

**UZMAN SİSTEMLERİN
ÜRETİM PROGRAMI İZLENMESİNDE
UYGULAMASI**

Mujgan SAĞIR

**Anadolu Üniversitesi
Merkez Kütüphane**

**Yüksek Lisans Tezi
Endüstri Mühendisliği Anabilim Dalı
Yöneylem Araştırması Bilim Dalı**

OCAK - 1992

Mujgan SAĞIR'ın YÜKSEK LİSANS tezi olarak hazırladığı
"Uzman Sistemlerin Üretim Programı İzlenmesinde Uygulaması"
başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin
ilgili maddeleri uyarınca değerlendirilerek kabul
edilmiştir.

08.01.1992

Uye : Prof. Dr. İmdat KARA

Uye : Doç. Dr. Nihat YUZUGULLU

Danışman : Doç. Dr. Ali GÜNEŞ

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
gün ve 3.00-7 sayılı kararıyla onaylanmıştır.

Prof. Dr. Rüstem KAYA

Enstitu Muduru

TEŐEKKUR

alıőmamda bana yardımcı olan hocam Sayın Prof. Dr. İmdat KARA' ya, danışman hocam Sayın Do.Dr. Ali GUNEŐ'e ve uygulamanın yapıldığı iőletmenin Uretim Hizmetleri Bölümü Uretim Planlama Őefi Sayın Munevver UZTURK'e teőekkür ederim.

ÖZET

Uretim işletmelerinde faaliyetler yürütülürken, uretime yönelik planlara uyulması amaçlanır. Talebin zamanında karşılanması, kar elde edilmesi ve işletmenin varlığını ve pazardaki payını sürdürebilmesi, yapılan bu planların gerçekleşmesi oranında mümkündür. Bu nedenlerle işletmeler, üretim sürecini izleyerek üretimin planlara uygun olarak gerçekleşip, gerçekleşmediğini kontrol etmek isterler. Üretimi izlemek, üretim sürecinden veri almayı gerektirir.

Bu çalışmada yukarıda sözü geçen üretimi izleme faaliyetlerinin, bir üretim işletmesinde uzman sistem yaklaşımıyla örneklenmesi amaçlanmıştır. Uzman sistem kavramını incelemenin yanısıra uzman sistemlerin Endüstri Mühendisliği ve Yöneylem Araştırması ile olan etkileşimlerine de yer verilmiştir.

Mevcut sistemde türetilen pek çok raporu süreçle ilgili bilgi ve tecrübelerini kullanarak her gün ayrı ayrı değerlendiren ve bazı sonuçlara varan uzman kişinin, bu işi yaparken izlediği mantığın bir bilgisayar programına aktarılmasına çalışılmıştır. Bu çalışma için uzman sistem geliştirme dillerinden birisi olan Prolog dili kullanılmıştır.

Anahtar kelimeler

Uzman sistem

Uretim izleme

Çıkartım mekanizması

Bilgi tabanı

S U M M A R Y

In the factories it is desired to follow predefined plans which are about production. To satisfy the demands, make profits and survival of the business and continuation of its share in the markets depend on the ratio of the application of those plans. Because of these reasons, factories want to monitor the production process and control whether the production is made according to plans or not. Production monitoring requires taking data from production process.

In this thesis, it is desired to monitor the production process using expert systems approach. In addition to examine the expert systems approach, interactions of expert systems with Industrial Engineering and Operational Research are included.

A computer program is written to simulate an expert person who monitors the process, evaluates the data and reaches some conclusions on daily basis depend on his knowledge and experiences. For this study one of the expert systems developing language, namely Prolog is used.

Key Words

Expert system

Production monitoring

Inference engine

Knowledge base

IÇİNDEKİLER

	<u>Sayfa</u>
UZET	iv
SUMMARY	v
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	x
1. GİRİŞ	1
2. UZMAN SİSTEMLERİN ENDÜSTRİ MUHENDİSLİĞİ VE YÖNEYLEM ARAŞTIRMASI İLE İLİŞKİLERİ.....	4
2.1. Uzman Sistemler.....	4
2.1.1. Uzman sistem tanımı ve temel bileşenleri..	5
2.1.2. Uzman sistemlerin klasik bilgisayar yazılımlarından farkları.....	10
2.1.3. Uzman sistem geliştirme yöntemi.....	13
2.1.4. Uzman sistem uygulamaları.....	16
2.2. Yöneylem Araştırması ve Uzman Sistemler.....	18
2.3. Endüstri Muhendisliği ve Uzman Sistemler.....	20
2.3.1. Uzman sistem ile Endüstri Muhendisliği etkileşimi.....	21
2.3.2. Endüstri Muhendisliği alanlarında uzman sistem uygulamaları.....	24

İÇİNDEKİLER (devam)

2.4. Endüstri Mühendisliği, Yöneylem Araştırması ve Uzman Sistem Uzantıları, Mevcut Durum Işığında Eğilim.....	27
3. BİR ÜRETİM İŞLETMESİNDE ÜRETİM PROGRAMINDAN SAPMALARIN UZMAN SİSTEM YAKLAŞIMIYLA İZLENMESİ.....	29
3.1. İşletmenin Üretim Süreci.....	29
3.2. Üretimi İzlemede Uygulanmakta Olan Yaklaşım	30
3.3. Üretim Programından Sapma Nedenlerini Bulmak İçin Uzman Sistem Yaklaşımının Uygulanabilirliği.....	31
3.4. Sistemin Geliştirilmesi.....	32
3.4.1. Belirleme.....	32
3.4.2. Kavramsallaştırma.....	33
3.4.3. Biçimselleştirme.....	37
3.4.4. Tasarım ve Gerçekleştirme (Geliştirilen Sistemin Yapısı).....	40
3.4.4.1. Programda kullanılan değişken ve veri tabanları.....	44
3.4.4.2. Programın modülleri, çalışma şekli...	45
3.4.5. Test Etme ve Değerlendirme.....	54
3.4.5.1. Programın gerçek verilerle çalıştırılması.....	54
3.4.5.2. Değerlendirme.....	59
3.4.6. Yaşatma ve bakım planlama.....	66
3.5. Geliştirilen Sistemin Değerlendirilmesi.....	67

İÇİNDEKİLER (devam)

4. SONUÇLAR VE ÖNERİLER	69
KAYNAKLAR DİZİNİ	74

EKLER

1. Veri Tabanları ve İçerdikleri Değişkenler
2. Programa Ait Menüler
3. Geliştirilen Bilgisayar Programı

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1.(a). Uzman sistemin temel yapısı.....	8
2.1.(b). Uzman sistemin temel yapısı.....	9
3.1. İşletmedeki Uretim süreci.....	34
3.2. Geliştirilen sistemin mantıksal akış şeması.....	41
3.3. TEZ adlı programın akış şeması.....	51
3.4. TEZ2 adlı programın akış şeması.....	53

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1.(a). 6280 kod numaralı mamulün beyaz rengi için gerçek veriler.....	58
3.1.(b). 6280 kod numaralı mamulün beyaz rengi için gerçek veriler.....	59
3.2. Günlük üretim değerlendirmesi için örnek veriler.....	65

1. GİRİŞ

Bilgisayar teknolojisi son yıllarda büyük bir hızla gelişmiş, veriyi saklama, veriye erişme, kullanma gibi konularda hızlı ve etkin olması yönüyle Üretime yönelik faaliyetlerde de bilgisayar kullanımı yaygın hale gelmiştir.

İşletmelerde Üretim çevrimi boyunca elde edilen veriler daha sonra alınacak kararlarda gözönünde bulundurulup, Üretimin değerlendirilmesinde de kullanılmaktadır. Bu verileri saklama ve gerektiğinde saklanan verilere erişme işlemleri önceleri klasik yöntemlerle yapılmakta iken artık bilgisayar aracılığıyla bu gibi işler daha etkin bir şekilde yürütülmektedir.

Bu çalışmada, bilgisayar teknolojisindeki gelişmeye paralel olarak oluşan Uzman Sistemler ele alınarak, endüstri mühendislerinin sorumluluk alabileceği en önemli alanlardan olan üretime yönelik faaliyetler içerisinde, Üretimi izlemek için bir uzman sistem geliştirmek, ayrıca uzman sistemlerin, Endüstri Mühendisliği ve Yöneylem Araştırması ile olan ilişkilerini örneklemek amaçlanmıştır.

İkinci bölümde, uzman sistemler konusunda yapılmış çeşitli tanımlar ele alınarak, bunlar tartışılmış, daha sonra bir uzman sistemde olması gereken bileşenler, uzman sistem geliştirme yöntemi, uzman sistemlerin klasik yazılımlara göre ne gibi farklılıkları olduğu ve uzman sistem uygulamaları üzerinde durulmuştur. Ayrıca bu çalışmaya esas olan ve

Yöneylem Araştırması, Uzman Sistemler, Endüstri Mühendisliği etkileşimlerine yönelik açıklamalar da bu bölümde yer almaktadır.

Üçüncü bölümde, uygulama amacıyla seçilen bir işletmede, üretim programından sapmaları izlemeye yönelik bir uzman sistemin; uzman sistem geliştirmek için açıklanıp benimsenen yöntemi kullanarak geliştirilmesi için yapılan çalışmalar açıklanmıştır. Önce çalışmanın yapıldığı işletmedeki üretim süreci, mevcut durumda işletmenin üretimi nasıl izlediği, uzman sistem yaklaşımının böyle bir işletmede uygulanabilirliği gibi konular üzerinde durulmuş, daha sonra uzman sistemi geliştirmek için benimsenen yöntemdeki adımların uygulanmasına yönelik çalışmaya ve en sonda da geliştirilen sistemin değerlendirilmesine yer verilmiştir.

Çalışmanın uygulama kısmında, gerçek sistem tümüyle ele alınamamıştır. Gerçek sistemde aslında bir mamul birden fazla renkte üretilebilmektedir. Bu çalışmada geliştirilen sistem ise birden fazla mamul fakat tek bir renk çeşidi için tasarlanmıştır. Örnek uygulama amacıyla toplam üretimin yaklaşık % 60' ını oluşturan beş mamul ve bunların sadece beyaz rengi alınmıştır. Beyaz rengin diğer renkler içindeki kullanım oranı % 70 civarındadır ve en çok kullanılan renktir.

İşletmede herhangi bir ay mamuller hem stok için hem de ayın belirli bir gününe yetişecek özel bir sipariş için üretilebilir. Bu durumda günlük üretimler bu iki partiyi de karşılayacak şekilde planlanmaktadır. Geliştirilen sistemde bu iki durumdan yalnızca birisinin olacağı varsayılmıştır ve belli bir tarihe kadar istenen miktarlarda üretim sözkonusudur. Bu tarih ya bir termin tarihidir ya da normal aylık planlama döneminin son tarihi olabilir.

Çalışmada üretimi izleme kavramı, kontrol etme kavramından farklı bir anlamda ele alınmıştır. İzleme, daha sınırlı ve sisteme müdahalenin yapılmadığı bir durumdur. Oysa kontrol, gerektiği durumlarda planları günclemeyi ve değiştirmeyi de içerebilir. Kontrol için mevcut durumu bilmenin gerektiği düşünülürse, izleme ile yapılan işin de bir anlamda, bu mevcut durumu bilmeye yönelik çalışmalara karşı geldiğini düşünmek yanlış olmayacaktır. Bu açıklamalara dayanarak, çalışmanın, üretim faaliyetleri içerisindeki yeri olarak, üretim planlama ve kontrolü çevriminin içinde ve hatta çevrimin başında yer aldığı söylenebilir. Sisteme gerekli müdahaleleri yapmak, çevrimin diğer aşamaları olarak düşünülebilir. Üretim kontrolü amacıyla geliştirilmiş teknikler vardır. Fakat kaynaklarda özellikle üretimi izleme konusuna yönelik ayrıntılara rastlanamamıştır. Bu konuya yönelik olarak, üretimi izleme yönteminin, sahip olunan üretim sürecinin kendine özgü özelliklerine göre değişeceği ve bağlı olarak çalışmada geliştirilen sistemin, seramik turu eşya üreten süreçlerde üretimi izlemek amacıyla kullanılabilirliği söylenebilir.

Dördüncü ve son bölüm sonuçlar ve öneriler bölümüdür. Bu bölümde çalışmanın sonuçları ve yapılabilecek diğer çalışmalardan bahsedilmiştir.

2. UZMAN SİSTEMLERİN ENDÜSTRİ MUHENDİSLİĞİ VE YÖNEYLEM ARAŞTIRMASI İLE İLİŞKİLERİ

Bu bölümde, Uzman Sistemlerle, Endüstri Muhendisliği ve Yöneylem Araştırması arasındaki etkileşimleri incelemek amaçlanmıştır. Bu amaçla önce uzman sistem kavramı üzerinde durulmuş, daha sonra, sırasıyla Uzman Sistem / Endüstri Muhendisliği, Uzman Sistem / Yöneylem Araştırması ve Uzman Sistem / Yöneylem Araştırması / Endüstri Muhendisliği etkileşimleri ele alınmıştır.

