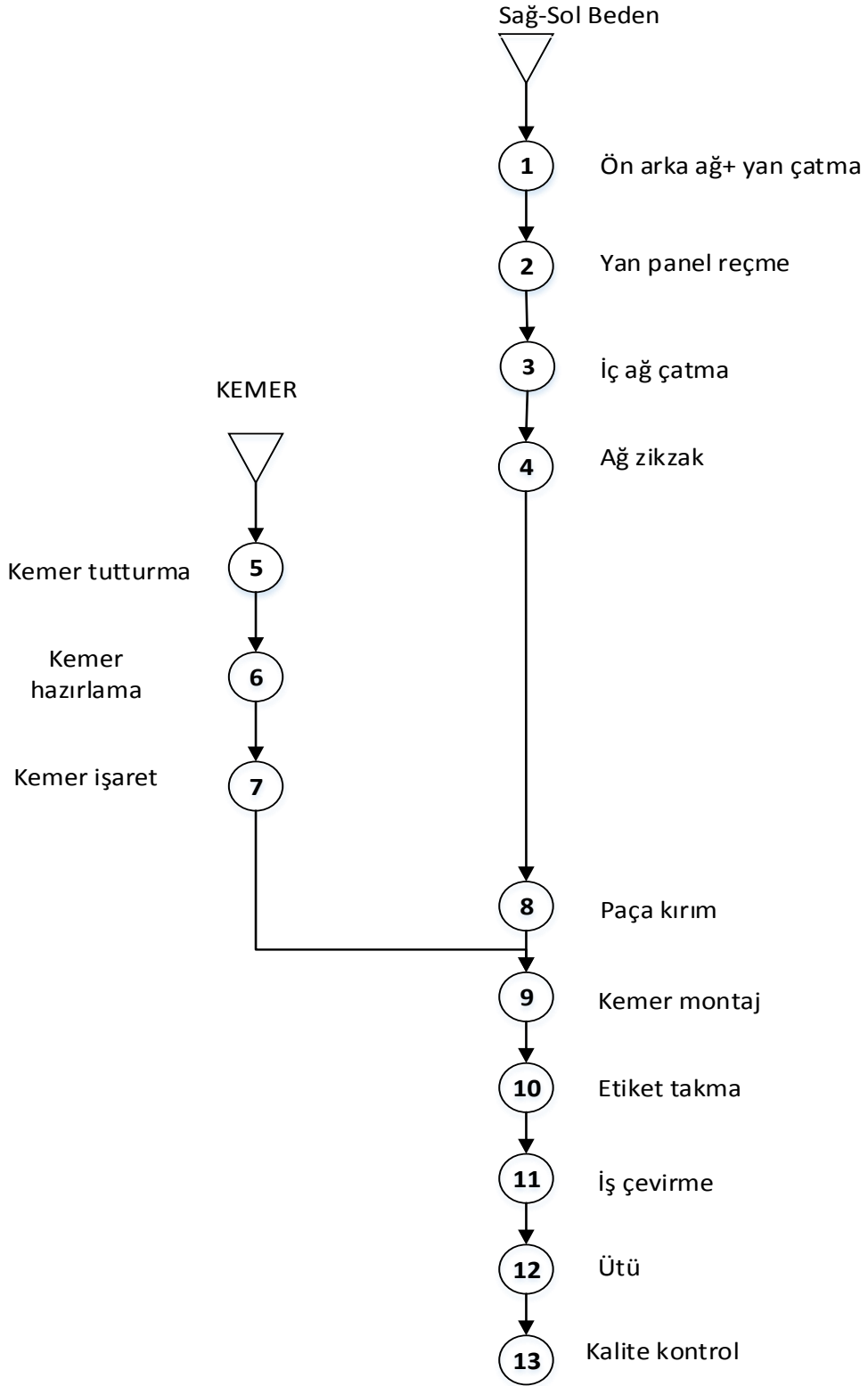


Şekil 3.40 M2A2 iş akış diyagramı

On-Arka Beden



Şekil 3.41 M2A3 alternatif iş akış diyagramı



Şekil 3.42 M3A1 Alternatif iş akış diyagramı

4. ARAŞTIRMA BULGULARI

4.1 Operasyonlara Ait Dağılımların Belirlenmesi

Her operasyonun gösterdiği istatistiksel dağılımı belirlemek amacıyla REFA Etüt formları örnek alınarak kronometreyle zaman tutulmuştur. Her operasyon için 60 adet ölçüm alınmıştır. Çizelge 4.1’de Model 1’in “sağ omuz çatma” operasyonu için alınan zaman değerleri örnek olarak verilmektedir. Bu dağılım sürelerinin içerisinde çalışanın iplik değiştirmesi, iplik kopması durumu da yer almaktadır.

Çizelge 4.1 Model 1 sağ omuz çatma operasyonu değerleri

Sağ Omuz Çatma Değerleri (sn)									
13,1	13,1	12,9	16,07	11,5	13,1	12,5	13,3	12,6	13,1
13,3	12,5	12,7	1,7	14,03	13,9	27,6	17,2	15,2	20,6
13,2	14,8	12,9	13,2	11,2	13,1	12,2	10,8	12,5	12,3
17,7	14,9	17,2	15,6	15,7	16,5	21,4	18,2	14,9	16,1
11,7	12,4	14,1	12,30	12,5	13,2	11,01	15,2	12,7	11,1
13,9	15,4	24,5	18,90	20,2	18,1	15,9	19,9	16,4	15,9

Çizelge 4.2 Model 1'e ait operasyonların istatistiki dağılımı

	MODEL 1	Dağılım Türü	Önerilen
1	Sağ omuz çatma+askılık	Gamma (min, alpha, beta)	Gamma(10,6.3,0.946)
2	Yaka biye	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6(13,2,2.98,5.47)
3	Sol omuz çatım+askılık	Gamma (min, alpha, beta)	Gamma(17,3.17,2.77)
4	Kol takma	Gamma (min, alpha, beta)	Gamma(32,2.66,5.34)
5	Yan çatım	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (13,2,2.98,5.47)
6	Kol kırım	Weibull (min, alpha, beta)	Weibull (29,1.5,8.4)
6b	Kol kırım eğitim	Weibull (min, alpha, beta)	Weibull (39,1.64,15.5)
7	Etek kırım	Gamma (min, alpha, beta)	Gamma (19,1.72,2.6)
8	Ense biye	Weibull (min, alpha, beta)	Weibull (6,2.44,6.39)
9	Yaka çima+yaka zikzak	Pearson 6 (min, beta, alpha1, alpha2)	Pearson6(17,3.14,9.68,22.9)
10	Ense biye kapama	Triangular (min, max, mode)	Triangular (21,40.6,24.1)
11	Etiket hazırlama	Pearson 5 (min, alpha, beta)	Pearson 5 (8,2.38,10.3)
12	Etiket takma	Pearson 6 (min, beta, alpha1, alpha2)	Pearson6 21,2.64,8.59,27.5)
13	İş çevirme	Lognormal (min, mu, sigma)	Lognormal (2,0.705,0.457)
14	Ütü	Weibull (min, alpha, beta)	Weibull (39,1.23,17.5)
15	Kalite	Weibull (min, alpha, beta)	Weibull (18,1.55,14.9)
16	Kol takma+yan çatma	Weibull (min, alpha, beta)	Weibull (52,2.37,27.4)

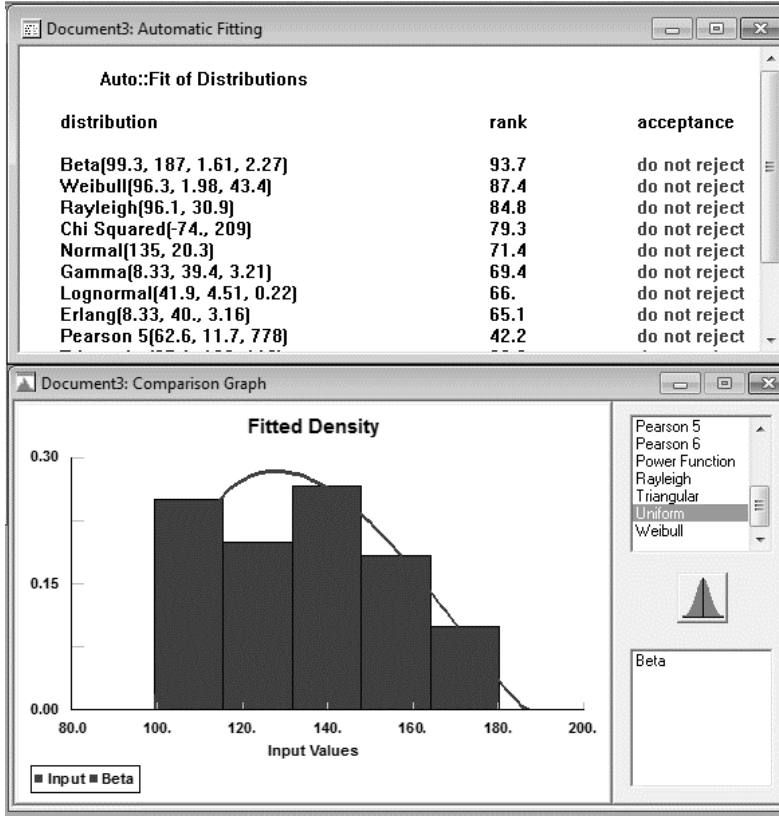
Üç farklı model için toplamda 43 farklı operasyonda ilgili değerler toplanmıştır. Daha sonra bu değerlerin hepsi Simul8 yazılımının Stat-Fit modülü ile değerlendirilerek istatistiksel dağılımlar elde edilmiştir (Çizelge 4.2, Çizelge 4.3, Çizelge 4.4). Stat-Fit modülü mevcut değerlere en uygun gelen yoğunluk fonksiyonunu otomatik olarak hesaplamaktadır (Şekil 4.1.).