2.1. Uzman Sistemler

Uzman sistemler, yapay zeka bilim dalının bir alt dalı olarak ortaya çıkmıştır. Yapay Zeka'nın robotlar ve doğal dili çözümlenmeye ilişkin başka çalışma alanları da olmasına karşın, özellikle, uzman sistemler konusu geniş bir ilgi görmüş ve bu konudaki çalışmalar önemli bir yoğunluk kazanmıştır. Özellikle, karmaşık problemleri sezgisel yöntemlerle çözüp, "eniye (optimum)" yerine "eniye yakın" çözümleri tercih ederek, kolay işletilir ve anlaşılır sistemlerle çalışmak isteyen kişiler uzman sistem yaklaşımına yönelmişlerdir (Daglı, 1986). Sezgisel yöntemlere de yer verip, karar verme sürecinin etkinliğini arttırmaya olanak veren uzman sistemler, bu nedenle, bu

konuyla yakından ilgili olan endüstri mühendisleri için de Endüstri Mühendisliği ve Yöneylem Araştırması çalışmalarında kullanabilecekleri bir araç olmuştur.

2.1.1. Uzman sistem tanımı ve temel bileşenleri

Diğer bilim dallarına göre daha yeni olması nedeniyle, uzman sistemlerle ilgili olarak değişik kaynaklarda, farklı tanımlara rastlamak mümkündür. Bu nedenle aşağıda bir kaç farklı uzman sistem tanımına yer verilmiştir:

"Uzman sistem, bir insan uzmanın davranışını taklit eden programlardır " (Schildt, 1987).

"Uzman sistem, önemli derecede uzman kişinin bilgisine ihtiyaç duyulan problemleri çözmek için, bilgiyi ve sonuç çıkarma mekanizmasını kullanan bir bilgisayar programıdır " (Gupta and Chin, 1989).

"Uzman sistemler, özel birtakım problemlerin çözümünde, uzmanların bilgisini ve uygulama sürecini taklit etmeyi amaçlayan, danışman bilgisayar programlarıdır " (Turban, 1990).

Yukarıdaki tanımlar birlikte ele alındığında; uzman sistem kavramıyla, esas itibarıyla bir uzmanın davranışını bilgisayar desteği ile taklit etmenin amaçlandığı anlaşılmaktadır. Uzman sistemin en belirgin özelliği olan taklit etme yeteneğini açıkça belirtmesi ve son yıllarda yapılmış bir tanım olması nedenleriyle, bu çalışmada, Turban'ın tanımı esas alınmıştır.

Uzman sistemin temel bileşenleri, o uzman sistemi oluşturan temel elemanlardan ibarettir. Farklı uzman sistem tanımlamaları olduğu gibi, uzman sistemin bileşenleri için

de farklı açıklamalar yapılmıştır. Aşağıda bir uzman sistemde olması gereken bileşenler için çeşitli kaynaklardaki açıklamalara yer verilmiştir:

"Bir uzman sistemi, bilgi tabanı, çalışma belleği ve çıkartım mekanizması oluşturur (Suraj, 1985) ".

"Uzman sistemler Uç ana parçadan oluşur: Bilgi tabanı, muhakeme motoru, kullanıcı bağlantısı (Dağlı, 1986) ".

İlk açıklamada yer alan çıkartım mekanizması, ikincisindeki muhakeme motoruna karşı gelmektedir. Her ikisi de mevcut duruma uygun kuralları bilgi tabanından bularak, karşı gelen sonuçları veren alt birimi temsil etmektedir. Benzer şekilde çalışma belleği ile de kullanıcı bağlantısı eş anlamdadır. Bu birimler sistemin kullanıcı ile iletişimini sağlayan bileşeni temsil etmektedirler. O halde ilk bakışta farklı açıklamalar gibi görülse de, bu iki araştırmacının belirttiği bileşenler aynı anlamdadır,

İzleyen yıllarda yapılmış diğer bir açıklama şu şekildedir:

"Bir uzman sistemin temel bileşenleri, bilgi tabanı ve çıkartım mekanizmasıdır (Watts and Eldin, 1987)".

Görüldüğü gibi Watts ve Eldin tarafından, yukarıda verilen ilk iki açıklamada "çalışma belleği" ya da "kullanıcı bağlantısı" olarak yer alan bileşenden söz edilmemektedir.

Hücre tipi üretim yapan sistemlerin kontroluna yönelik olan bir çalışmada, uzman sistemin; bilgi tabanı, çıkartım mekanizması ve çalışma belleği olmak üzere üç bileşeni

olduđu öne sürülmüştür (Teng and Black, 1989). Bu açıklamada yer alan bileşenlerin de, ilk iki açıklamadakilere eşlenebileceği görülmektedir.

Bileşenleri daha ayrıntılı ele alarak, konunun gerek anlaşılması gerekse uygulamasına daha fazla açıklık getiren Turban, bileşenleri aşağıdaki başlıklarda açıklamıştır (Turban, 1990):

- Bilgi kazanma birimi,
- Bilgi tabanı,
- Çıkartım mekanizması,
- Çalışma ekranı,
- Kullanıcı arayuzu,
- Açıklama birimi,
- Bilgi arıtma sistemi .

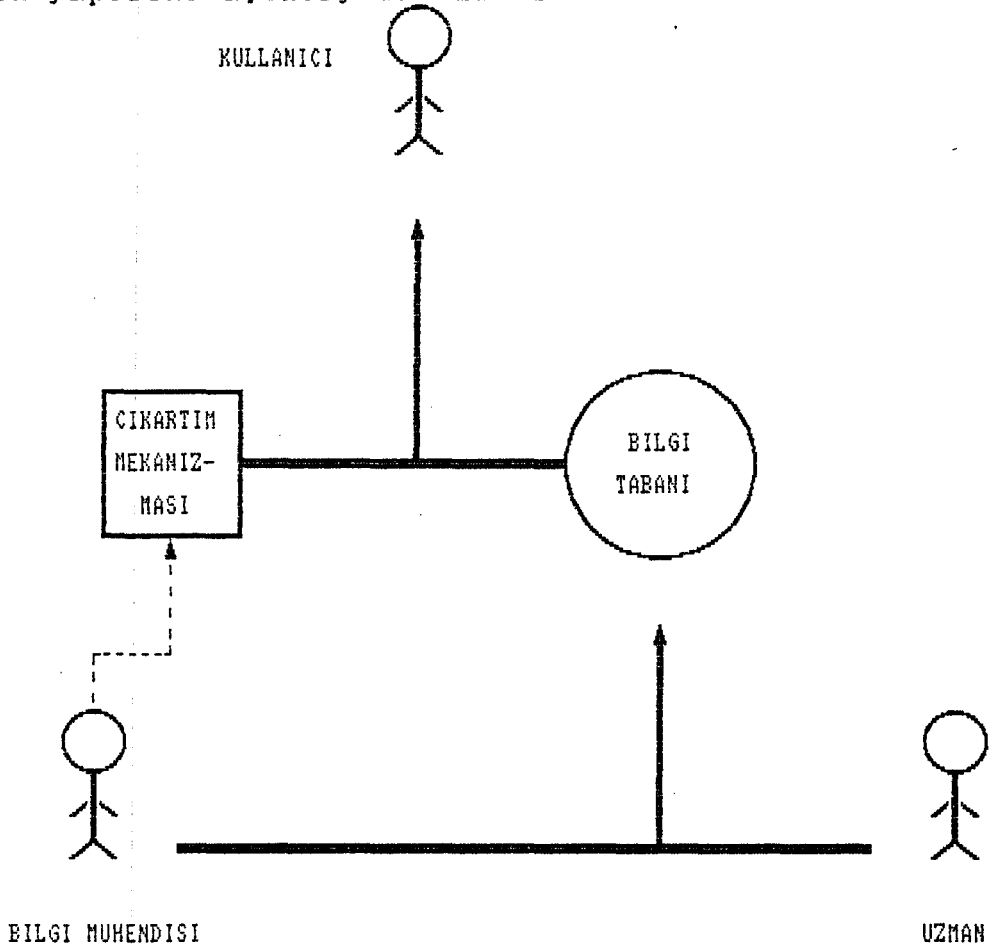
Son bileşen grubu diğerlerine göre daha çok sayıda elemanı içermektedir. Bu grupta olupta önceki tanımlamalarda yer almayan bileşenler, sistemin etkinliğini arttırıcı yan bileşenlerdir.

Yukarıdaki açıklamaların tümünde ortak olan bileşenler; bir bilgi tabanı ve bir çıkartım mekanizmasıdır. Bunlara ek olarak, sistemin kullanıcıyla iletişimini sağlayacak olan bir iletişim birimine ihtiyaç duyulacağı gerekçesiyle, önceki açıklamalarda " çalışma belleği ", " kullanıcı bağlantısı " ya da " kullanıcı arayuzu " isimleriyle yer alan bileşen de uzman sistemin temel bileşenidir. Waterman, uzman sistemlerde, çalışma belleği, kullanıcı bağlantısı veya kullanıcı arayuzu fonksiyonunu üstlenen bileşeni

"diyalog yapısı" olarak ifade etmektedir (Gupta and Chin, 1989) ki, farklı kavramlar yerine, belirtilen fonksiyonlara karşılık olarak diyalog yapısını kullanmak yerinde olacaktır.

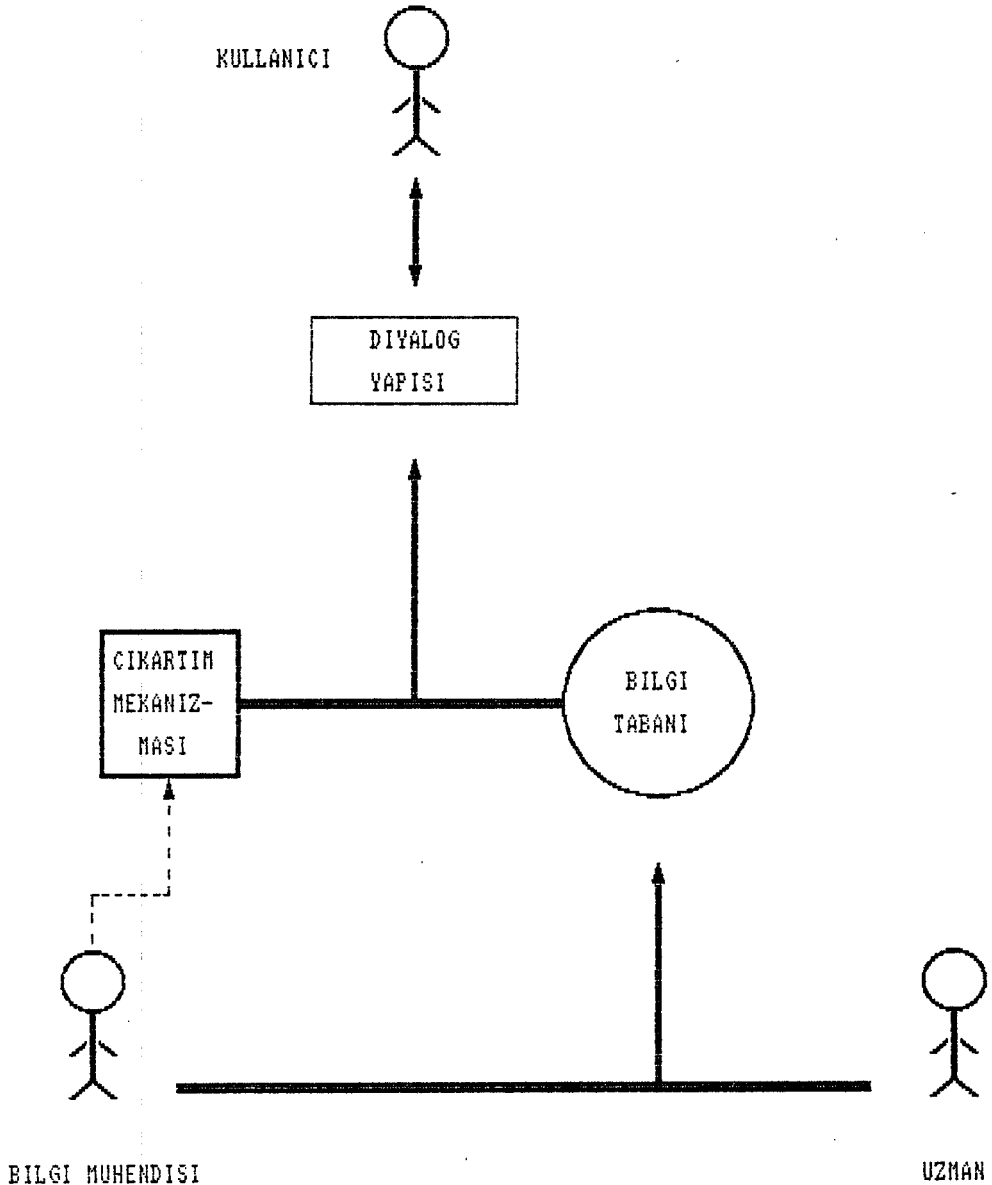
Böylece, farklı uzman sistemlerin farklı bileşenleri içermesi mümkün olmakla birlikte; bir uzman sistemde olması gereken temel bileşenler olarak bilgi tabanı, çıkartım mekanizması ve diyalog yapısı esas alınmıştır.