Çizelge 4.3 Model 2'ye ait operasyonların istatistiki dağılımı

	MODEL 2	Dağılım Türü	Önerilen
1	Sağ Omuz Çatma	Erlang (min, average, k)	Erlang (5, 3, 4.54)
2	Yaka Biye	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (9, 28.1, 4.64, 38.6)
3	Sol Omuz Çatma	Beta (min, max, alpha1, alpha2)	Beta (7, 20, 1.45, 3.4)
4	Ense Biye	Beta (min, max, alpha1, alpha2)	Beta (17, 52.1, 2.73, 9.65)
5	Ense Biye Kapama + Etiket	Beta (min, max, alpha1, alpha2)	Beta (17, 42.6, 5.06, 5.51)
6	Askılık Tutturma	Weibull (min, alpha, beta)	Weibull (19, 1.58, 8.03)
7	Yaka Zikzak	Erlang (min, average, k)	Erlang (7, 3, 1.96)
8	Alt Üst Beden Birleştirme	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (41, 37, 2.41, 7.01)
9	Kol Takma	Weibull (min, alpha, beta)	Weibull (35, 2, 14.8)
10	Yan Çatma	Beta (min, max, alpha1, alpha2)	Beta (37, 98.7, 1.51, 2.03)
11	Kol Kırım	Weibull (min, alpha, beta)	Weibull (35, 2, 15.6)
12	Etek Kırım	Weibull (min, alpha, beta)	Weibull (28, 2.14, 11.6)
13	İş Çevirme	Triangular (min, max, mode)	Triangular (2, 6.99, 4.16)
14	Ütü	Weibull (min, alpha, beta)	Weibull (42, 2.17, 43.7)
15	Kontrol	Erlang (min, average, k)	Erlang (22, 3, 13.3)

Çizelge 4.4 Model 3'e ait operasyonların istatistiki dağılımı


	MODEL 3	Dağılım Türü	Önerilen
1	Ön arka Ağ + Yan Çatma	Beta (min, max, alpha1, alpha2)	Beta (99, 180, 1.29, 1.72)
2	Yan Panel Reçme	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (51, 54.7, 1.35, 7.03)
3	İç Ağ Çatma	Weibull (min, alpha, beta)	Weibull (41, 1.53, 18.1)
4	Ağ Zikzak	Lognormal (min, mu, sigma)	Lognormal (2, 0.771, 0.456)
5	Kemer Tutturma	Gamma (min, alpha, beta)	Gamma (5, 2.25, 2.58)
6	Kemer Hazırlama	Pearson 5 (min, alpha, beta)	Pearson 5 (27, 1.21, 4.22)
7	Kemer İşaret	Erlang (min, average, k)	Erlang (2, 4, 0.566)
8	Kemer Montaj	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (27, 28.2, 3.64, 9.19)
9	Etiket Takma	Beta (min, max, alpha1, alpha2)	Beta (6, 26.8, 2.65, 7.58)
10	Paça Kırım	Pearson 6 (min, beta, alpha1, alpha2)	Pearson 6 (20, 375, 1.63, 53.6)
11	İş Çevirme	Weibull (min, alpha, beta)	Weibull (5, 3.77, 4.82)
12	Ütü	Beta (min, max, alpha1, alpha2)	Beta (35, 74.3, 1.16, 2.06)
13	Kontrol	Weibull (min, alpha, beta)	Weibull (24, 2.03, 24.9)

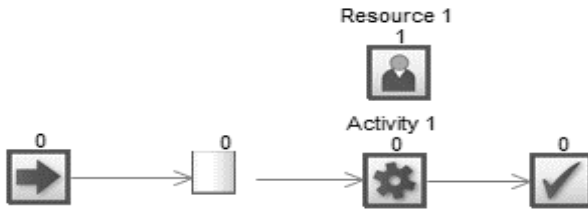


Şekil 4.1 Stat-Fit modülü görüntüsü

4.2 Modelin Simul8 Yazılımına Aktarılması

Şekil 4.2’de görüldüğü gibi bir model oluşturmak için yapılacak ilk şey, modele ait akış sıraları göz önüne alınarak, araç çubuklarından “Başlangıç Noktası”nı seçmek ve ekrana sürüklemektir. Daha sonra aynı şekilde “Kuyruk” ve “Aktivite” ekrana sürüklenir. Eğer şekiller birbirlerine otomatik bağlanmazlarsa, onları daha sonra oklar yardımıyla birbirine bağlamakta mümkündür. Her aktiviteye ait kaynakları eklemek için araç çubuklarında “Create Resources” düğmesine tıklamak gerekmektedir. Kaynak listesinden ihtiyaç duyulan kaynaklar eklenir.

Model oluşuncaya kadar yukarıdaki işlemler tekrar edilir ve araç çubuklarındaki  ikona basılarak model çalıştırılır.



Şekil 4.2 Örnek model

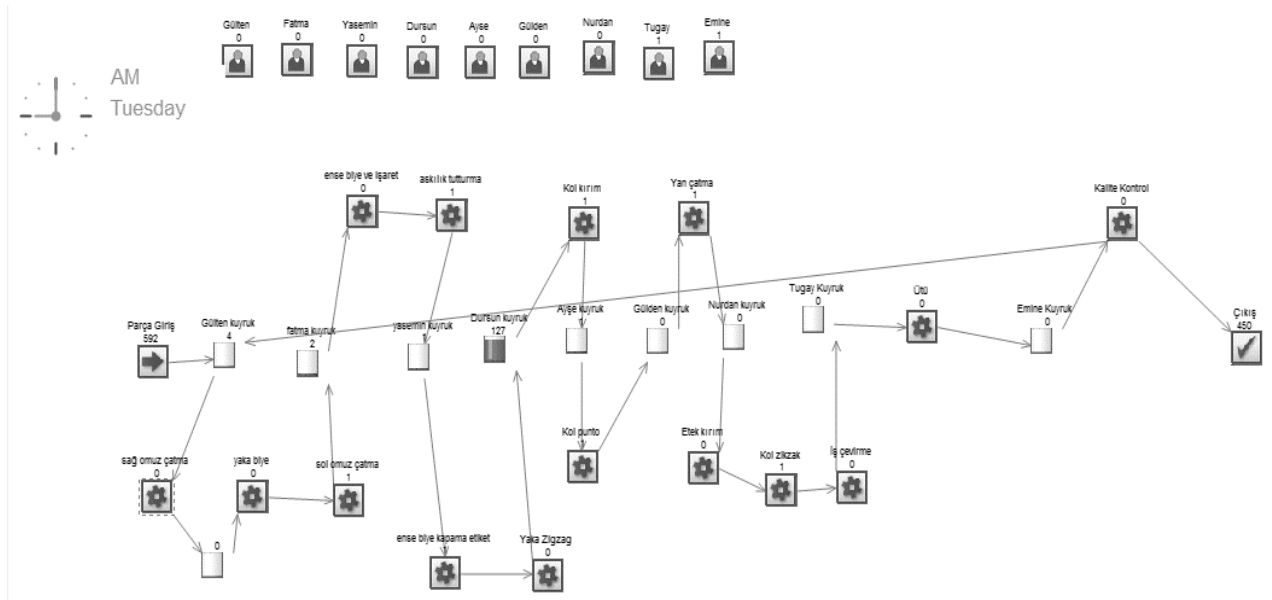
4.3 Model Simul8 Yazılımına Aktarılırken Kabul Edilen Varsayımlar

Simülasyon çalışmalarında kurulan modellerin gerçek sisteme ait tüm detayları içermesi mümkün olmayacağı için, sıklıkla modellerle ilgili varsayımlarda bulunmaktadır. Varsayımlar dışında kalan etkenler ise model içerisinde yer almaktadır. Konfeksiyon üretiminde gerçekleştirilen bu proje çalışmasında aşağıda varsayımlarda bulunulmuştur;

- Üretim, elektrik kesintisi gibi herhangi bir dış etkene bağlı olarak duraksamamaktadır,
- Çalışanlar, üretim süresi boyunca herhangi bir sebeple çalışma alanını terk etmemektedir,
- Üretim bandının beslenmemesi, parça beklemesi gibi bir durum söz konusu değildir,
- Herhangi bir üretim bandı gün içerisinde model değişikliği yapmamaktadır.
- Çalışanın iplik değiştirmesi durumu dağılım sürelerinin içerisinde yer almaktadır.

İş akış diyagramlarına uygun olarak oluşturulan modelin, gerçeğe daha yakın özellikler gösterebilmesi için Simul8’de yazılımında aşağıdaki değişkenler modellere eklenmiştir.

- Makine arıza yüzdeleri,
- Kalite kontrolden dönüş yüzdeleri,
- Modeli (gereğinden fazla kuyruk oluşturmayacak) en uygun besleme zamanının (sn) belirlenmesi, (aşağıda belirtildiği üzere daha sonra giriş besleme miktarı tanımlanması tercih edilmiştir) (Şekil 4.3).



Şekil 4.3 SIMUL8 programında oluşturulan model arayüzü

4.3.1 Makine Arıza Yüzdeleri

Uygulama verileri toplanan firmadan 11 aylık makine türlerine göre arıza sayıları ve arıza süreleri alınmıştır. Proje için belirlenen modellerde overlok, reçme ve düz dikiş makinesi kullanılmaktadır (Çizelge 4.5, Çizelge 4.6).

Çizelge 4.5 Makine sayıları

Makine cinsi	Sayısı
Overlok	178
Reçme	118
Düz	180
Özel	56

Çizelge 4.6 Makine türlerine göre arıza süreleri (dk)

Aylar	Ocak	Şubat	Mart	Nisan	Mayıs	Haziran	Temmuz	Ağustos	Eylül	Ekim	Kasım
OVERLOK	574	869	1.428	1.928	1.316	1.471	1.630	738	517	545	856
REÇME	2.060	1.529	2.927	2.194	3.283	3.958	2.359	1.246	835	1.155	1.993
DÜZ	947	1.263	1.740	1.605	1.902	1.516	2.028	785	652	990	1.070
ÖZEL	224	98	465	197	404	383	295	62	87	131	166
TOPLAM	3.805	3.759	6.560	5.924	6.905	7.328	6.312	2.831	2.091	2.821	4.085

Simul8 yazılımında oluşturulan modelin bu gerçeğe uygun çalışabilmesi için makinaların arıza oranları operasyonlarda kullanılan makine türlerine göre tanımlanmıştır.

Bu verilere göre;

1 ayda ortalama 22 gün çalışılmaktadır.

Overlok makinası 11 ayda toplam 11872 dk arıza yapmıştır. Aylık arıza yüzdesini bulmada şu formül uygulanmıştır;

$$\text{Aylık Arıza yüzdesi} = \frac{\text{Toplam arıza süresi}}{\text{Günlük çalışma süresi (dk)} * \text{gün sayısı} * \text{ay sayısı}}$$
$$= \frac{11872}{540 * 22 * 11} = \%9$$

Elde edilen bu rakam tüm overlok makineleri için geçerli olduğundan, bulunan yüzdeler makine sayısına bölünerek bir makinenin ortalama arıza oranı bulunmuştur;

Overlok makinesi arıza oranı = %0,05

Reçme makinası arıza oranı = %0,16

Düz dikiş makinesi arıza oranı = %0,06

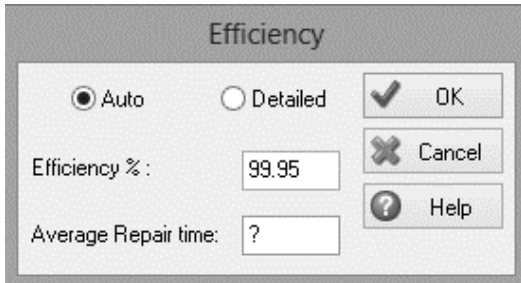
Belirlenen makine arıza oranlarına göre Simul8 yazılımında oluşturulan model üzerinde aktivite özellikleri sekmesinden makine etkinlikleri her operasyonda kullanılan makine türüne göre;

Overlok makinası için %99,5

Recme makinası için %99,84

Düz makine için %99,94 olarak tanımlanmıştır.

İlgili arıza yüzdelerinin Simul8 programına girişi Şekil 4.4’de gösterilmektedir



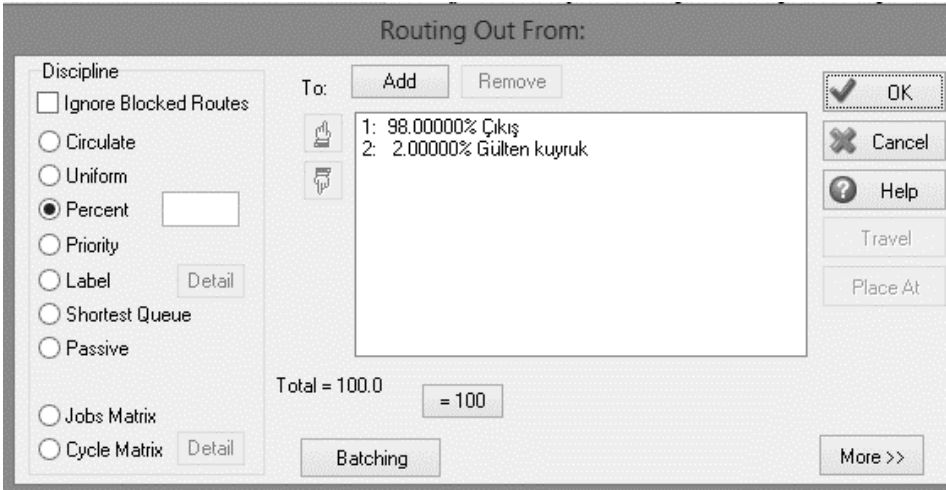
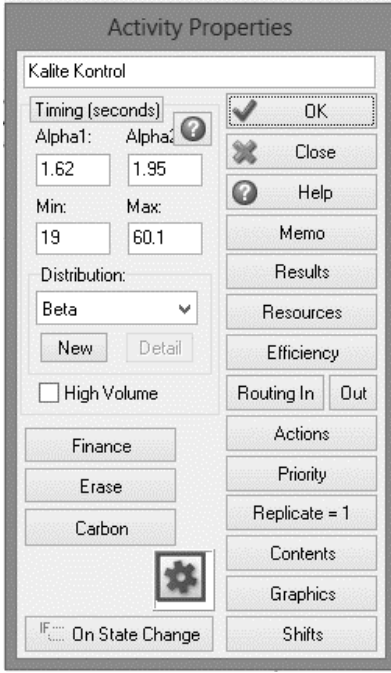
Şekil 4.4 Aktivite özellikleri penceresi

4.3.2 Kalite Kontrolden Dönüş Yüzdeleri

Üretimde iş akışı süresince kalite kontrolde kontrol edilen ürünlerde hatalar çıkmaktadır. Hatalı ürünler tekrar üretime geri verilerek akışa dâhil edilmektedir. Projede belirlenen modellerde tamir dönüş oranları;

- Model 1 = %2,
- Model 2 = %5,
- Model 3 = % 2 olarak belirlenmiştir.

Belirlenen kalite kontrol dönüş oranları Şekil 4.5’de aktivite özellikleri penceresinde “routing out” sekmesinden tanımlanmıştır.



Şekil 4.5 Aktivite özellikleri tamir oranları tanımlama

4.3.3 Modeli en uygun besleme zamanının belirlenmesi

Yazılımda oluşturulan modelin beslenmesi için, iki farklı yöntem ele alınmıştır;

- Tek tek besleme,
- Demet halinde besleme.

Tek tek besleme sisteminde, her parça için en uygun zaman aralıklarını belirlemek amacıyla farklı süreler için ilgili modelin günlük çıktı adedi gözlemlenmiştir. Örneğin model için, parçaların modele giriş aralıkları (exponential dağılımda) 74 sn'den başlanarak her denemede 1 sn düşürülmüş, elde edilen günlük çıktı adedi gözlemlenmiştir. Modele ait parça giriş süreleri azaldıkça (model daha sık beslendikçe) modelin günlük üretim adedinin arttığı belirlenmiştir. Ancak bu artış belirli bir besleme süresinden sonra sabitlenmektedir. Bunun sebebi ise modele ait belirli bir çevrim süresinin olması ve bandın çalışma hızının en yavaş operasyona bağlı

olmasıdır. Önemli olan model içinde çok fazla kuyruk olmadan, maksimum üretim adedine ulaşmasını sağlamaktır. Bu amaç doğrultusunda, besleme süresi ile model çıktı adedi arasındaki ilişkiyi ifade eden doğrusal denklemin eğiminin “0” olduğu ilk nokta “kritik besleme süresi” olarak belirlenmiştir.