Bir uzman sistemde olması gereken bileşenleri bilmek, sistem hakkında yeterince bilgi sahibi olmayı sağlamayacaktır. Bu bileşenleri belirtmenin yanısıra, bileşenler arası ilişkilerin, bilgi alışverişlerinin, ayrıca her bir bileşenin ne gibi özelliklere sahip olduğunun da bilinmesi gerekmektedir. Butun bu özellikler sistemin yapısını açıklayıcı özelliklerdir.



Şekil.2.1.(a). Uzman sistemin temel yapısı

Şekil.2.1.(a)., uzman sistemin temel yapısını göstermektedir. Bu şekil temel bileşenlerden çıkartım mekanizması ve bilgi tabanını içermekte olup, şekilde, kullanıcı ile sistem arasındaki iletişime yönelik bir ara birimin olmayışının eksikliği hissedilmektedir. Bu eksikliği gidermek için, daha önce diyalog yapısı olarak adlandırılan etkileşim birimi, üçüncü temel bileşeni temsilen şekle dahil edildiğinde, Şekil.2.1.(b).’deki yapı oluşur.



Şekil.2.1.(b). Uzman sistemin temel yapısı

Şekil.2.1.(b).’ ye göre, batı kaynaklarında bilgi mühendisi olarak nitelenen analiz ve tasarımcının, ilgili konudaki uzman(lar) ile yaptığı görüşmeler ve bilgi alışverişleri sonucunda elde ettiği bilgiler bilgi tabanını oluşturmaktadır. Bilgiyi yorumlayarak, bir sonuca ulaşmak için kullanan sonuç çıkarma mekanizması da yine bilgi mühendisi tarafından oluşturulmaktadır. Sistemin kullanımı sırasında bilgi tabanı ve sonuç çıkarma mekanizması arasında bir bilgi alışverişi olmaktadır. Kullanıcı tarafından sistemin kullanılmasına izin veren ve iletişimi sağlayan bileşen ise diyalog yapısıdır.

2.1.2. Uzman sistemlerin klasik bilgisayar yazılımlarından farkları

Yakın geçmişte geliştirilmiş olmaları ve bilgisayara dayalı bulunmaları nedeniyle, uzman sistemlerle klasik bilgisayar yazılımları arasındaki farklara ilişkin doyurucu açıklamalara rastlanamamaktadır. Her ne kadar, uzman sistem bilgiyi, klasik program veriyi işler (Ulengin, 1987), gibi görüşler ileri sürülse de, bunlar arasındaki farkların çok daha genelde ele alınıp, ayrıntılarıyla belirlenmesinde yarar vardır. Bu yönlü farklılık göstergelerinin türetilmesi, en azından, bu alandaki çalışmalarını, klasik bilgisayar yazılımlarından ayırt etmeye, yanısıra uzman sistemler konusunun gelişimini sağlamaya olanak verecektir.

Belirtilen düşünceler doğrultusunda, uzman sistemlerle klasik yazılımlar arasındaki farklılıklar aşağıdaki başlıklarda toplanabilir:

- . Analiz ve Gerçekleştirmede Bağımlılık
- . Karar Verme Sürecindeki Yeri

- . Yürütmedeki Katkı
- . Yapısal Özellik
- . Yaşatma ve Bakım

Yukarıda sıralanan farklılık göstergeleri şöyle açıklanabilir:

Analiz ve Gerçekleştirmede Bağımlılık: Önceki kısımlarda verilen uzman sistem tanımlarından anlaşıldığı gibi, bir uzman sistemde esas olan, uzman kişinin davranış biçimini taklit etmektir. Bunu yapabilmek için uzman kişiyi analiz etmek gerekmektedir. Bu noktada, o kişinin ilgilenilen konudaki tüm bilgisi, bu bilgileri ne zaman, ne şekilde ve hangi amaçlarla kullandığı, karşılaştığı problemleri çözerken nasıl bir yol izlediği, ayrıntılı olarak ele alınır. Klasik yazılımlarda ise, esas olan iştir. Bu nedenle, program tarafından yapılması istenen iş analiz edilir.

Karar Verme Sürecindeki Yeri: Uzman sistemde, karar verme sürecine, iş ve çevrenin özellikleri doğrultusunda, doğrudan destek verilerek, çoğu kararların sistemden alınmasına çalışılır. Klasik yazılımlar ise, karar vericiye veri-bilgi üreterek destek olurlar.

Yürütmedeki Katkı: Bir uzman sistemin görevi, işin akışında karşılaşılan karar problemlerinde uzmana olan bağımlılığı azaltmaktır. Bunun sağlanabilmesi için, sisteme, uzman kişinin hangi koşullar altında hangi tip davranışlarda bulunduğuna yönelik bilgiler girilmelidir. Böylece, benzeri koşullarla karşılaşıldığında sistem, nasıl davranılması gerektiğini bilecek ve uzman kişi olmadan, olabildiğince uzmaninkine yakın kararlar alabilecektir. Bu şekilde uzmana olan bağımlılık azaltılacaktır.

Yapısal Özellik: Gerçek hayat problemlerinde, her şeyi sisteme dahil etmek mümkün değildir. Bunun nedeni, bazı kavramları sembollerle ifade etmenin çok güç olması ya da mümkün olamamasıdır. Oysa gözönünde bulundurulamayan bu etkenler, sistemi etkilemektedirler. Sonuçta, sistemi etkiledikleri halde sisteme dahil edilemeyen bu tür değişkenler, sistemden alınan sonuçların ve verilen kararların etkinliğinin azalmasına yol açmaktadır. Bu kısıt daha çok klasik yazılımlarda bir eksiklik olarak ortaya çıkmaktadır. Çünkü klasik yazılımlar, bir problemin çözümüne ilişkin herkeze kabul edilmiş, genel bir çözüm yolunu esas alırlar. Bu yol, probleme ilişkin özel bazı durumları, sezgisel yaklaşımları içermez. Bir başka deyişle, klasik yazılımlar algoritmiktir. Uzman sistemler ise, sadece kitaplardan veya makalelerden elde edilen bilgileri değil, tecrübe ile elde edilen bilgileri de içermektedir. Bu durum, ilgili sistemi oluştururken esas alınan uzman kişinin bu bilgiye sahip olmasıyla mümkün olmaktadır. Çünkü uzman kişiyi diğerlerinden ayıran özellik; kişiye mahsus, herkes tarafından bilinmeyen, üzerinde genel bir anlaşma olmayan ve çoğunlukla kişilerin deneyimlerine dayanan bilgiye sahip olmasıdır. Bu nedenle uzman sistemlerde sezgisel yöntemler önemli yer tutmaktadır.

Yaşatma ve Bakım: Klasik yazılımlar, daha çok, tekrarlayan işlerin yapılmasında kullanılmaktadırlar. Bu tür programlar çok karmaşık bir iş için yazılacak olsalar bile, o konuyla ilgili gerekli bilgileri bir kere elde edip programa dahil etmek yeterlidir. Programın yürütülmesinde artık uzmana gerek yoktur. Bir uzman sistem ise, bu şekilde genellenemeyen problem veya süreçlerin temsilinde kullanılmaktadır. Bu nedenle uzmana olan ihtiyaç

sureklidir. Daha önce sözkonusu olmayan yeni bir durumla karşılaşıldığında bunun sisteme nasıl yansiyacağını, bağlı olarak yapılması gereken davranışları belirleyebilmek için uzman kişi, ihtiyaç duyulduğunda ulaşılabileceği bir konumda bulunmalıdır.

2.1.3. Uzman sistem geliştirme yöntemi

Bugüne kadar uzman sistem geliştirmek için tanımlanmış bir kaç süreç vardır. Aşağıdaki adımlar en çok kabul edilen süreci temsil etmektedir (Watts and Eldin, 1987).

- 1- Belirleme,
- 2- Kavramsallaştırma,
- 3- Biçimselleştirme,
- 4- Tasarım ve Gerçekleştirme,
- 5- Test Etme ve Değerlendirme,
- 6- Yaşatma ve Bakım Planlama.

Watts ve Eldin, belirleme adımını "problemi uzman sistem için uygun bir şekilde tanımlama ve problem karakteristiklerini belirleme" şeklinde açıklamıştır. Bir uzman sistem geliştirmek için gereken işlemleri yapmadan önce, hangi durumlarda bir uzman sisteme ihtiyaç olduğunu bilip, sözkonusu problem için gerekliyse, uzman sistem geliştirme süreci başlatılmalıdır. Yukarıda bahsedilen "problemi uzman sistem için uygun şekilde tanımlama ve problem karakteristiklerini belirleme" ifadesinin bu ön araştırmayı içerip içermediği açıkça anlaşılamamaktadır. Bu nedenle, belirleme adımına yönelik olarak şu işlemlerin yapılması yerinde olacaktır:

- i) Problem tanımlanmalı ve bir uzman sistem problemi olduğu açıkça ortaya konmalıdır. Problem, bir uzman sistem geliştirmeye ihtiyaç duyulacak boyutta değilse, bu işe hiç girişmemek daha iyi olacaktır.
- ii) Eger bir uzman sistem ihtiyacı görülüyorsa, daha sonra yapılacak iş, problem karakteristiklerinin belirlenip, tanımlanmasıdır. Sistem ve sistemin faaliyetleriyle ilgili olan ve sistemi etkileyen değişkenler bu tanımlamaların içine girmektedir.

Kavramsallaştırma, "bilgiyi organize etmek ve bilginin simgelenmesinde kullanılacak kavramları bulmak" şeklinde açıklanmaktadır. Bu adım şöyle ele alınabilir:

- i) Belirleme adımında yer alan değişkenlerin, birbirleriyle olan ilişkileri tanımlanır.
- ii) Bu değişkenler ve aralarındaki ilişkileri temsil etmek için kullanılacak olan simgeler belirlenir.
- iii) Tanımlanan bu değişkenlerin, değişkenler arası ilişkilerin ve kuralların nasıl ifade edileceği, bir başka deyişle, bilginin simgelenmesi için kullanılabilen farklı araçların neler olabileceği incelenir.

Biçimselleştirme adımında, bir önceki adımda incelenen kullanılabilir araçlardan bir tanesi seçilir. Ayrıca bu adımda, uzman(lar)dan sistemi gerçekleştirirken gerekli olacak kurallara yönelik bilgiler de alınır.

Tasarım ve gerçekleştirme adımı sırasında, bilgi, kurallar içinde ifade edilir ve seçilen araç kullanılarak kodlanır. Böylece değişkenler arası ilişkiler ve daha önce tanımlanan tüm kurallar, kullanılan aracın bilgi temsil

etme şekline uygun olarak ifade edilmiş olunur. Bu işlemler, uzman(lar) ile analiz ve tasarımcı görevini üstlenen kişi arasında yoğun bir bilgi alışverişini gerektirir. Bu arada, kullanıcı ile etkileşimi sağlayacak olan diyalog yapısını da içerecek şekilde, istenen uzman sistemin ilk modeli geliştirilmiş olunur.

Test etme ve değerlendirme adımında, sistem, uzmanların sistemi test edip onaylamasıyla geçerli hale getirilir. Uzmanlar, örnek problemleri kullanırlar ve sistemin problemleri çözdüğünden emin olurlar. Test sonrası, herhangi bir işlem adımını geliştirmek üzere geri dönmek mümkündür. Bu test etme ve geliştirme adımları sonrasında, sistemin istenilen düzeyde oluştuğuna karar verildiği anda, sistemi gerçekleştirme süreci sona erer. Bu açıklamalar doğrultusunda, sistemi gerçekleştirme sürecinin, aslında, bir önceki adım olan tasarım ve gerçekleştirme adımının sonucunda değil, burada açıklanan test etme ve değerlendirme adımının sonucunda tamamlandığını söylemek de yerinde olacaktır.

Son adım yaşatma ve bakım planlamadır. Bu adım Watts ve Eldin (1987) tarafından " bilgi tabanındaki bilginin güncel haliyle tutulması" şeklinde açıklanmaktadır. Bilginin en yeni haliyle sistemde bulunması, sistemin gerçekleştirilmesinden sonra meydana gelebilecek değişik durumların sisteme yansımalarına yol açacaktır. Söz konusu durumlar, sistemin gerçekleştirilmesi sırasında ele alınmamış ve daha önce karşılaşılmamış türden olursa, sistem bu farklı durumlar karşısında yapılabilecek herhangi bir işlem konusunda bilgi sahibi olmadığı için, bu değişikliklere cevap veremeyecektir. Bu olay bazı değişiklik ve/veya eklentilere olan ihtiyacın hissedilmesini sağlayacaktır. Gerçekten de

sistemin kuruluşunun tamamlanması, artık o sistemin dışarıdan gelecek hiç bir desteğe ve değişikliğe ihtiyacı olmadığı anlamına gelmemelidir. Aksine, yürütme sırasında, yukarıda açıklanan türden yeni durumlarla karşılaşırsa, yeni duruma yönelik olarak sistem üzerinde gerekli eklentiler ve değişiklikler yapılmalıdır. Bunun yapılabilmesi için de, yine o konunun uzmanı olan kişinin bilgi ve tecrübelerine başvurmak zorunludur. Böylece uzman kişiye olan ihtiyacın yalnızca sistemin gerçekleştirilmesi sırasında değil, yürütme sırasında da olduğu açıkça ortadadır.

2.1.4. Uzman sistem uygulamaları

Uzman sistemler, son yıllarda hızla gelişmiş ve pek çok alanda uzman sistem kullanımı denenmiştir. Bunun sonucu olarak, çeşitli konularda uzman sistem uygulamaları yapılmıştır.