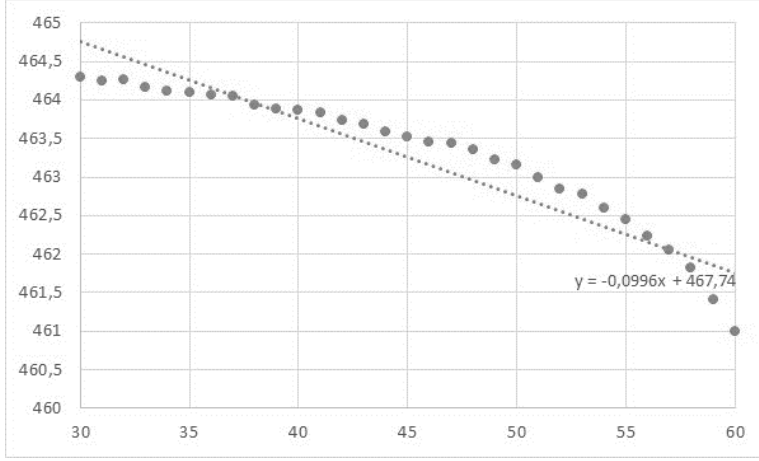
Örnek model için gerçekleştirilen besleme süresi ve günlük çıktı adedi arasındaki değerler Çizelge 4.7’de verilmektedir. 74 sn’den başlayarak 30 sn’ye kadar düşürülen besleme sürelerinden görüldüğü üzere 60. sn’den sonra çıktı adedinin 463 adet civarlarında sabitlendiği belirlenmiştir. Çizelge 4.7’de elde edilen değerlerin küsuratlı olma sebebi ise her besleme saniye aralığı için modelin 30 defa çalıştırılması ve günlük ortalama çıktı adedinin dikkate alınmasıdır.

Çizelge 4.7 Örnek model besleme süreleri ve çıktı adedi ortalamaları

BS	ÇO	BS	ÇO	BS	ÇO
74	421,15	59	461,4	44	463,58
73	425,82	58	461,82	43	463,68
72	430,52	57	462,05	42	463,73
71	434,97	56	462,23	41	463,83
70	439,35	55	462,45	40	463,87
69	443,18	54	462,6	39	463,88
68	446,63	53	462,77	38	463,93
67	449,75	52	462,85	37	464,05
66	452,58	51	463	36	464,07
65	455,02	50	463,15	35	464,1
64	456,82	49	463,22	34	464,12
63	458,37	48	463,35	33	464,17
62	459,65	47	463,43	32	464,27
61	460,48	46	463,45	31	464,25
60	461	45	463,52	30	464,3

BS: Besleme süresi, ÇO: Çıktı adedi ortalaması

Örnek model çıktı adedi ile besleme süresi arasındaki ilişkiyi ifade eden doğrusal denklem Excel’de çizilmiş, her saniye değişiminde tekrar kontrol edilerek, eğimin 0 olduğu ilk besleme saniyesi (kritik besleme süresi) “61” olarak belirlenmiştir (Şekil 4.6).



Şekil 4.6 Excel’de örnek model besleme kritik süresi hesaplama

Bu yöntemle belirlenen besleme kritik süresi Simul8 programına “Start Point Properties” penceresinde aktarılmıştır (Şekil 4.7).



Şekil 4.7 Giriş besleme zamanının Simul8 yazılımına aktarılması

Giriş besleme süresini bu yöntemle belirlemek yerine geliştirilen alternatif yöntem ise, üretilecek olan tüm parça adedinin (demet halinde) modelin başlangıcında programa aktarılmasıdır. Model’in günlük çalışma süresi işletme (mola saatleri çıkmış haliyle) 9 saattir (32400 sn). Bunu gerçekleştirmek için günlük besleme süresi için dağılım sabitlemiş (fixed dağılım) ve 32400’den büyük bir süre girilmiştir (Şekil 4.8).

Start Point Properties

Parça Giriş

Input Work Item Type:

Main Work Item Type

Inter-arrival times (seconds)

Fixed Value: 32401

Distribution: Fixed

First at start time

Unlimited arrivals

None File

Schedule Sheet

Finance

Erase

Ignore hints about lost Work Items

OK

Cancel

Help

Memo

Results

Batching

Routing Out

Actions

Graphics

Carbon

Şekil 4.8 Giriş besleme zamanının sabitlemesi

Daha sonra Şekil 4.8’de yer alan “Batching” düğmesine basılarak, sabit 700 adet parçanın demet halinde sisteme girmesi sağlanmıştır (Şekil 4.9).

Batching

Batch size leaving this object:

Fixed Value: 700

Distribution: Fixed

New

Detail

OK

Cancel

Help

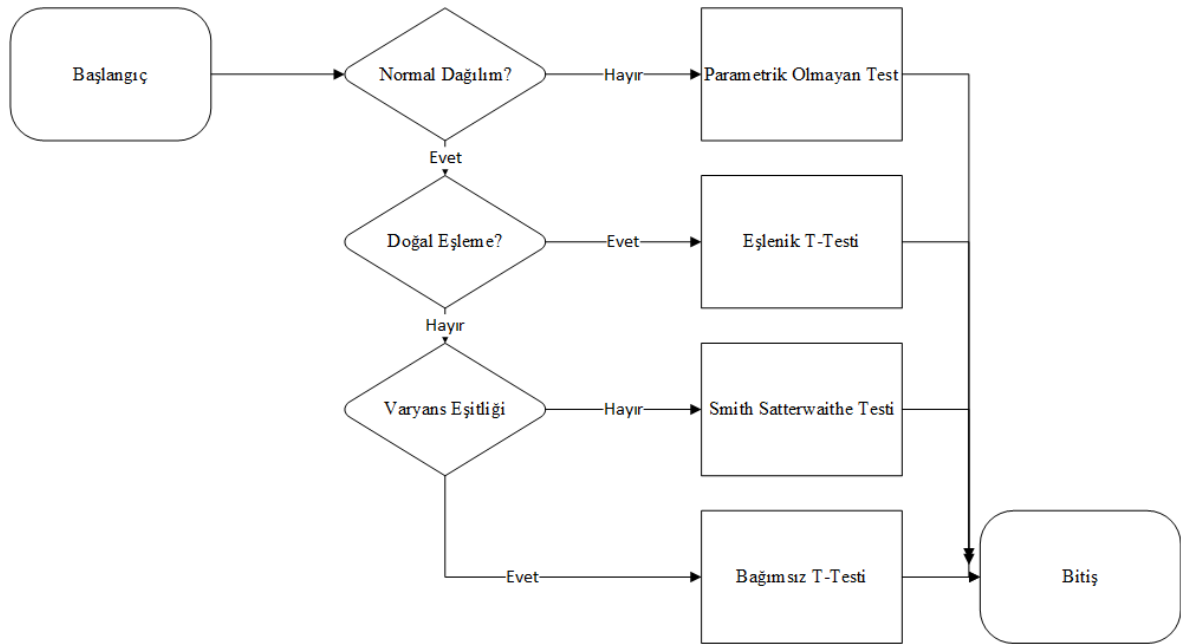
Şekil 4.9 Giriş besleme adedinin sabitlemesi

Bir sonraki parça girişi 32401. saniyede gerçekleşecektir. Ancak model, 1 günlük çıktı adedini raporlayacağı için, ikinci (demet halinde) parça girişi dikkate alınmayacaktır. Bu yöntemle gerçekleştirilen parça girişi, program kullanıcılarını hızlandırmaktadır. Fakat dikkat edilmesi gereken nokta, parça giriş (demet) adedidir. Basic t-shirt bilindiği üzere günlük yaklaşık olarak 570 adet üretmektedir. Bu nedenle başlangıçta 700 adet giriş yapmak yeterlidir. Giriş adedi arttıkça günlük çıktı sayısı değişmeyecek ancak operasyonlar arası kuyruk artacaktır. Kullanıcı kuyrukları daha iyi gözlemlenmek istediği takdirde giriş adedini arttırabilir. Ancak şu

noktaya dikkat edilmelidir; modele demet halinde giriş yapan parçalar, ilk operasyonda kalıcı bir kuyruk oluşturmaktadır. Bu nedenle ilk operasyona ait gerçek bir kuyruk olduğunu belirlemek zorlaşmaktadır. Kullanıcı ilgili problemin üstesinden gelebilmek için “kritik giriş adedini” belirlemelidir. Bunun için, aynı “kritik besleme süresi”nin belirlenmesi gibi belirli aralıkları baz alarak hareket etmek mümkündür. Fakat “kritik giriş adedini” modeldeki kuyrukları gözlemleyerek belirlemek mümkün olduğundan, kullanıcının “doğrusal denklem eğimi” gibi bir parametre hesaplanmasına gerek olmayacaktır. Böylece daha hızlı çalışmak mümkün olacaktır.

4.4 Model Geçerliliği

Gerçek sistemden toplanan ve yazılımda oluşturulan modelden elde edilen verilerin Şekil 4.10’teki işlem sıralamasına göre istatistiksel geçerlilikleri sağlanır.



Şekil 4.10 Hipotez testleri

Gerçek sistem ve simülasyon modeli arasında kantitatif ve objektif karşılaştırma söz konusudur. Veri setleri arasındaki istatistiksel hesaplamalarda, sistemden ve modelden elde edilen günlük üretim adetlerinin karşılaştırması yapılmıştır. İlk olarak her model için iki veri setine ait değerlerin normal dağılım gösterip göstermediği Kolmogrov-Smirnov (KS) testi ile kontrol edilmiştir. KS testinin adımları şu şekildedir;

- H_0 ve $H_{\text{alternatif}}$ hipotezinin kurulması,
- Test önem düzeyinin belirlenmesi,
- KS kritik değerinin hesaplanması,
- Normal dağılım ile gözlem dağılımının en büyük farkının hesaplanması,
- Farkın kritik KS değeri ile karşılaştırılması,
- H_0 'ın kabul ya da ret edilmesi.