Uygulamada ilk uzman sistem 1965' lerde Stanford Üniversitesi' nde geliştirilen ve organik maddelerin moleküler yapısının belirlenmesinde kullanıcıya (kimyagerler ve kimyasal maddeler üzerinde çalışmalar yapan araştırmacılar vb.) yardım eden DENDRAL' dir. Daha sonra 1970' lerde Carnegie Melon Üniversitesi' nde XCON uzman sistemi geliştirilmiştir. XCON, bilgisayar sistemlerindeki hataların saptanmasına yardım etmektedir (Ulengin, 1987). DOC, bilgisayarların incelenmesinde, CATS ise dizel lokomotiflerin incelenmesinde kullanılır. Bunlar, durum ve hareket bilgisi ile sezgiselliği kullanan kural temelli sistemlerdir (Türker ve Taşkın, 1989).

METAWASP, İngilizce cümle analizi yapan bir sistemdir. Daha önce bilmediği bir cümle yapısı ile karşılaştığında

onu inceler ve yeni bir yapı kalıbı yaratır. Böylece bildiği cümle yapılarının sayısını sürekli olarak arttırır (Timuroglu, 1988).

Bir başka uzman sistem, işgörenlerin sigortası için, bir sigorta şirketine düzenli olarak ödenen primlerin hesaplarını kontrol etmek amacıyla geliştirilmiştir. Bu kontrolün amacı, iş ve işçi faaliyetlerinin sigorta yönetim kurulunun yönergelerine uygunluğunu araştırmaktır. Bu konuda alınan kararlar, riskleri sınıflandırmaya ve her bir risk sınıfı için ayrılabilen ödeneklere yönelik olmalıdır. Bu, oldukça karmaşık, zaman alıcı ve hata olasılığı içeren bir iştir. Geliştirilen uzman sistem ile karşılaşılan zorlukların ve hataların önemli ölçüde azaldığı görülmüştür (Koster and Raafat, 1990).

ACE, telefon şebekelerindeki hataların bulunmasına yardımcı olan bir uzman sistemdir (Turban, 1990).

Özellikle tıp alanında çok sayıda uzman sistem geliştirilmiştir. PUFF, akciğer hastalıklarının teşhisine yardımcı olurken, VM yoğun bakımdaki hastaların durumunu izlemekte, BLUE BOX, ruhsal depresyon durumunda tedavi yöntemleri önermektedir (Ulengin, 1987).

İzleyen kesimlerde Endüstri Mühendisliği kapsamına girebilecek uzman sistem uygulamaları ayrıca ele alınacaktır.

2.2. Yöneylem Araştırması ve Uzman Sistemler

Yöneylem Araştırması ve Yapay Zeka (bağlı olarak da Uzman Sistemler) bilim dalları arasında paylaşılan temel amaç, karar verme ve problem çözmeyi destekleyici yöntemler geliştirmektir. Bu iki disiplinin bu amaca yönelik olarak yaklaşımları farklı olmakla beraber, birbirlerini tamamlayıcı özellik göstermektedirler. Yapay Zeka problem çözme teknikleri, çıkarımcı ve uzmanın bilgisine bağlı kalan bir yaklaşım izlerken; Yöneylem Araştırması algoritmik, matematiksel temellere dayanan yaklaşımları esas almaktadır. Yapay Zeka, problemlerin niteliksel yönlerini, Yöneylem Araştırması ise niceliksel yönlerini vurgular. Fakat problem çözmek için, bu iki yaklaşımın dikkatlice bütünleştirilmesi ile, problem çözme sistemlerinin yeteneğini ve özellikle de kabul edilebilirliğini geliştirme konusunda anlamlı sonuçlar elde edilebileceği anlaşılmıştır (Doukidis and Ray, 1990).

Yukarıdaki paragrafta Yöneylem Araştırması ve Uzman Sistemler tekniklerinin, problem çözme ve karar vermede birbirlerini tamamlayıcı özellik gösterdiği öne sürülmektedir. Gerçekten de bu iki yaklaşımın her biri diğerinin ele alamadığı ya da yeterince gözönünde bulundurmadığı özellikleri çözüme yansıtılabilmektedirler. Örneğin, Yöneylem Araştırması, çoğunlukla algoritmik ve niceliksel bir yaklaşım izlerken, Uzman Sistemler sezgisel ve niteliksel yaklaşıma daha fazla önem verirler. Bu açıdan bakıldığında, Yöneylem Araştırması tekniklerini kullanırken, sayısal olarak ifade edilemeyen, ölçülemeyen, bir başka deyişle modele katılamayan bazı etkenler Uzman Sistem

yaklaşımıyla dikkate alınabilmektedir. Böylece bu iki yaklaşım, birbirini tamamlayıcı bir özellik göstermektedirler.

Bölüm 2.1.3.' de açıklanan, Uzman Sistem geliştirmek için izlenebilecek yöntem ile Yöneylem Araştırması bilimsel yaklaşımındaki (Kara, 1985),

- Problemin belirlenmesi,
- Modelin geliştirilmesi,
- Modelin çözülebilirliğinin sağlanması,
- Modelin çözümü ve kanıtlanması,
- Çözümün uygulanması

adımları incelenecek olursa, iki yöntemin adımları arasındaki benzerlikler dikkati çekecektir. Buradaki problemin belirlenmesi, uzman sistem geliştirmek için tanımlanan yöntemdeki belirleme adımına; modelin geliştirilmesi, kavramsallaştırma, biçimselleştirme ve tasarım ve gerçekleştirme adımlarına; modelin çözülebilirliğinin sağlanması, test etme ve değerlendirme adımına karşı gelmektedir. Çözümün uygulanması da, kısmen yaşatma ve bakım planlama ile eşlenebilir. Yalnız yaşatma ve bakım planlama sırasında, ileride gerekli olması durumunda sistemde değişiklikler yapmak mümkündür. Yukarıdaki, çözümün uygulanması adımı ise, önceki adımlarda erişilen çözümün kullanılması anlamındadır.

Aslında uzman sistemlerle de problem çözmeye dönük bilimsel araç ve tekniklerin kullanımı amaçlandığından, bir anlamda bilimsel bir disiplin olarak uzman sistemler uygulamalı bilim niteliği göstermektedir. O halde karmaşık

Örgütlerin problemlerinin belirlenmesi ve çözümünde, Yöneylem Araştırması ve Uzman Sistem yaklaşımlarının birlikte kullanımı, bu benzerlikten dolayı kolaylaşmakta, birlikte kullanım halinde ise ilk iki paragrafta açıklanan nedenlerden dolayı problem çözümündeki etkinlik artmaktadır.

2.3. Endüstri Mühendisliği ve Uzman Sistemler

Endüstri Mühendisleri, yönetime kurmaylık fonksiyonlarından dolayı, uzman sistemlerle ilgili araştırma ve uygulamalarda önemli bir yere sahiptirler. Fabrika yerleşimi planlaması, süreç kontrolü, ekonomik analiz, otomasyon, proje yönetimi gibi endüstri mühendislerinin fonksiyonel sorumlulukları ile doğrudan kesişen alanlarda yapılmış uzman sistem uygulamaları da bu durumun bir göstergesidir. Uzman sistemler, uzman tarafından kullanılan problem çözme yordamlarını esas alarak, uzmanın düşünme sürecini taklit etmektedirler. Böyle uzman yordamlamaları uzun zamandır endüstri mühendislerine özgü bir özellik olarak gözönünde bulundurulmaktadırlar. Bu nedenlerle endüstri mühendisleri ve uzman sistemler bileşiminin hali hazırda zaten var olduğunu söylemek mümkündür (Badıru, 1988).

Bu açıklamasında Badıru, uzman sistemler konusunun bu derecede önem kazanmasından önce de, gerek problemlerin çözümünde, gerekse klasik sistemlerin yürütülmesinde, uzman sistemlerdekine benzer sezgisel yaklaşımların ve tecrübelerin zaman zaman dikkate alındığını, yalnız bunların uzman sistemlerdeki gibi belli bir yöntemi izleyerek ve sistematik biçimde değil de, daha dağınık ve tanımlanmamış işler olarak yapıldığını söylemektedir. Badıru, önceden de yapıldığını açıkladığı bu çalışmaların o zamanlar endüstri mühendisleri tarafından gerçekleştirildiğini öne sürerek, daha sonra bir

bilim dalı olarak tanımlanıp, şekillenen uzman sistem kavramını da bu nedenle Endüstri Mühendisliği ile doğrudan ilişkilendirmektedir.

2.3.1. Uzman sistem ile Endüstri Mühendisliği etkileşimi

Endüstri Mühendisliği ve uzman sistemler arasındaki etkileşimi araştırmaya yönelik olarak yapılan çalışmalar, bu iki bilim dalı arasında aşağıdaki alanlarda bazı etkileşimlerin varlığını göstermiştir (Turban, 1986):

- Bir danışman olarak uzman sistemler,
- Bir modelleme aracı olarak uzman sistemler,
- Bir öğretici olarak uzman sistemler,

Bir danışman olarak uzman sistemler: Uzman sistemlerin çoğunun temel amacı, danışmanlıktır. Uzmanın bilgisi elde edilir ve bilgisayarda kodlanır. Kullanıcı (genellikle uzman olmayan bir kişidir) uzman yerine bu sisteme danışmaktadır.

Burada iki farklı durum ele alınacaktır: Yöneticiye verilen öneri, endüstri mühendisine verilen öneri.

Yöneticiye verilen öneri: Burada bilgisayar, yöneticiye danışmanlık eden bir endüstri mühendisi gibidir. Danışmanlık hizmeti, stok yönetiminden kapasite planlamaya kadar çeşitli problem çözme alanlarını içermektedir. Bu konudaki süreç şöyledir: (Turban, 1986):

- Yöneticiyle problemi tartışmak,
- Problemi tanımlamak,
- Problemi modellemek,
- Modeli çözmek,
- Duyarlılık analizi yapmak,
- Çözüm önermek,
- Yururluğe koymak.

Eğer endüstri mühendislerinin bilgisi kodlanır ve yeniden ihtiyaç duyulduğu bir zaman da bu bilgiye erişmek mümkün olursa, bir endüstri mühendisi gibi danışmanlık edecek bir uzman sistem yaratılmış olunur. Bu konuda ilginç bir çalışma Oregon State Üniversitesi'nde yapılmıştır. Geliştirilen uzman sistem şunları yapmaktadır (Turban, 1986):

- 1- Problemin genel yapısı belirlenir, (yerleşim mi, stok mu, yatırım mı?...),
- 2- Karakteristikler analiz edilip, model kurulur. Ürnegin 1. adımda bir yerleşim problemi olduğu sonucunu vermişse, 2. adımda amaç fonksiyonu ve kısıtlar belirlenecektir,
- 3- Bu adım niceliksel aracın önerilmesidir (Ürnegin doğrusal programlama ya da dinamik programlama gibi),

4- Bu adımda bilgisayar önerilen aracı kullanır ve problemi çözer. En son "Eğer olursa...(what-if)" analizi yapılır.

Yukarıda açıklanan örnek uygulamada, karşılaşılan problemin ne tür bir problem olduğunun belirlenmesinden, problemi çözmek için kullanılacak aracın seçimine kadar bir endüstri mühendisinin desteğini almak mümkün olmaktadır.

Endüstri mühendisine verilen öneri: Burada uzman sistem, endüstri mühendisi için bir danışman veya bir asistan gibi hizmet verir. Endüstri mühendisi, bilgisayarı bilgi kaynağı ya da bir öğretmen olarak kullanabilir. Örneğin, Endüstri Mühendisliği'ne özgü bir konuya ilişkin uzmanlık bilgisi içeren bir sistem, endüstri mühendisi için bir öğretici olacaktır. Böylece bir kitaba, bir kişiye başvurmak yerine sisteme danışabilecektir. Ya da bir uzman sistem, kullanıcı ile istatistiksel paketler (Örn. SPSS) arasında bir arayüz olabilir. Bu uzman sistem kullanıcıya özel bir analiz için hangi istatistiksel paketin uygun olacağını önerebilir. Böyle bir uzman sistem, bir uzman istatistikçinin yerini tutar. Böylece istatistik konusunda uzman olmayan bir endüstri mühendisi de bilgisayarın asistanlığı ile istatistiksel analiz yapabilecektir.

Bir modelleme aracı olarak uzman sistemler: Uzman sistemler karmaşık yönetsel problemleri modelleme de endüstri mühendislerine yardımcı olacak bir araç olarak

kullanılabilirler. Benzetim alanında bu konuda öncü çalışmalar yapılmıştır (Turban, 1986). Böylece uzman olmayan endüstri mühendisleri de karmaşık benzetimler yapabilirler.

Bir öğretici olarak uzman sistemler: Uzman sistemlerin amacı, uzmanın bilgisini önce bilgisayara, sonra da buradan uzman olmayan kullanıcılara aktarmaktır. Aktarılan bilgi bir öğretmene ait ise, bu durumda geliştirilen sistem öğretici bir uzman sistem olacak ve bir öğretmen gibi hizmet verecektir. Öğretici uzman sistemlerin temel üstünlüğü, öğrenme eyleminin pasif bir süreç olmaktan çok aktif halde gerçekleşmesini sağlamalarıdır. Bu tür bir uzman sistem tasarlanırken, kullanıcının (öğrencinin) bireysel ihtiyaçları ve özellikleri (öğrenme hızı, kullanılan terminoloji gibi) de gözönünde bulundurulabilir.

2.3.2. Endüstri Mühendisliği alanlarında uzman sistem uygulamaları

Bu kısımda, Endüstri Mühendisliği'ne yönelik olarak yapılmış uzman sistem çalışmalarından örnekler verilmiştir.

Çizelgeleme: Gerçek yaşam problemlerinde en iyi çizelgeleri elde etmek mümkün değildir. Bu güçlük çeşitli nedenlerden kaynaklanmaktadır. Birincisi, en iyi çizelgeye erişebilmek için çok sayıda alternatif sürecin değerlendirilmesi gerekmektedir. İkincisi, kullanılan bilgi genellikle hatalı veya eksiktir. Üçüncüsü ise, bilgi sürekli olarak değişmektedir. Bu çalışmada, uzman sistem metodolojisini kullanarak "iyi" çizelgeler geliştirip, gerektiği zamanda bu çizelgeleri güncleyebilecek yeni bir yaklaşım önerilmiştir (Biegel and Wink, 1989).

ISIS-II, Carnegie Melon Üniversitesi'nde gaz turbunu parçalarının üretim sürecini çizelgelemek için geliştirilmiş bir uzman sistemdir. Bu sistem Akıllı Yönetim Sistemi (Intelligent Management System) diye adlandırılan, firma düzeyindeki simulasyon ve kontrol programının bir parçasıdır. Sistem; maliyet, tezgah veya süreç kullanım imkanı, vardiya, ustabaşının tercihleri gibi kısıtlara dayanan değerlendirme fonksiyonlarını kullanan sezgisel arama yöntemleri yardımıyla, çizelgeler türetmektedir. Kısıtların kullanılmasıyla arama uzayı küçültülmüştür (Dağlı, 1986).

Üretim sürecinin pek çok aşamasında sorumluluk alabilen endüstri mühendisleri için, üretim faaliyetleri içerisine giren işlerin sırasını ve zamanını belirlemek önem taşımaktadır. Bu açıdan bakıldığında, çizelgeleme konusuna yönelik olarak yapılan çalışmaların endüstri mühendisleri ile yakından ilgili olduğu söylenebilir. Uzman sistemler konusu da, diğer pek çok Endüstri Mühendisliği alanında olduğu gibi, çizelgeleme faaliyetlerine yönelik olarak da ele alınmıştır. Öyle ki, Endüstri Mühendisliği alanları içerisindeki uzman sistem çalışmalarında, çizelgelemeyle ilgili olanlarının ağırlıklı olduğunu söylemek mümkündür.

Hücre Tipi Üretim Sisteminin Kontrolü: Bu kontrol sistemi geliştirilmeden önce, bütün hücresel faaliyetleri gösteren bir süreç modeli geliştirilmiştir. Bu model, tüm kontrol faaliyetlerini de içermektedir. Üretim süreci sırasında, kontrol kuralları bu konuda yetersiz kalabilir. Model, bu durumda sistemin kontrol fonksiyonlarına başvuracak, bilgi tabanındaki hücresel kontrol faaliyetleri incelenecek, çıkartım mekanizması da buradan süreci yönetmek için gerekli

olan, mevcut duruma uygun kuralları seçecektir. Bilgi tabanı, hücrelerdeki süreçler sırasında olabilecek problemleri çözmek için çeşitli kuralları içermektedir. Bu kurallar "Eğer olursa... (if-then)" yapısındadır (Teng and Black, 1989).

Yukarıda açıklanan uzman sistem, üretim süreci sırasında karşılaşılabilecek problemleri ve bunlarla karşılaşılması durumunda yapılması gereken işlemleri içerecek şekilde tasarlanmış, üretim kontrolü amaçlı bir uzman sistemdir.

Stok Kontrolü: Bu konuda yapılmış bir uzman sistem, stok kontrolü teorisinin ayrıntıları ve karmaşıklığı konusunda sıkıntı çekmeksizin kararlar vermek isteyen herhangi bir stok kontrol yöneticisi tarafından kullanılabilir. Kullanıcı bir karara ihtiyaç duyduğu zaman, sisteme basitçe danışabilmektedir. Sistem, verilen problem için en uygun olan stok kontrol modelinin seçiminde yardımcı olup, problemlerin çözümü için gerekli olan parametreleri de hesaplamaktadır (Sinha, Ghiaseddin and Matta, 1989).

Yukarıdaki paragrafta açıklanan uzman sistem, stok kontrolüne yönelik olarak yapılmış çeşitli uzman sistem uygulamalarına bir örnektir.

Benzetim: Atölye ortamının, gerçeğe aktarılabilir bilgilerin elde edilmesi amacıyla bir model üzerinde incelenmesine yönelik olan benzetim uygulamalarında, genellikle probleme özgü yaklaşımlar görülmektedir. Belli bir atölyenin belli bir sorunu modellenerek çözüm önerileri geliştirilmektedir. Bir çok atölyenin farklı sorunlarının incelenebileceği genel amaçlı bir atölye benzetim modeli

henüz görülmemektedir. Bu konudaki örnek bir çalışma, atölye tipi üretimin çok yönlü benzetimini gerçekleştirerek etkileşimli genel bir paketin oluşturulmasına yöneliktir. Bunun için önce atölye tipi üretim sistemi benzetiminde gerekli fonksiyonların analizi yapılmaktadır. İşlerin gelişleri, sıralanmaları, tezgahlarda işlemlere alınmaları, işlem sonrası montaj ve müşteriye sevk gibi atölye tipi üretim sistemi fonksiyonları, olabildiğince çok amaçlı, küçük ve standart olabilen bir dizi benzetim fonksiyonlarına dönüştürülmektedir. Bu fonksiyonlar kümesinden uygun kombinasyonlar yardımıyla, istenen özel şartlara kolayca uyabilen atölye benzetim modelinin kurulması, çalışmanın amacıdır (Dinçmen ve Demirci, 1989).

Atölye tipi üretim sisteminde verilmesi gereken bir çok kararın sonuçları, benzetim yardımıyla önceden incelenebilmektedir. Yukarıda açıklanan çalışma, bu şekildeki bir benzetim çalışmasında uzman sistem yaklaşımını göstermektedir.

2.4. Endüstri Mühendisliği, Yöneylem Araştırması ve Uzman Sistem Uzantıları, Mevcut Durum Işığında Egilim

Önceki kesimlerde, Yöneylem Araştırması ve Uzman Sistem yaklaşımlarının problem çözmek için birlikte kullanılmaları halinde, daha etkin sonuçlar elde edilebileceği belirtilmiştir. Diğer taraftan Yöneylem Araştırması tekniklerinin, özellikle endüstri mühendisleri tarafından problem çözmek amacıyla yaygın olarak kullanıldığı bir gerçektir. Bu durumda Yöneylem Araştırması ve Uzman Sistemler arasındaki etkileşimin Endüstri Mühendisliği çalışmalarına da yansımaları doğaldır. Endüstri mühendisleri, örgütsel faaliyetler sırasında karşılaştıkları problemlerin çözümü için Yöneylem Araştırması tekniklerini kullanırken, artık bununla birlikte Uzman Sistem kavramını da düşünmeye

başlayacak ve bu uç bilim dalının kendine özgü yaklaşımlarını kullanarak daha etkin çözümlere ulaşabileceklerdir. Karşılaşılan bir problem, eğer Yöneylem Araştırması teknikleriyle ele alınıyorsa, konu çoğunlukla, matematiksel ve algoritmik bir yöntemle ele alınmış olmaktadır. Diğer taraftan Uzman Sistem yaklaşımı da, Yöneylem Araştırması teknikleri ile çözüme katılamayan sezgisel yaklaşımların, matematiksel olarak ifade edilip modele katılamayan diğer etkenlerin de dikkate alınmasını sağlayacaktır. Endüstri mühendisi, kendine özgü problem belirleme ve çözmeye yönelik yöntemi takip ederek bu etkileşimlerden yararlanacak ve problemlerini çözebilecektir.

Önceki kesimlerde Endüstri Mühendisliği ve Uzman sistemler arasındaki etkileşimler ele alınmış ve bu arada endüstri mühendislerinin uzman sistemlerden nasıl yararlanabilecekleri açıklanmıştır. Endüstri mühendisleri, geliştirilmiş bir uzman sistemi kullanabilir olmakla birlikte, aynı zamanda uzman sistem geliştirmede de rol oynayabilirler. Bu rol tartışıldığı zaman, onların uzman sistem geliştirme yönteminin yaratıcıları değil kullanıcıları oldukları ortaya çıkar (Watts and Eldin, 1987). Burada bazı kavramlara açıklık getirmekte yarar vardır. Yöntem ile, uzman sistemi geliştirmek için izlenecek yol ve kullanılacak araç anlatılmak istenmektedir. Endüstri mühendisleri, mevcut yöntemi kullanıp uzman sistem geliştirebilirler. Nitekim olay bir sistem geliştirme çalışması olduğuna göre, endüstri mühendislerinin böyle bir konuya yakın olmaları da kaçınılmazdır. Fakat yöntemi, yani bu iş için kullanılacak bir dili veya paket programı geliştirmek bilgisayar mühendislerinin işidir.

Bu kısımda yapılan açıklamalara dayanarak, izleyen yıllarda Endüstri Mühendisliği, Yöneylem Araştırması ve Uzman Sistemler arasındaki etkileşimin giderek daha da önem kazanması beklenmektedir. Öyle ki, gelecekte endüstri mühendisleri için Uzman sistemler, belki de lisans seviyesinde Yöneylem Araştırması ile birlikte ele alınacak bir konu olacaktır.

3. BİR ÜRETİM İŞLETMESİNDE ÜRETİM PROGRAMINDAN SAPMALARIN UZMAN SİSTEM YAKLAŞIMIYLA İZLENMESİ

İşletmede günlük olarak türetilen raporları inceleyerek üretimin planlara uygunluk derecesini ölçmek mümkündür. Mevcut sistemde, ilgili raporlar yalnızca sayısal değerleri içermektedir. Bu değerleri yorumlamak, bir problem varsa nedenlerine ulaşmak için daha başka hangi raporlarda hangi bilgilere bakmanın gerekeceğini bilmek, sistemi iyi bilen birisi ile mümkün olur.

Çalışmanın bu bölümünde, bir üretim işletmesinde, üretim programındaki sapmaları izleyebilmek için, sadece sayısal değerleri gösteren raporların ötesinde, mevcut durumu yorumlayıp, sapmaların nedenlerini de açıklayan bir uzman sistem geliştirmek amaçlanmıştır.

3.1. İşletmenin Üretim Süreci

Ele alınan firma, seramik türü ev eşyası (klozet, lavabo, evye vs.) üreten bir işletmedir. Üretim süreci içerisindeki temel işlemler aşağıdaki gibidir:

- 1- Üretilecek mamulün alçı kalıplara dökümü yapılır,
- 2- Kalıplardan çıkarılan yaş dökümler kurutma dolaplarında kurutulur. Bu haliyle yarı mamul kuru bir çamur görünümündedir. Fakat biçimsel olarak istenen şekle girmiştir,
- 3- Sırlama kabinlerinde yarı mamulün üzerine istenen renkte sır puskürtülerek, renk verilir,

- 4- Sırlanan yarı mamul fırında pişirilerek ana işlemler tamamlanır,
- 5- Kalite kontrol işlemi ile mamuller sağlam, iskarta ya da taşlama sınıflarına ayrılır. Taşlama, gerekli görülmesi durumunda mamuller üzerinde rötüş yapma işlemidir.

3.2. Üretimi İzlemede Uygulanmakta Olan Yaklaşım

İşletmenin Üretim Hizmetleri Bölümü, yıllık üretim planını yaparak, bu plandan hareketle günlük programlanan değerleri üretmektedir. Bu değerler aynı zamanda günlük programlanan adette üretimi yapabilmek için, sürecin ilk aşamasından başlayarak, her safhada üretilmesi gereken yarı mamul adetlerini de kapsamaktadır. Üretimi izlemek için bu planlanmış (aylık) ve programlanmış (günlük) değerlere paralel olarak her gün karşılık gelen gerçek değerler tutulup günlük raporlar üretilmekte, plan-gerçek karşılaştırmaları yapılmaktadır. Bu bilgiler bilgisayara girilip raporlar da bilgisayardan alınmaktadır.

Plan-gerçek karşılaştırmasının sonucunda görülebilecek sapmaların nedenlerini bulmak için süreci iyi bilip problemlerin kaynağını tahmin edebilecek kişilere ihtiyaç vardır. Mevcut sistemde Üretim Hizmetleri Bölümü'nde çalışmakta olan iki mühendis bu konuda yeterince bilgi sahibidirler. Günlük sapmalarda, nedenleri hangi aşamalarda aramaları gerektiğini bilip, ilgili raporları inceleyerek gerekli uyarılarda bulunabilmektedirler. Koşullarda herhangi bir değişiklik olduğunda gerekli değişiklikleri yapıp, yeni karar politikalarını belirlemek bu kişilerin varlığıyla mümkün olmaktadır. Bir başka deyişle, sistemde uzman kişiye olan ihtiyaç süreklidir. Çünkü uzman sistemin yaşayabilirliğini yine uzman sağlayacaktır.