Sistem verileri ile model verileri doğal eşlemeden gelmediği için, geçerlilik işleminin devamında varyans analizi yapılmıştır. Bu amaçla (varyans analizi) F testi uygulanmıştır. F testinin uygulanma adımları şu şekildedir;

- Sıfır hipotezi: Her iki grubun varyansları eşittir,
- Alternatif Hipotez: Her iki grubun varyansları eşit değildir,
- Test önem düzeyinin belirlenmesi,
- Veri örneklerinin serbestlik derecelerinin bulunması,
- Test istatistiğinin hesaplanması,
- Test istatistiğinin kritik değeri karşılaştırılması, H_0 'ın kabul ya da ret edilmesi.

Şekil 4.10'da görüldüğü üzere varyans testinin sonucuna göre Smith-Satterwaithe ya da bağımsız t-testi uygulanmaktadır. Tüm bu testlerin uygulanması SPSS ver. 18'de gerçekleştirilmiştir.

Üç farklı model için, hem gerçek üretim değerlerinden hem de Simul8 çıktı adedi sonuçlarından 30 farklı örneklem grubu ele alınmıştır. Toplamda 30 farklı veriye sahip 6 farklı grup için gerçekleştirilen Kolmogorov-Smirnov (KS) testinin sonuçları Şekil 4.11'de verilmektedir.

	Kolmogorov-Smirnov		
	Statistic	df	Sig.
sistem1	,105	30	,200*
model1	,087	30	,200*
sistem2	,130	30	,200*
model2	,136	30	,167
sistem3	,081	30	,200*
model3	,096	30	,200*

Şekil 4.11 Kolmogorov-Smirnov (KS) testi

%95 önem düzeyi için gerçekleştirilen KS testi sonucunda 6 farklı veri setinin de önem düzeyinin 0,05'den büyük olduğu ($p>0,05$) gözlemlenmiştir. Sonuç olarak tüm veriler normal dağılım göstermektedir. Bu aşamadan sonra yapılması gereken; varyans analizine göre hangi testin uygulanacağına karar verilmesidir. SPSS bağımsız t-testi sonuçlarında ilgili varyans testini otomatik olarak gerçekleştirmekte, varyansların eşit olup olmama durumunu göz önüne alarak gerekli sonuçları sunmaktadır.

Sistem 1 ve Model 1'in sonuçları Şekil 4.12'de, Sistem 2 ve Model 2'in sonuçları Şekil 4.13'de, Sistem 3 ve Model 3'ün sonuçları Şekil 4.14'de sırasıyla verilmektedir.

Group Statistics					
Grup	N	Mean	Std. Deviation	Std. Error Mean	
Çıktılar	Sistem1	30	842,7000	14,86758	2,71444
	Model1	30	847,0000	10,08242	1,84079

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Çıktılar	Equal variances assumed	9,031	,004	-1,311	58	,195	-4,30000	3,27974	-10,86510	2,26510
	Equal variances not assumed			-1,311	51,017	,196	-4,30000	3,27974	-10,88429	2,28429

Şekil 4.12 Sistem 1 ve model 1 geçerlilik sonuçları

Group Statistics					
Grup	N	Mean	Std. Deviation	Std. Error Mean	
Çıktılar	Sistem2	30	546,7333	26,71940	4,87827
	Model2	30	541,7667	2,99060	,54601

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Çıktılar	Equal variances assumed	56,086	,000	1,012	58	,316	4,96667	4,90873	-4,85923	14,79257
	Equal variances not assumed			1,012	29,726	,320	4,96667	4,90873	-5,06218	14,99551

Şekil 4.13 Sistem 2 ve model 2 geçerlilik sonuçları

Group Statistics					
Grup	N	Mean	Std. Deviation	Std. Error Mean	
Çıktılar	Sistem3	30	488,6667	16,00934	2,92289
	Model3	30	481,4667	25,97842	4,74299

Independent Samples Test										
		Levene's Test for Equality of Variances		t-test for Equality of Means					95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Çıktılar	Equal variances assumed	5,762	,020	1,292	58	,201	7,20000	5,57129	-3,95214	18,35214
	Equal variances not assumed			1,292	48,250	,202	7,20000	5,57129	-4,00032	18,40032

Şekil 4.14 Sistem 3 ve model 3 geçerlilik sonuçları

İki farklı grubun ortalamalarını karşılaştırmak için bağımsız t-testi, ilk olarak varyans eşitliğini kontrol etmekte, varyanslar eşitse üst satırdaki, değilse alt satırdaki değerler dikkate alınmaktadır. Üst satırla, alt satırdaki veriler arasındaki fark, serbestlik derecesinin (df) yeniden hesaplanmasıdır. Şekil 4.10'da gösterilen Smith-Satterwaithe testi de zaten varyans eşitliği olmadığı durumda kullanılmakta, aynı şekilde ilk olarak iki veri seti için ortak bir "df"

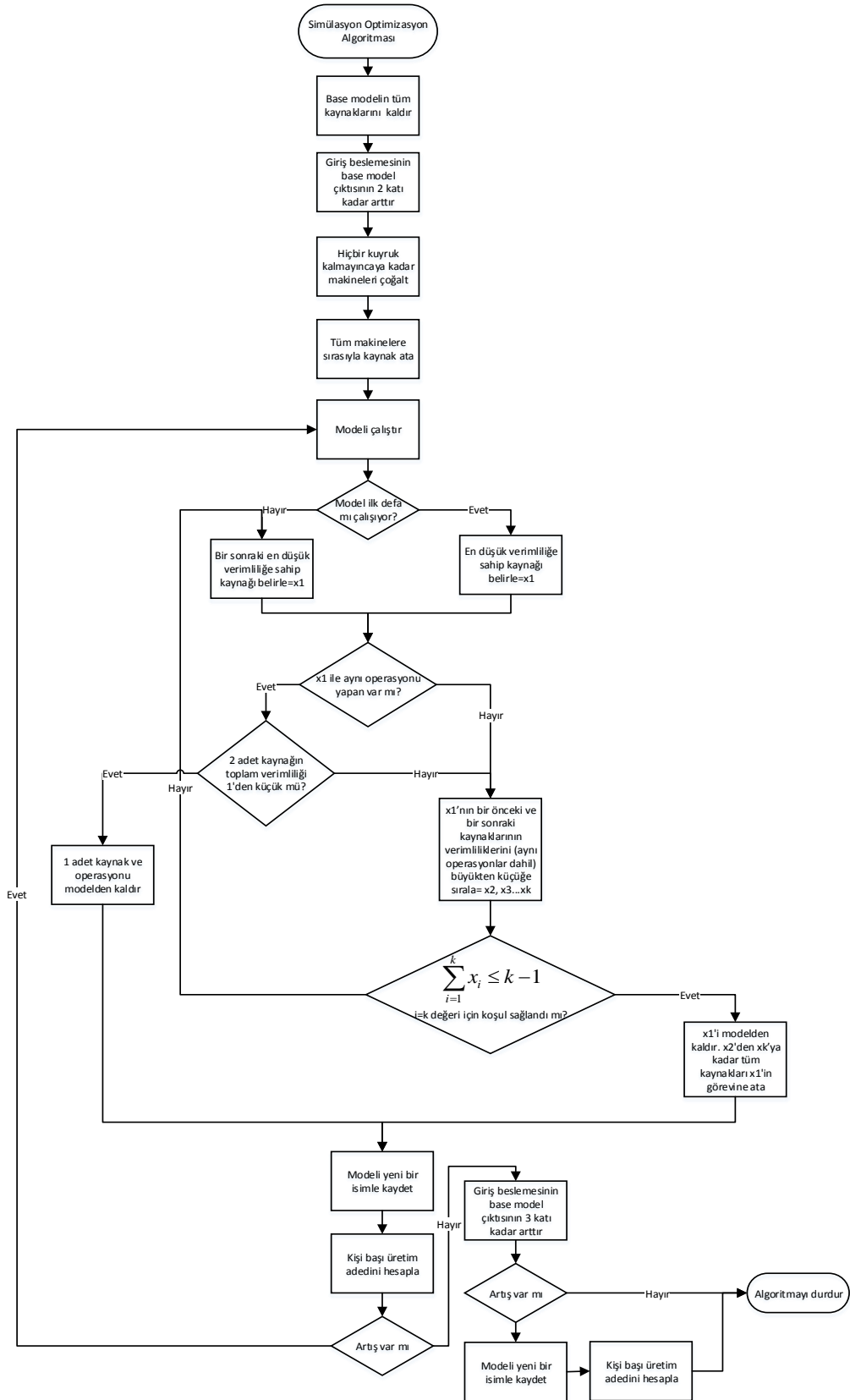
değeri hesaplamaktadır. Daha sonra bu “df” değeri bir üst değere yuvarlanarak t-tablo değeriyle karşılaştırmak için kullanılmaktadır. Örneğin Şekil 4.13’de yer alan Sistem 2 ve Model 2’nin karşılaştırmasında kabul edilen “df” değeri $29,276 \approx 30$ ’dur. Kısacası, SPSS varyans eşitliği ve eşitsizliği durumunda gereken hesaplamaları otomatik olarak gerçekleştirmektedir.