3.3. Uretim Programından Sapma Nedenlerini Bulmak İçin Uzman Sistem Yaklaşımının Uygulanabilirliği

Bir uzman sistem, kullanıldığında, hazırlanmasında esas alınan konuyla ilgili olarak bir uzmanın vereceği yanıtı veya çözümü verebilmelidir. Bunun için, daha önceden, hangi koşullarda, hangi sonuçları vereceği konusunda programlanmalıdır. Ele alınan sistemde, bir sapmanın nedenini bulabilmek için, o nedenin nerelerde aranması gerektiği bilinmelidir. Bu bilgi ancak o süreci iyi tanıyan, problem çıkabilecek aşamaları tahmin edebilecek bir kişiden temin edilebilir. Bu nedenle, program(lar)ı yazacak kişinin, ele alınan konuda derinlemesine bilgi sahibi olan bir uzmana ihtiyacı vardır.

Mevcut sistemde uzman kişi zihnine yerleşmiş ve alışkın olduğu belli bir sistematik içerisinde, bazı sorgulamaları yapıp, sisteme ilişkin tüm bilgisini ve tecrübelerini de katarak günlük birtakım değerlendirmeler yapmaktadır. Sayısal bilgilerin yine bilgisayarda tutulup, günlük raporların da bilgisayardan alınıyor olmasına rağmen, geliştirilen sistem olmadığında uzman, her gün üretimi kafasında kurduğu bu sistematik doğrultusunda bir dizi faaliyeti inceleyerek değerlendirecek ve bu işe de yalnızca sayısal bilgileri içeren çok sayıda raporu alarak başlayacaktır. Fakat aynı mantığı bir kez programa aktarmakla aynı değerlendirmeyi her gün yapmak mümkün olacaktır. Üstelik, bilgisayar kullanımından dolayı, değerlendirme sonuçları daha hızlı alınacaktır. Burada bir kez programı hazırlamak için zaman harcanacak, bu arada uzman, boşalan zamanını daha başka işlere ayırabilecektir.

Uzman sistemlerin kullanım amaçlarından birisi de "gerçekleşen bir olayı olması beklenen durum ile karşılaştırarak izlemek, herhangi bir sapma meydana geldiğinde bunu bildirmek" tir (Ulengin, 1987). İlgilenilen süreçte de "üretim planı", uyulması beklenen durumdur. Günlük fiili

Uretim, gerekleŒen olaylardır. Bu iki durumu karŒılaŒtırarak sapmalarla ilgilenilmektedir. alıŒma bu aıdan uzman sistemlerin belirtilen amaca ynelik kullanımına bir rnek olacaktır.

3.4. Sistemin GeliŒtirilmesi

Bu kısımda, gerekleŒtirilen uzman sistemi geliŒtirme alıŒması aıklanmıŒtır. Bu alıŒma iin, daha nce uzman sistem geliŒtirmeye ynelik olarak aıklanan yntem kullanılmıŒtır.

3.4.1. Belirleme

Problem, gnlk gerekleŒen Uretim gnlk programlara uymaması durumunda bu sapmanın nedenlerini bulmaya yneliktir. Gerek üretim srecinin karmaŒık olması, klasik bilgisayar yazılımlarının kullanımına olanak verecek Œekilde genellenemeyiŒi, klasik bilgisayar yazılımlarında ele alınması g olan bazı zel durumları iermesi, dolayısıyla o konuda uzman bir kiŒinin bilgisine ihtiya duyulup bu kiŒiye istenildiğinde eriŒme imkanının olması gibi nedenler, problemin zumnde uzman sistemin uygunluęu iin ne srlebilecek gerekelerdendir.

Problem bu Œekilde belirlenip, zumnde uzman sisteme ihtiya duyulduęu ortaya ıkarıldıktan sonra ele alınan problemle ilgili aŒaęıdaki karakteristikler belirlenmiŒtir:

- * Bir mamulden aylık uretilmesi planlanan adet,
- * Planlanan adette aylık uretimi yapabilmek iin, aylık olarak yapılması gereken brut dkm adedi,
- * Aylık, brut dkmden elde edilmesi gereken toplam net dkm adedi,
- * Aylık planlanan adette uretime eriŒebilmek iin o mamulden gnlk uretilmesi gereken adet
- * Gnlk programlanan adette mamul uretebilmek iin o gnku üretim srecine baŒlarken dklmesi gereken yarı mamul adedi ve bu dkmden elde edilmesi gereken net

döküm adedi,

- *Günlük sızlanıp fırına girmesi gereken mamul adedi
- *Günlük programlanan fırın fire ve döküm fire değerleri

Yukarıda belirlenen karakteristikler için, gün bazında olduğu gibi her güne gelindiğinde birikimli olarak olması planlanan değerler de ele alınmıştır. Buna göre herhangi bir gün için, o güne gelindiğinde, birikimli olarak:

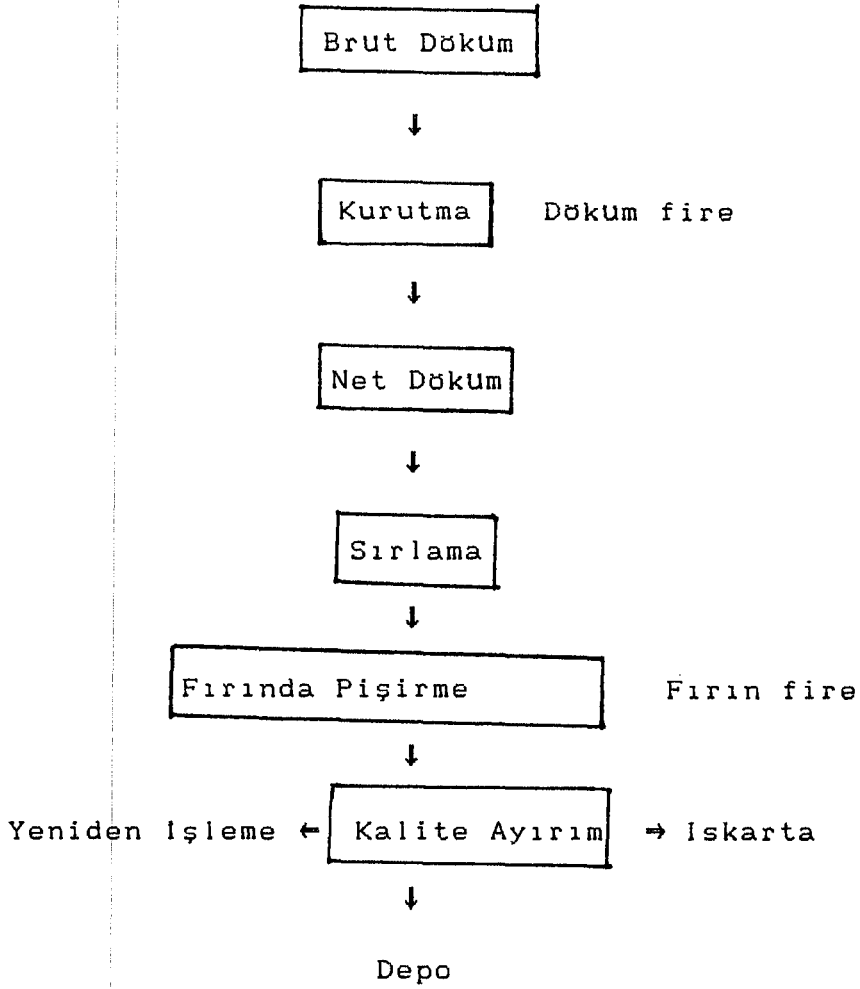
- *Depoya girmiş olması gereken toplam mamul adedi,
- *Yapılması gereken toplam brut döküm adedi,
- *Elde edilmesi planlanan toplam net döküm adedi de ele alınacaktır.

Ayrıca, aylık olarak planlanan ve günlük programlara dönüştürülen tüm bu değerler için gerçekleşen değerler de tutulacaktır.

3.4.2. Kavramsallaştırma

Bu aşama, belirleme adımında yer alan değişkenlerin birbirleriyle olan ilişkilerinin tanımlanmasıyla başlamaktadır.

Şekil 3.1.'den de görüleceği gibi, üretim süreci, belli sayıda döküm yapılarak başlar. Bu dökümlerin bir kısmı fire olur ve elde net dökümler kalır. Bunların hepsi sızlanıp fırına girer. Bir kısım yarı mamul de bu aşamada fire olur (fırın fire). Geriye depoya girecek mamuller kalır. Bu açıklamalardan anlaşılacağı, ayrıca Şekil.3.1.'deki süreçten de görüleceği gibi, döküm fire ve fırın fire olmak üzere iki farklı fire kavramı vardır. Süreç aşamalarına ilişkin olarak planlanan ve programlanan değerler, depoya girmesi planlanan adetten sürecin ilk aşamalarına doğru giderek ve fire paylarını ekleyerek bulunur.



Şekil.3.1. İşletmedeki Üretim Süreci

Ürneğin fırından çıkan mamullerin fırın fire olanları ayrıldıktan sonra, geriye kalanı depoya girer. Buradan, depoya belirlenen adette mamulün girebilmesi için, bundan bir önceki adım olan fırında pişirmede, kaç adet mamulün fırına girmesi gerektiği hesaplanabilir ve bu değer, planlanan depo girişine fırın firesi eklenerek bulunabilir. Aynı mantıkla aşağıdaki çıkarsamaları elde etmek mümkündür. Sırlanan mamullerin de hepsi fırına girmektedir. Sırlama adedi de aynı zamanda eldeki net döküm adedine eşittir. Çünkü yapılan brut dökümden döküm fire olan yarı mamuller ayrıldıktan sonra geriye kalan sağlam dökümlerin (net döküm) hepsi sırlanmaktadır. Buradan

$$\text{Sırlama adedi} = \text{Net döküm adedi}$$

olduđuna ve sirlama adedi de bilindiđine (= fırına giren mamul adedi) göre, brut döküm adedi, net döküm adedine döküm fire deđeri etki ettirilerek bulunabilir.

Böylece aylık planlanan deđerler, her safha için ayrı ayrı bulunabilir. Daha sonra bu planları günlük programlara dönüştürmek için, ne gibi iliřkilerin kullanılabilceđi incelenmelidir.

Aylık planlanan adette üretimi yapabilmek için günlük kaç adet üretim yapılmalıdır sorusuna yanıt bulabilmek için, aylık plan deđeri o ay için kalite ayırım bölümünün eldeki toplam çalışma günü sayısına bölünür. Çünkü üretimi tamamlanan mamuller kalite ayırımdan geçmeden depoya giremezler. Bađlı olarak depoya girecek mamul adedi, kalite ayırım bölümünün bir günde deđerlendirebileceđi mamul adedine bađlıdır. Benzer řekilde her safha için bulunan aylık plan deđerleri de ilgili safhada günlük yapılacak yarı mamul üretimini programlamada kullanılır. Öyleyse,

$$\begin{array}{r} \text{aylık planlanan brut} \\ \text{döküm adedi} \\ \text{günlük programlanan} \\ \text{döküm adedi:} = \frac{\text{-----}}{\text{aylık döküm günü sayısı}} \\ \text{(brut döküm)} \end{array}$$

$$\begin{array}{r} \text{aylık planlanan net} \\ \text{döküm adedi} \\ \text{günlük programlanan} \\ \text{net döküm adedi:} = \frac{\text{-----}}{\text{aylık döküm günü sayısı}} \end{array}$$

$$\begin{array}{r} \text{aylık planlanan sirlama} \\ \text{adedi} \\ \text{günlük sirlanması} \\ \text{programlanan adet} = \frac{\text{-----}}{\text{aylık kalite ayırım} \\ \text{günü sayısı}} \end{array}$$

Yukarıda açıklanan değişkenler için daha sonra birikimli planlanan değerler de hesaplanır. Her safhanın haftalık çalışma günü sayısı farklıdır. Örneğin: Fırınlara haftada 7 gün çalıştığından diğer safhalar 5 veya 6 günde, fırına, 7 günde pişireceği kadar yarı mamul üretirler.

Herhangi bir gün herhangi bir safha için birikimli plan değeri; o safha için günlük olarak planlanan değerle, bulunulan günün o safha için kaçınıcı iş günü olduğunu veren değer çarpılarak bulunur. Örneğin, günlük dökülmesi programlanan adet 500 iken 3. döküm gününe gelindiğinde, o günkü dökümle beraber birikimli olarak 1500 ($=3*500$) adet döküm yapılmış olmalıdır.

Günlük programlanan değerlere karşılık olarak girilecek gerçek değerler de, bir önceki günün gerçek değerlerine eklenerek her safha için ayrı ayrı birikimli değerler hesaplanır.

Buraya kadar değişkenler ve değişkenler arasında olan ilişkiler açıklanmıştır. Bundan sonra, tanımlanan bu değişkenleri temsil etmek için kullanılacak olan simgeler belirlenmiştir. Örneğin:

DAdet: Üretilip depoya girmesi gereken mamul adedi

BrDok: DAdet mamul elde etmek için elde bulunan süre içerisinde toplam olarak dökülmesi gereken brut döküm adedi

NetDok: Elde bulunan süre içerisinde yapılacak toplam brut dökümden elde edilmesi gereken net döküm adedi

Yukarıda örnek olarak uçu verilen değişkenlerin tamamı, izleyen kısımlarda, programda kullanılan veri tabanları ile birlikte açıklanmıştır.