%95 önem düzeyi için gerçekleştirilen bağımsız t-testi sonucunda önem düzeyi 0,05’den büyük olduğu ($p > 0,05$) durumda veri setleri için ortalamaları arasında fark olmadığı kabul edilmekte, böylece geçerlilik sağlanmaktadır.

Sonuçlar şu şekilde özetlenebilir;

- Sistem 1’den elde edilen çıktı adedi ($M=842,7$ $SE=14,86$) model 1’den elde edilen çıktı adedine göre ($M=847$ $SE=10,08$) daha azdır. Bu fark önemsizdir $t(30)=-1,311$, $p=0,195 > 0,05$. Model geçerliliği sağlanmıştır
- Sistem 2’den elde edilen çıktı adedi ($M=546,7$ $SE=4,87$) model 2’den elde edilen çıktı adedine göre ($M=541,76$ $SE=0,54$) daha fazladır. Bu fark önemsizdir $t(30)=1,012$, $p=0,32 > 0,05$. Model geçerliliği sağlanmıştır.
- Sistem 3’ten elde edilen çıktı adedi ($M=488,67$, $SE=2,92$) model 3’ten elde edilen çıktı adedine göre ($M=481,46$ $SE=4,74$) daha fazladır. Bu fark önemsizdir $t(49)=1,292$, $p=0,202 > 0,05$. Model geçerliliği sağlanmıştır.

4.5 Simülasyon Yazılımında Geliştirilen Algoritma



Şekil 4.15 Simülasyon yazılımında geliştirilen algoritma

Şekil 4.15’de geliştirilen algoritmada, esas amaç kaynakların verimliliğini (%100 verimlilik anlamına gelen) “1” değerine olabildiğince yaklaştırmaktır. Bu amaç doğrultusunda işçilere görev atamasının ne şekilde yapılması gerektiği, mevcut sisteme yapılan yoğun “varlık beslemesi” ile belirlenmektedir. Yapılan yoğun besleme kuyrukları ortaya çıkarmakta, kuyrukları azaltmak ise kaynakların verimliliğini artırmakla sağlanmaktadır. Algoritma başında sistemin yoğun bir şekilde beslenmesi, kalbin kanı insan vücuduna pompalamasına benzediği için uygulanan yönteme “kalp algoritması” adı verilmiştir.

Çalışmada simülasyon modeli oluşturulduktan sonra ve ana modelin ürettiği adedin 2 katı besleme yapılarak sistem çalıştırılmıştır. Kuyruk oluşan operasyonlarda, kuyruk kalmayınca kadar makine sayısı artırılmış ve her birine tek bir kaynak atanmıştır. Model bir kez çalıştırıldıktan sonra, operasyonlar arası kuyruk kalmayınca tüm kaynakların verimlilikleri sıralanmış, en düşük verimliliğe sahip kaynaktan başlayarak bir önceki ve bir sonraki kaynağın en uygunu seçilerek tek bir kaynakta birleştirilmiştir.

Örneğin M2A3B (2. modele ait 3. alternatif akışın simülasyon modelinin), SIMUL8 yazılımında operasyon sıralaması oluşturulmuş kuyruk kalmayınca kadar makine ve kaynak sayısı artırılmıştır. Kuyruk kalmadığı anda sistemde toplam 22 kaynak olmuştur ve sistem çalıştırılmıştır. 50 tekrarlı çalıştırılan üretim adedi ortalaması 953,94 olarak SIMUL8 raporlarından alınmış ve excel dosyasına kayıt edilerek kişi başı üretim adedi (953,94/22) 43 olarak hesaplanmıştır. Yine SIMUL8 raporlarında kaynakların verimlilikleri incelenmiş en düşük verimliliğe sahip kaynaklar belirlenmiş, bir önceki ve bir sonraki operasyon ile toplam verimlilik %100 geçmeyecek şekilde birleştirme ihtimali düşünülmüş ve kaynaklar birleştirilerek sistem 21 kaynak ile çalıştırılmıştır. SIMUL8 raporlarından üretim adedi 953,62 olarak alınmış ve excel dosyasına kayıt edilerek kişi başı üretim adedi 45 olarak hesaplanmış ve yeni model M2A31 olarak kayıt edilmiştir. Kişi başı üretim adedi artmaya devam ettiği sürece kaynak birleştirme işlemi bu şekilde devam ettirilmiştir.

Birleştirilen her kaynak sonucunda model çalıştırılmış, toplam üretim adedi üzerinden kişi başı üretim adedi hesaplanarak kayıt altına alınmıştır. Kişi başı üretim adedi arttığı sürece kaynak birleştirme işlemi devam ettirilmiştir. Kişi başı üretim adedi artışı durduğu anda sistem besleme miktarı, ana modelin ürettiği adedin 3 katı yapıp, sistem tekrar çalıştırılmıştır. Kişi başı üretim adedinin artışı devam ettiği görüldüğünde model yeni isim ile kayıt edilmiştir; örneğin; M1A31L (Şekil 4.8, Şekil 4.9, Şekil 4.10). Kişi başı üretim adedi artışı sağlanmadığı anda algoritma durdurulmuştur.

Çizelge 4.8 Model 1 Alternatif akışlar algoritma sonuçları

	Çalışan Sayısı	Üretim	Kişi Başı Üretim
Model1	16	847	52,9
M11	31	1641	52,9
M12	30	1642,5	54,8
M13	29	1642,6	56,6
M1A1B	30	1483,5	49,5
M1A11	29	1482,6	51,1
M1A12	28	1461,8	52,2
M1A13	27	1447,6	53,6
M1A14	26	1447,5	55,7
M1A15	25	1392,5	55,7
M1A16	24	1378	57,4
M1A2B	30	1643,2	54,8
M1A21	29	1643,6	56,7
M1A21L	29	1703,7	58,7
M1A3B	30	1643,6	54,8
M1A31	29	1643,6	56,7

Çizelge 4.9 Model 2 alternatif akışlar algoritma sonuçları

	Çalışan Sayısı	Üretim	Kişi Başı Üretim
Model2	12	542,6	45,2
M21	21	962,3	45,8
M22	20	962,8	48,1
M23	19	961,5	50,6
M2A1B	22	952,3	43,3
M2A11	21	952,3	45,3
M2A12	20	949,5	47,5
M2A13	19	939,2	49,4
M2A14	18	938,9	52,2
M2A2B	22	953,5	43,3
M2A21	21	953,5	45,4
M2A22	20	949,7	47,5
M2A23	19	940,5	49,5
M2A24	18	939,7	52,2
M2A3B	22	953,9	43,4
M2A31	21	953,6	45,4
M2A32	20	952,2	47,6
M2A33	19	951,9	50,1

Çizelge 4.10 Model 3 alternatif akışlar algoritma sonuçları

	Çalışan Sayısı	Üretim	Kişi başı Üretim
Model3	10	500,7	50,1
M31	22	930,4	42,3
M32	21	924,2	44
M33	20	923,1	46,2
M34	19	902,9	47,5
M35	18	886,3	49,2
M3A1B	23	979,9	42,6
M3A11	22	979,6	44,5
M3A12	21	952,7	45,4
M3A13	20	952,7	47,6

5. TARTIŞMA VE SONUÇ

Projede incelenen modellerin çeşitli algoritmalar ile dengelenmesi sonucunda modele ait ortalama verimlilikler hesaplanmış ve Simul8 yazılımında oluşturulan model sonuçları ile karşılaştırılması yapılarak, yüksek verimliliğe sahip dengelemelerin uygulamaya geçirilmesi amaçlanmıştır.

Öncelikle bilinen farklı algoritmalarla mevcut modeller için uygulanabilecek hat dengeleme sonuçları incelenmiştir. Kullanılan algoritmaların işleyiş biçimlerine, çeşitli kaynaklardan ulaşılabileceği için burada genel açıklamalara yer verilmiştir.

- Önceden belirlenmiş üretim kapasitesine göre hat dengeleme algoritması: Bu yöntemde belirli bir üretim kapasitesinden yola çıkarak hat dengeleme yapılır. Gerekli işçi, makine sayısı ve makine çeşidi bulunup iş yüklemeleri yapılır.
- Sezgisel artırımı verimlilik yöntemiyle hat dengeleme algoritması: Bu yöntemde operasyonlar öncelik sırasına göre, verimliliğin %100 olduğu ya da verimliliğin düştüğü gözlemlenene kadar iş istasyonuna eklenir.
- Yalın üretime uygun hat dengeleme algoritmaları: bu kapsamda kullanılan toplamda 5 farklı algoritma mevcuttur. Algoritmanın çalışma şekli hepsinde aynıdır. Ancak tek fark operasyonların farklı şekillerde sıralanmalarıdır. Gerekli sıralanma oluşturulduktan sonra, operasyonlar uygun görülen çevrim süresi ve öncel operasyon kısıdına bağlı olarak atanırlar. 5 farklı sıralanma şekilleri şöyledir: en uzun işlem süresi, en kısa işlem süresi, konum ağırlığı, öncellerin çokluğu, öncellerin azlığı.

3 farklı giysi tipi için tüm algoritmalar çalıştırılmış ve elde edilen verimlilik değerleri Çizelge 5.1’de verilmiştir.

Çizelge 5.1 Algoritmalar ve Simul8 hat dengeleme verimlilikleri

Algoritmalar	Önceden belirlenmiş (%)	Sezgisel artırımı (%)	En uzun işlem süresi (%)	Öncellerin çokluğu (%)	Konum ağırlığı (%)	En kısa işlem süresi (%)	Öncellerin azlığı (%)	Simul 8 (%)
Model 1	85,00	96,00	74,14	74,14	74,14	74,14	74,14	84,00
Model 2	83,00	83,00	75,56	75,56	75,56	75,56	75,56	74,00
Model 3	90,00	90,00	87,46	87,46	87,46	65,60	65,60	90,00

Simul8 yazılımında oluşturulan ana modellerde ise genel bir verimlilik bulunmamaktadır. Her kaynak kendi verimliliğine sahip olup ayrıca operasyonlara ait kaynak verimlilikleri de

mevcuttur. Modellere ait kaynakların ortalama verimlilikleri hesaplanarak model denge verimliliği hesaplanmıştır (Çizelge 5.1).

Bu karşılaştırma sonucunda çeşitli algoritmaların simülasyona göre daha iyi verimlilik verdikleri görülmektedir. Ancak mevcut algoritmaların çalışması için operasyon sürelerinin sabit bir zamandan oluştuğu varsayımı kabul edilmektedir. Örneğin bir yan kapama dikişinin sürekli 23 sn’de yapıldığının kabul edilmesi gibi. Fakat, gerçek hayatta böyle bir uygulama olamayacağı felsefesi ile yola çıkıldığından dolayı operasyon sürelerinin istatistiksel dağılımlarla tanımlandığı simülasyon uygulamaları geliştirilmiştir. Tüm bu nedenlerden dolayı, belirleyici (deterministik) veri setiyle oluşturulan hat dengeleme yerine olasılıklı (stokastik) veri setinin kullanıldığı simülasyon modeliyle dengeleme çalışması yapmak çok daha rasyonel yaklaşımdır. Özetle, farklı algoritmalarla simülasyon sonuçlarını karşılaştırmak doğru değildir. Bu nedenden dolayı, simülasyon çalışmalarında kullanılmak üzere “kalp algoritması” geliştirilmiştir.

Bunun dışında işletmenin 3 farklı model için uyguladığı bir hat dengeleme çözümlemesi vardır. Ancak işletmenin teorik uygulamaları hatalı olduğu için burada ilgili çözümlemelere değinilmemiştir. Ayrıca işletmenin teorikle gerçekleşen arasındaki uyumsuzluktan dolayı bu çalışmaya başlanması karar verildiği için mevcut yalın üretim hattı modellenmiştir.

Simul8 yazılımında kullanılmak üzere geliştirilen “kalp algoritması” kurallarına uyularak oluşturulmuş ana modellere bakıldığında algoritmanın hat dengeleme üzerinde, kişi başına düşen üretim miktarı kriterinde ciddi iyileştirmeler sağladığı görülmüştür (Çizelge 5.2). İşletmedeki denge kurallarına göre kurulan Model 1’de kişi başı üretim adedi 53 iken, kalp algoritması kurallarına uygun kurulan M13’ de kişi başı üretim adedi 57’ye kadar artmıştır. Diğer modellerde de bu doğrultuda iyileştirmeler mevcuttur.

Ancak 3. Modele ait kişi başı üretim adedine baktığımızda yeterli seviye iyileşmenin gerçekleşmediği gözlemlenmiştir. Bu durumun nedeni %30 verimliliğe sahip bir kaynağın, önceki ve sonraki operasyonlara (operatörleri çok dolu olduğu için) atanamamasından kaynaklanmaktadır. Bu durumun üstesinden gelebilmek için “kalp algoritması”nın atama yapısında bazı kural esneklikleri getirilmesi gerekmektedir. Öte yandan bazı üretim sistemleri belirli çevrim süresinde daha verimli ve daha az kuyrukla çalışmaktadır. Fakat en uygun çevrim süresinin araştırılması başka bir çalışmanın konusudur.

Ancak aşağıda yer alan alternatif akışlara ait kişi başı üretim adetlerinde herhangi bir sıkıntı yaşanmaması, algoritmanın işe yararlılığını ispatlar niteliktedir.

Ana modeller üzerinde sağlanan iyileşmeler görüldükten sonra modellerdeki operasyon sıralamaları değiştirilerek alternatif akışlar oluşturulmuş, kişi başı üretim adedi daha fazla olan modellerin kurulması amaçlanmıştır. Oluşturulan alternatif akışlar Simul8 yazılımında “kalp algoritması” kurallarına uyularak kurulmuştur. Kalp algoritmasının hat dengeleme üzerindeki etkisi alternatif akışlarda da kişi başı üretim adedinin artışı şeklinde görülmüştür (Çizelge 5.3, Çizelge 5.4, Çizelge 5.5).

Çizelge 5.2 Model kalp algoritması sonuçları

Modeller	Çalışan Sayısı	Üretim	Kişi üretim başı
Model1	16	847	53
M11	31	1641	53
M12	30	1642,5	55
M13	29	1642,6	57
Model2	12	542,6	45
M21	21	962,3	46
M22	20	962,8	48
M23	19	961,5	51
Model3	10	500,7	50
M31	22	930,4	42
M32	21	924,2	44
M33	20	923,1	46
M34	19	902,9	48
M35	18	886,3	49

Çizelge 5.3 Model 1 alternatif akışlar kalp algoritması sonuçları

	Çalışan Sayısı	Üretim	Kişi Başı Üretim
M1A1B	30	1483,5	49
M1A11	29	1482,6	51
M1A12	28	1461,8	52
M1A13	27	1447,6	54
M1A14	26	1447,5	56
M1A15	25	1392,5	56
M1A16	24	1378	57
M1A2B	30	1643,2	55
M1A21	29	1643,6	57
M1A21L	29	1703,7	59
M1A3B	30	1643,6	55
M1A31	29	1643,6	57
M1A31L	29	1703,6	59
M1A4B	30	1634	54
M1A41	29	1643,6	57
M1A41L	29	1703,6	59

Çizelge 5.4 Model 2 alternatif akışlar kalp algoritması sonuçları

	Çalışan Sayısı	Üretim	Kişi Başı Üretim
M2A1B	22	952,3	43
M2A11	21	952,3	45
M2A12	20	949,5	47
M2A13	19	939,2	49
M2A14	18	938,9	52
M2A2B	22	953,5	43
M2A21	21	953,5	45
M2A22	20	949,7	47
M2A23	19	940,5	50
M2A24	18	939,7	52
M2A3B	22	953,9	43
M2A31	21	953,6	45
M2A32	20	952,2	48
M2A33	19	951,9	50

Çizelge 5.5 Model 3 alternatif akışlar kalp algoritması sonuçları

	Çalışan Sayısı	Üretim	Kişi Başı Üretim
M3A1B	23	979,9	43
M3A11	22	979,6	45
M3A12	21	952,7	45
M3A13	20	952,7	48

Sonuç olarak, çalışmada konfeksiyonda yalın üretim yapan bir konfeksiyon işletmesinin üretim sistemi simülasyon yardımıyla ayrıntılı olarak incelenmiş, daha verimli bir üretim için farklı alternatif modeller oluşturulmuştur. Simülasyon uygulamasında, hat dengelemede gerekli iyileştirmeyi sağlamak amacıyla yeni bir hareket algoritması oluşturulmuştur. Geliştirilen kalp algoritmasının hat dengeleme üzerinde son derece önemli iyileştirmeler yaptığı görülmüştür.

Tüm bunların yanı sıra, konfeksiyon üretiminde ürün temel özelliklerini etkilemeyecek şekilde belirli operasyonların sırası değiştirilerek, alternatif üretim modellerinin oluşturulmasının üretim verimliliğine olası etkileri de incelenmiş ve elde edilebilecek olan en iyi sonuçlar, operasyonlara atanacak uygun çalışan sayısı şeklinde aşağıda verilmiştir.

M1 mevcut iş akış diyagramında 16 kişi ile kişi başı 53 adet üretim yapar iken model 1'e ait M1A21L alternatif iş akış diyagramında kalp algoritması kurallarına uyularak yapılan dengelemede toplam 29 kişi ile kişi başı üretim 59 adede kadar yükselmiştir.