Problem belirlenip, kavramları ve kullanılacak değişkenler belirlendikten sonra kullanılacak aracın seçilmesi gerekmektedir. Bir uzman sistem, çeşitli araçları kullanarak geliştirilebilir. Basic, Pascal gibi bilinen programlama dilleri bu amaçla kullanılabilir. Fakat bunların yanısıra özel olarak uzman sistem geliştirmek için hazırlanmış başka diller de vardır. Bunlara örnek olarak Prolog ve Lisp verilebilir. Ayrıca programlama dillerinin yanısıra, uzman sistem kabuğu (shell) olarak isimlendirilen bazı özel yazılımlar da geliştirilmiştir.

3.4.3. Biçimselleştirme

Bu adım, bir önceki adımda bahsedilen araçlardan birisini seçerek, uzmanlardan gerekli bilgileri alma çalışmalarını içermektedir.

Bilinen bilgisayar programlarına göre, özel olarak uzman sistemler konusu için geliştirildiklerinden, bu durumun getireceği kolaylıklardan yararlanmak için ya bir uzman sistem geliştirme amaçlı programlama dili ya da bir uzman sistem geliştirme kabuğunun kullanılması düşünülmüştür. Daha sonra, uzman sistem geliştirmeye yönelik bir bilgisayar programlama dilini kullanmanın kabuklara göre daha esnek bir çalışma olanağı vereceği, böylece kabukların biçimsel formatlarına uyma zorunluluğundan kurtulunacağı gerekçeleriyle, en yaygın kullanılan uzman sistem geliştirme dillerinden birisi olan PROLOG' da karar kılınmıştır.

Bu çalışmada Uretim izlenmesi amacıyla Prolog dilini kullanarak altı ayrı program yazılmıştır. Hazırlanan sistem, bu programların birbirleriyle bütünleşik olarak çalışmasını sağlayarak, bu arada kullanıcı ile etkileşime de imkan vermektedir.

Uzman Sistem çalışmaları için en çok kullanılan dillerden birisi olan Prolog, 1970' lerin başlarında

Marseilles Universitesi' nde Professor Alain Colmerauer ve meslektaşları tarafından geliştirilmiştir (Scown,1985). Bir Prolog programı dört temel program kesiminden oluşmaktadır:

DOMAINS: Programda kullanılacak değişkenlere ilişkin (Tanımlamalar) tanımlamaların yapıldığı kesimdir.

PREDICATES: Programda kullanılacak argümanların ve bu (Yüklemeler) argümanların tiplerinin tanımlandığı kesimdir.

CLAUSES: Bir Prolog programının en önemli kesimidir. (Cümleler) Olay (fact)'lar ve kural (rule)'lardan oluşur. Olaylar ve kurallar, programın verilen amacı gerçekleştirmek için yapması gereken işlemlerin tanımlandığı, birbirinden ayrı yazılmış bir dizi algoritmadır. Koşullar ve koşullara göre yapılması gereken işlemler burada tanımlanmaktadır.

GOAL: Kullanıcı veya tasarımcı tarafından belirlenip, program tarafından gerçekleştirilmesi istenen amaç, bu kesimde belirtilir. (Amaç)

Bu aşamada daha sonra uzman kişi ile yoğun bir bilgi alışverişinde bulunularak, mevcut sistemde üretime yönelik olarak o kişinin üstlendiği görevleri ve fonksiyonları tartışıp, hangi durumlarda ne gibi yaklaşımlarda bulunduğunu öğrenmek amaçlanmıştır. Buna göre, günlük üretimi değerlendirmeye nereden başladığı, (hangi raporların üretildiği, hangi raporlarda hangi bilgilere öncelikle baktığı), bir gün içerisinde hangi farklı durumlarla karşılaşılacağı, bunların nerelerden (sürecin hangi aşamalarından) kaynaklanmış olabileceğini nasıl tahmin edebildiği gibi konulara yönelik bilgiler alınmıştır.

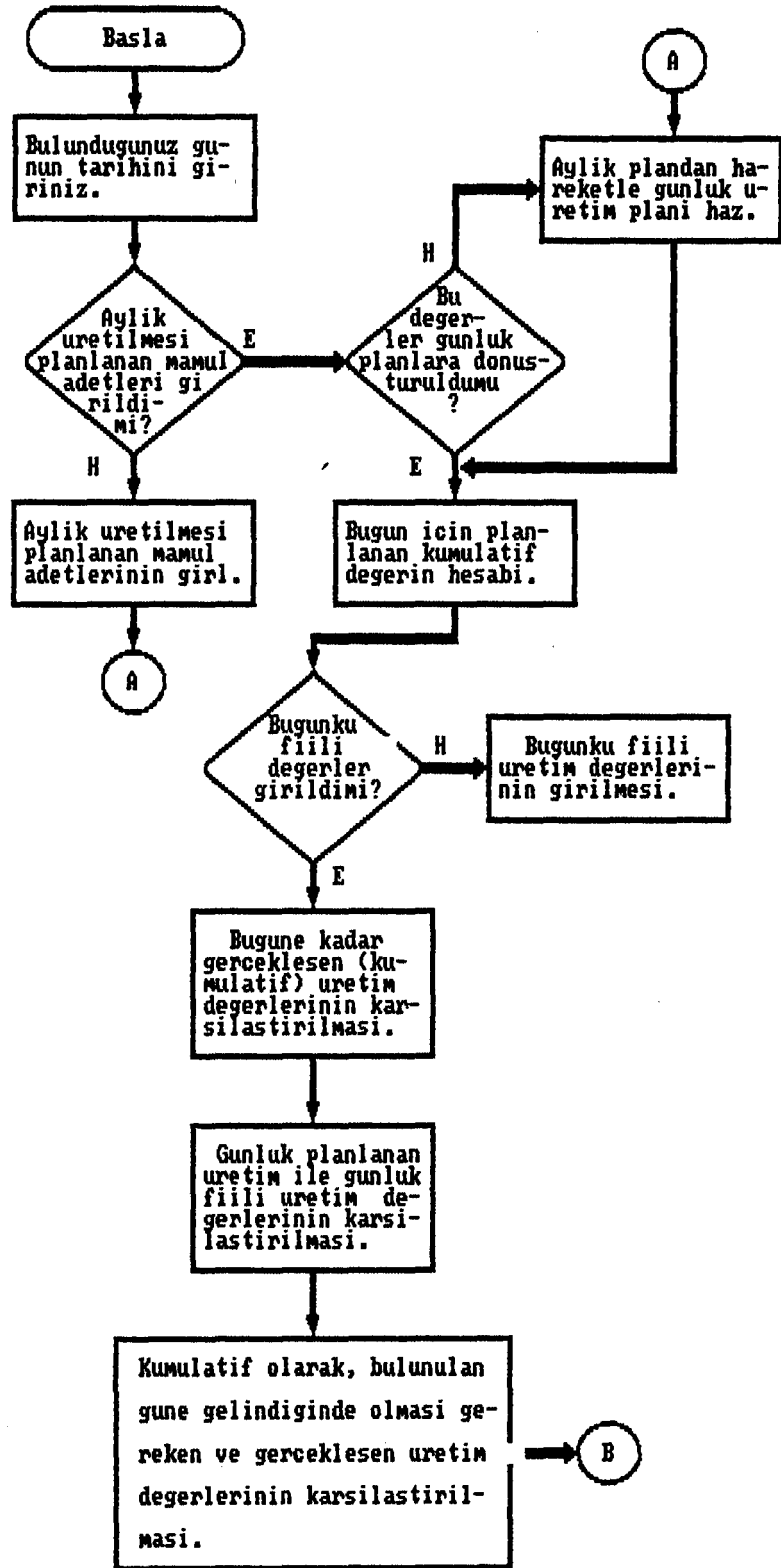
İlgili uzman kişiden alınan bilgilere göre, günlük Üretimi izlemek için geliştirilen uzman sistemde aşağıdaki işlem adımlarının olmasına karar verilmiştir. Bu adımlar, gerçek Üretim sürecinde Üretimi izlemeye yönelik faaliyetleri temsil etmektedir.

- 1- **Aylık planlanan Üretim değerlerinin son ürün temelinde girilmesi:** İşletmede her ürün için o ay Üretilmesi planlanan adetler belirlenmektedir. Daha sonra bu adetleri kullanarak, planlanan adette Üretimi yapabilmek için sürecin her safhasında işlem görmesi gereken yarı mamul adetleri hesaplanmaktadır.
- 2- **Aylık planlanan değerlerin günlük programlara dönüştürülmesi:** Bir ürünle ilgili olarak aylık planlanan değerler her safha için elde bulunan çalışma günü sayısına bölünerek günlük Üretilmesi programlanan değerlere dönüştürülmektedir.
- 3- **Günlük programlanan değerlere karşılık gelen günlük gerçek değerlerin girilmesi:** İlgilenilen güne ait Üretimde bir sapma olup olmadığını öğrenebilmek için programlanan değerlerle karşılaştırılacak gerçek değerlere ihtiyaç vardır. Bu nedenle günlük Üretime ilişkin gerçek değerler sisteme veri olarak girilmelidir.
- 4- **Günlük programlanan ve gerçek değerlerin karşılaştırılması, ilgilenilen kod için Üretim programlara uygun olarak gerçekleşmemişse (Üretim, programın gerisinde kalmışsa), bu aksaklığın süreçteki hangi aşama(lar)da ortaya çıktığının bulunması:**
 Üretimin beklenen adetlerde gerçekleşmemesi bir veya birkaç aşamadan kaynaklanabilir. Genellikle bir aşamada (safhada) ortaya çıkan sapma izleyen safhalardaki işlemlere de yansiyacaktır. Sisteme herhangi bir Üretim gününde karşılaşılabilecek farklı durumlar bir bilgi tabanı oluşturularak

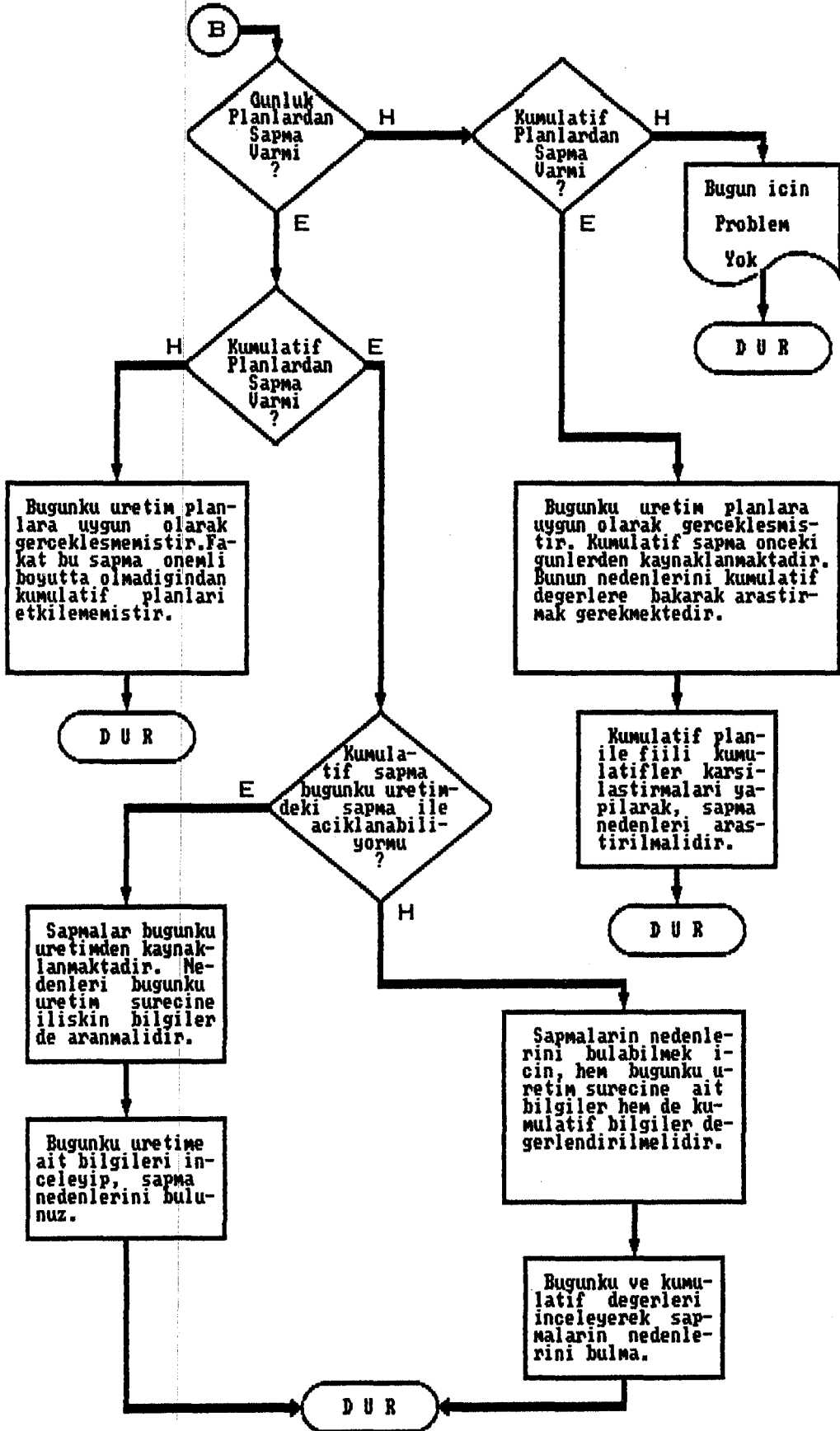
yazılmıştır. Herhangi bir üretim gününde bu durumların bir veya bir kaç tanesiyle karşılaşılabilir. Bir sapma varsa bunun karşılaşılabılır durumlardan hangilerine uyduğu, bilgi tabanındaki farklı durumlar taranarak bulunur ve karşı gelen uyarılar yapılır.