Çizelge 5.6 Model1 operasyonları için önerilen çalışan sayısı

M1	Kişi sayısı	M1A21L	Kişi sayısı
Sağ omuz çatma+Yaka biye	1	Sağ omuz çatma	1
Sağ omuz çatma+Sol omuz çatma	1	Yaka biye	2
Kol takma	1	Sol omuz çatma	2
Kal takma+Yan çatma	1	Kol takma	3
Yan çatma	1	Yan çatma	2
Ense biye	1	Ense biye	1
Yaka çima zikzak	1	Yaka çima zikzak	2
Ense biye kapama	1	Ense biye kapama	2
Etek+kol kırım	1	Etiket hazırlama	1
Kol kırım	1	Etiket takma	2
İş çevirme	1	Etek kırım	2
Etiket hazırlama	1	Kol kırım	2
Etiket takma	1	Kol kırım+iş çevirme	1
Ütü	2	Ütü	4
Kalite	1	Kalite	2

M2 mevcut iş akış diyagramında 12 kişi ile kişi başı 45 adet üretim yapar iken model 2'e ait M2A14 alternatif iş akış diyagramında kalp algoritması kurallarına uyularak yapılan dengelemede toplam 18 kişi ile kişi başı üretim 52 adede kadar yükselmiştir.

Çizelge 5.7 Model2 operasyonları için önerilen çalışan sayısı

M2	Kişi sayısı	M2A14	Kişi sayısı
Sağ+Sol omuz çatma+Yaka biye	1	Sağ+Sol omuz çatma+Yaka biye	1
Kol takma	1	Ense biye	1
Ense biye+Askılık takma	1	Ense biye kapama+Etiket takma	1
Ense biye kapama+Etiket takma+Yaka zikzak	1	Askılık tutturma	1
Alt üst beden birleştirme	1	Yaka zikzak	1
Yan çatma	2	Alt üst beden birleştirme	2
Etek kırım	1	Kol takma	2
Kol kırım+İş çevirme	1	Yan çatma	2
Ütü	2	Kol kırım	1
Kalite	1	Kol+Etek kırım	1
		Etek kırım	1
		Ütü+İş çevirme	2
		Ütü	1
		Kalite	1

M3 mevcut iş akış diyagramında 10 kişi ile kişi başı 50 adet üretim yapar iken, model 3'e ait M3A13 alternatif iş akış diyagramında kalp algoritması kurallarına uyularak yapılan dengelemede toplam 20 kişi ile kişi başı üretim 48 adet olarak kalmıştır. Bunun nedeni tartışma ve sonuç bölümü içerisinde anlatılmıştır.

Çizelge 5.8 Model3 operasyonları için önerilen çalışan sayısı

M3	Kişi sayısı	M3A13	Kişi sayısı
Ön arka ağ+Yan çatma	3	Ön arka ağ+Yan çatma	5
Yan panel recme	1	Yan panel recme	2
İç ağ çatma+Ağ zikzak	1	İç ağ çatma	2
Kemer tutturma+Hazırlama+İşaret	1	Kemer tutturma+ağ zikzak	1
Kemer montaj+Etiket takma	1	Kemer hazırlama	1
Paça kırım+İş çevirme	1	Kemer hazırlama+İşaret	1
Ütü	1	Paça kırım	1
Kalite	1	Kemer montaj	2
		Etiket takma+İş çevirme	1
		Ütü	2
		Kalite	2

Gelecekte bu konu üzerinde çalışacak araştırmacılara aşağıdaki tavsiyeler verilmektedir;

- Algoritma, makine cinslerine göre kaynak atamaları yapılacak şekilde geliştirilmelidir.
- Algoritmanın bir önceki veya bir sonraki çalışana iş yüklemeye çalışması esnetilmelidir.
- Algoritma, farklı ürün grupları için uygulanmalıdır.

6. KAYNAKLAR

- Argüden Y (2006). Strateji ve Simülasyon Modelleri.
- Banks J ve Carson JS (1984). Discrete-Event System Simulation, Prentice-Hall, Inc.
- Banks J (2000). Introduction to simulation, Proceedings of the 2000 Winter Simulation Conference, p 8.
- Carrie A (1988). Simulation of Manufacturing Systems, John Wiley and Sons Ltd.
- Chung AC (2004). Simulation modeling handbook: a practical approach, CRC Press, New York.
- Cocks S, Harlock S (1989). Computer-aided Simulation of Production in the Sewing Room of a Clothing Factory, J. Text. Inst., Vol.80, 455-463.
- Fozzard G, Spragg J and Tyler D (1996). Simulation of flow lines in clothing manufacture:Part 1:model construction, Int. J. Of Clothing Science and Technology, Vol.8, 17-27.
- Gordon G (1978). System Simulation, Prentice Hall Incorporation, Englewood Cliff,, New Jersey.
- Güner M, Ünal C (2008). Line Balancing in the Apparel Industry Using Simulation Tecniques, Fibres&Textiles in Eastern Europe, Vol.16, 75-78.
- Gürkan P (2004). Dokuma İşletmelerinde Simülasyon Modelleme Tekniğinin Uygulanması, Ege Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, İzmir.
- Hiller FS, Lieberman GJ (1980). Introduction To Operations Research, 3rd Ed, Holden-Day.
- Hu A, San Y and Wang Z (2001). Verifying and validating a simulation model, Proceedings of the 2001 Winter Simulation Conference, p 5.
- Ingalls R (2001). Introduction to simulation, Proceedings of the 2001 Winter Simulation Conference, p 10.
- Kurşun S, Kalaoğlu F (2009). Simulation of Production Line in Apparel Manufacturing, Fibres&Textiles in Eastern Europe, Vol.17, 68-71.
- Kurşun S, Dinçmen M, Kalaoğlu F (2009). Production Line Modelling in Clothing Industry by Simulation, Tekstil, Vol.58, 186-195.
- Kurşun S, Kalaoğlu F (2010). Line Balancing by Simulation in a Sewing Line, Tekstil ve Konfeksiyon, 3: 257-261.
- Kuş P (2003). Simülasyon uygulamaları, www.odevsitem.com, (erişim tarihi,05.04.2014)
- Law M, Kelton WD (2000). Simulation Modeling and Analysis, McGraw-Hill NewYork.

- McMullen PR, Frazier GV (1998). Using Simulated Annealing to Solve a Multiobjective Assembly Line Balancing Problem With Parallel Workstations, International Journal of Production Research, No. 36, 2717-2741
- Musselman KJ (1998). Guidelines for Success, in Handbook of Simulation, New York.
- Pegden DC, Shannon ER, and Sadowski PR (1995). Introduction To Simulation Using Siman, McGraw-Hill, New York.
- Pritsker AAB (1995). Introduction to Simulation and SLAM II, 4th ed., Wiley&Sons, New York.
- Rajakumar S, Arunachalam V and Selladurai V (2005). Simulation of workflow balancing in assembly shopfloor operations, J of Manufacturing Technology Management, Vol.16, 265-281.
- Sarıcam C, Kalaoğlu F (2007). Analysis of Modular Manufacturing System in Clothing Industry by Using Simulation, Fibres &Textile in Eastern Europe, Vol.15, 93-96.
- Shalliker J, Ricketts C (2005). An Introduction to SIMUL8 2005, Release 12, University of Plymouth.
- Shannon RE (1975). Systems Simulation:The Art and Science, Prentice-Hall, Englewood Cliffs, NJ.
- Shannon RE, Long SS, Buckles BP (1981).Operation Research Methodologies in Industrial Engineering:ASurvey, Vol.12, 364-367.
- Shannon RE (1998). Introduction to the Art and Science of Simulation. in Proceedings of the Winter Simulation Conference, 7-14, Washington.
- Ünal C, Tunalı S, Güner M (2009). Evaluation of Alternative Line Configurations in Apparel Industry Using Simulation, Textile Research Journal, Vol.79, 908-916
- Watson HJ, Blackstone JH (1989). Computer Simulation,Reading, Massachusetts, Addison.
- Womack JP (1997). What is Lean? Lean Interprise Institute, www.lean.org, (erişim tarihi,08.08.2014)
- Varol A (1999). Benzetim (simülasyon), www.listweb.bilkent.edu.tr, (erişim tarihi,03.02.2014)
- Zielinski J, Czacherska M (2004). Optimisation of the Work of a sewing team by using simulation, Fibres&Textiles in Eastern Europe, Vol.12, 78-83.

www.simul8.com

ÖZGEÇMİŞ

07.07.1987 tarihinde İstanbul'da doğan Selin BİLGET, İlk,orta ve lise öğrenimini Tekirdağ'da tamamladıktan sonra, 2006 yılında Uşak Üniversitesi Tekstil Mühendisliği bölümünü kazanarak yükseköğrenimine başlamıştır. Tekstil Mühendisliği bölümünden mezun olduktan sonra, 2011 yılında yüksek lisans yapmak üzere, Namık Kemal Üniversitesi Fen Bilimleri Enstitüsü Tekstil Mühendisliği anabilim dalına başlamıştır. Şu anda özel bir tekstil firmasında iş hayatına devam etmektedir.