3.4.4. Tasarım ve gerçekleştirme (geliştirilen sistemin yapısı)

Bu aşamaya gelindiğinde artık kullanılacak araç seçilmiş ve bu araçta kodlanacak bilgiler de elde edilmiştir. Seçilen aracı (bu çalışma için Prolog) kullanarak, uzmanlardan elde edilen ve gerçek sistemi niteleyen bilgilerin değişkenler ve kurallar ile ifade edilmesi sağlanır. Bu kurallar, gerekli tanımlamalar vb. ile birlikte sistemin bilgisayar programlarını oluştururlar. Şekil.3.2., bu sistemin işleyişindeki mantıksal akışı göstermektedir.



Şekil.3.2. Geliştirilen sistemin mantıksal akış şeması



Şekil.3.2. (devam)

Aslında Prolog dilinde diğer programlama dillerinde alışkın olduğumuz terminolojinin dışında kalan farklı bazı terimler ve ifade biçimleri vardır. Bu farklı ifade biçimleri diğer dillerde mümkün olmayan ya da daha zor olan bazı gösterimleri mümkün kılar.Örneğin:

```
YeniGun(EskGun,Gun):- EskGun="pazartesi",Gun="salı";
                       EskGun="salı",Gun="çarşamba";
                       EskGun="çarşamba",Gun="perşembe";
                       EskGun="perşembe",Gun="cuma";
                       EskGun="cuma",Gun="cumartesi";
                       EskGun="cumartesi",Gun="pazar";
                       EskGun="pazar",Gun="pazartesi".
```

Yukarıdaki gösterim Prolog'da bir kurala karşı gelmektedir. Kuralların bir baş kısmı bir de gövdesi vardır. Bu örneğe göre YeniGun(EskGun,Gun) ifadesi kuralın baş kısmı, yanında verilen diğer ifadeler ise gövdesidir. Gövdeyi bir koşul ya da bir koşullar seti oluşturur. Kuralın başarılı olabilmesi, bu koşulun sağlanması durumunda mümkündür. Bazen de kuralın başarılması için birden fazla koşuldan birisinin gerçekleşmesi yeterlidir. Bu durumda "mantıksal veya" ya karşı gelen ";" ile kurallar birbirinden ayrılır. Burada ele alınan kural, EskGun değişkeninin değerine bakarak, eğer bu değer kuralın gövdesinde tanımlanan koşulların birisindeki değere uyuyorsa, Gun değişkenine o koşulun gerçekleşmesi halinde öngörülen değeri atamaktadır. Örneğin EskGun="salı" iken Gun değeri bulunmak isteniyorsa kural, YeniGun("salı",Gun) ifadesi için ele alınacaktır. Burada bilinen değer "salı", bilinmeyen değer Gun değişkeninin değeridir. Tanımlanan durumlar içerisinde EskGun="salı" iken karşı gelen Gun değişkeninin değeri "çarşamba" dır. Böylece bu kural başarılı olacak ve Gun değişkenine "çarşamba" değeri atanacaktır.

Aşağıda, geliştirilen sisteme ait kurallara başka bir örnek verilmiştir:

```

BrutD(BrDokS, Gunfbdok, Gunpbdok):-BrDokS>,
write("Planlanan brut döküm adedi"),
write(Gunpbdok),write("iken"),
write(Gunfbdok),write("adet döküm yapılmıştır"),nl,
wite("Sapma oranı="),write(BrDokS);
true.

```

Bu kurala göre, günlük programlanan ve gerçekleşen brut döküm adetleri ve sapma değerleri verildiğinde, eğer sapma sıfırdan büyük ise, bu durum kullanıcıya bildirilecek, değilse bir mesaj yazılmayacaktır. Buradaki "," işareti "mantıksal VE" anlamındadır. Kuralın başarılması için birbirinden ";" ile ayrılmış iki koşul vardır. Birinci koşul kendi içinde "," lerle ayrılmış birden fazla ifadeden(koşuldan) oluşmaktadır. Bu nedenle ilk koşul ancak bu ifadelerin hepsinin gerçekleşmesi durumunda sağlanır. Buna göre, kurala BrDokS, Gunfbdok ve Gunpbdok değerleri gönderildiğinde BrDokS>0 ise izleyen ifadelere geçilecek ve sırasıyla onlar yapılacaktır. Bu ifadeler, ekrana yazılacak bazı mesajları içermektedirler. Eger bu koşul sağlanamazsa (BrDokS>0 değilse) henüz kural başarılamadığından ";" Un diğer tarafına geçilecek ve diğer koşula bakılacaktır. Buradaki "true" ifadesi her durumda kuralın başarılı sonuçlanmasını sağlar.

Geliştirilen sisteme ilişkin bilgisayar programı ekte verilmiştir. Sisteme, kullanıcı ile etkileşimi sağlayacak bir diyalog yapısı da eklenmiştir. Sistem ilk çalıştırıldığında, kullanıcı, kolay anlaşılır mesajlar içeren menülerle karşılaşmakta ve işlemlerini bu mesajlara göre yönlendirmektedir.

3.4.4.1. Programda kullanılan değişken ve veri tabanları

Bir veri tabanının bir kaydını, bir dizi değişken oluşturur. Bu nedenle, programda kullanılan veri tabanları

açıklanırken, bunların kayıt desenlerinde yer alan değişkenler de birlikte açıklanmıştır. Örnek olarak aylık1 veri tabanı şu şekilde açıklanabilir.

aylık1(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gu2, Ay2).

"aylık1" veri tabanının bir kayıdı, yukarıda parantez içine yazılmış değişkenlerden oluşmaktadır. Bu değişkenler şöyle açıklanabilir:

Belirtilen kod numarası ve renge sahip ürün için,

DAdet: "Ay2" inci ayın "Gu2" inci gününe kadar üretilip depoya girmesi gereken adet,

BrDok: Belirtilen tarihe kadar "DAdet" adet mamul elde etmek için elde bulunan süre içerisinde toplam olarak dökülmesi gereken brüt döküm adedi,

NetDok: Elde bulunan süre içerisinde yapılacak toplam brüt dökümden elde edilmesi gereken net döküm adedi,

NetSır: Elde bulunan süre içerisinde sırlanması gereken toplam yarı mamul adedi,

BrFırGiriş: Elde bulunan süre içerisinde fırına girmesi gereken toplam mamul adedi.

Benzer şekilde diğer veri tabanlarının kayıt desenleri ve içerdikleri değişkenlerin anlamları Ek.1.' de açıklanmıştır.

3.4.4.2. Programın modülleri, çalışma şekli

Buyuk hacimli programların derlenmesi ve çalıştırılması sırasında bazı problemler çıkabilmektedir. Bu tür problemleri azaltmak ve programın çalışma hızını arttırmak için, büyük ve tek bir program yazmak yerine aynı

programı daha küçük birden fazla programa bölerek tek bir programmış gibi çalışmasını sağlamak mümkündür. Buna **Modüler Programlama** denir. Bu yöntemde birden fazla modülde kullanılması gereken değişkenlerin (eğer varsa) tek bir yerde bir kez tanımlanıp daha sonra diğer modüller tarafından ortak kullanımı mümkündür.

Modüler programlama ile çalışma hızı artmakta, program hacminin büyük olmasından kaynaklanabilecek problemler azalmaktadır. Bu çalışmada da aynı yol izlenmiştir. Aslında her modül diğer modüllerden bağımsız olarak yazılmış ayrı bir prolog programıdır. Ayrı bir yerde hangi modüllerin birlikte tek bir programmış gibi çalışacakları tanımlanmıştır.

Sistem,

TEZ1.PRO	TEZ3.PRO	TEZAMAC.PRO	GLOBTEZ.PRO
TEZ2.PRO	TEZ4.PRO	TEZ5.PRO	

modüllerinden oluşmaktadır. Aşağıda bu modüller açıklanmaktadır.

TEZ1.PRO: Menüler yardımıyla aylık veri girişlerinin yapıldığı ve günlük değerlere çevrildiği modüldür. Aylık bilgiler iki grup halinde girilir.

Birinci grup veri girişinde aşağıdaki bilgilerin girilmesi gerekmektedir:

- Kod numarası,
- Renk kodu,
- Eldeki süre içinde üretimi tamamlanıp depoya girmesi planlanan adet,
- Planlama döneminin bitiş tarihi.

İkinci grup veri girişinde ise renk detayı gerekmeksizin aşağıdaki bilgilerin girilmesi gerekmektedir:

- Kod numarası,
- Aylık planlanan alçık kalıp üretim adedi,
- Son üretim tarihi.

TEZ2.PRO: Bu modülde, programla ilgili sabit bazı tanımlamalar ve birkaç kural vardır.

TEZ3.PRO: Günlük gerçek verilerin girişi bu modülü kullanarak yapılmaktadır. İki ayrı grup veri girişi söz konusudur. Birinci grup veriler olarak aşağıdaki bilgilerin girilmesi gerekmektedir:

- Kod numarası,
- Renk kodu,
- Günlük depoya giren adet,
- Günlük sırlanan adet,
- Günlük fırına giren brüt adet.

İkinci grup veriler olarak aşağıdaki bilgilerin girilmesi gerekmektedir:

- Kod numarası,
- Günlük brüt döküm adedi,
- Günlük net döküm adedi,
- Günlük yarı mamul adedi,
- Günlük üretilen alçık kalıp adedi.

TEZAMAC.PRO: Bu modül yukarıda açıklanan üç modülün kullanımına izin veren menüyü içerir ve kullanıcının seçimine bağlı olarak programın akışını ilgili işlemlere yöneltir. Ayrıca programları doğru kullanabilmek için gerekli olan açıklamaları yaparak, yanlış kullanımları önlemeyi de amaçlar.

TEZ4.PRO:

Bir günlük üretimi değerlendirmek için gerekli olan verilerin bir kısmı, daha öncede açıklandığı gibi, diğer birkaç modul yardımıyla girilmektedir. Bu modülde elde bulunan bu verileri kullanarak gerekli hesaplamalar yapıp sonuçta o günkü üretimin planlara uygunluğu değerlendirilir. Bu modul şu verileri kullanır:

- Kod numarası,
- Renk kodu,
- Günlük depoya giren adet,
- Günlük depoya girmesi planlanan adet,
- Günlük birikimli olarak depoya girmesi planlanan adet,

Bu bilgileri kullanarak planlanan değerlerden sapma olup olmadığına bakılır. Bir sapma varsa yapılması gereken işlem, bu sapmanın hangi nedenlerle ortaya çıktığının bulunmasıdır. Bunun için üretim sürecinin başından sonuna kadar gerçekleştirilen tüm işlemlerle ilgili gerekli görülen bilgiler değerlendirilir. Bu bilgiler şunları içerir:

Günlük,

- Gerçek net döküm adedi,
- Gerçek brut döküm adedi,
- Planlanan brut döküm adedi,
- Planlanan net döküm adedi,
- Gerçek alçık kalıp üretim adedi,
- Gerçek döküm fire,
- Planlanan döküm fire,
- Planlanan fırın fire,
- Gerçek fırın fire,
- Fırın fire sapma,
- Birikimli depo girişi sapması,
- Brut döküm sapması,
- Alçık kalıp sapması,

- Net döküm sapması,
- Depo girişindeki sapma,
- Döküm firedeki sapma,
- Gerçek brut fırın girişi,
Gerçek birikimli döküm adedi,
- Gerçek birikimli brut döküm adedi,
- Gerçek birikimli alçık kalıp üretim
adedi,
- Planlanan birikimli brut döküm
adedi,
- Planlanan döküm adedi,
- Planlanan birikimli alçık kalıp üretim
adedi,
- Depo girişindeki birikimli sapma,
- Brut döküm için birikimli sapma,
- Birikimli net döküm üretimindeki sapma,
- Birikimli alçık kalıp üretimindeki sapma.

Bu modul yukarıda belirtilen değişkenlerin, üretim sırasında, planlandıkları gibi değer alıp almadıklarını tek tek kontrol ederek son durumu bildirir. Herhangi bir gün yapılan üretim içerisinde karşılaşılabilecek farklı durumlar sözkonusudur. Bu durumlar TEZ5 modülünde tanımlanmıştır. İhtiyaç duyulan değerler elde edildikten sonra bu kurallar taranarak karşı gelen durumlar bulunur ve ilgili açıklamalar yapılır. Her kuralın devamında gerekli açıklamalar bulunmaktadır.

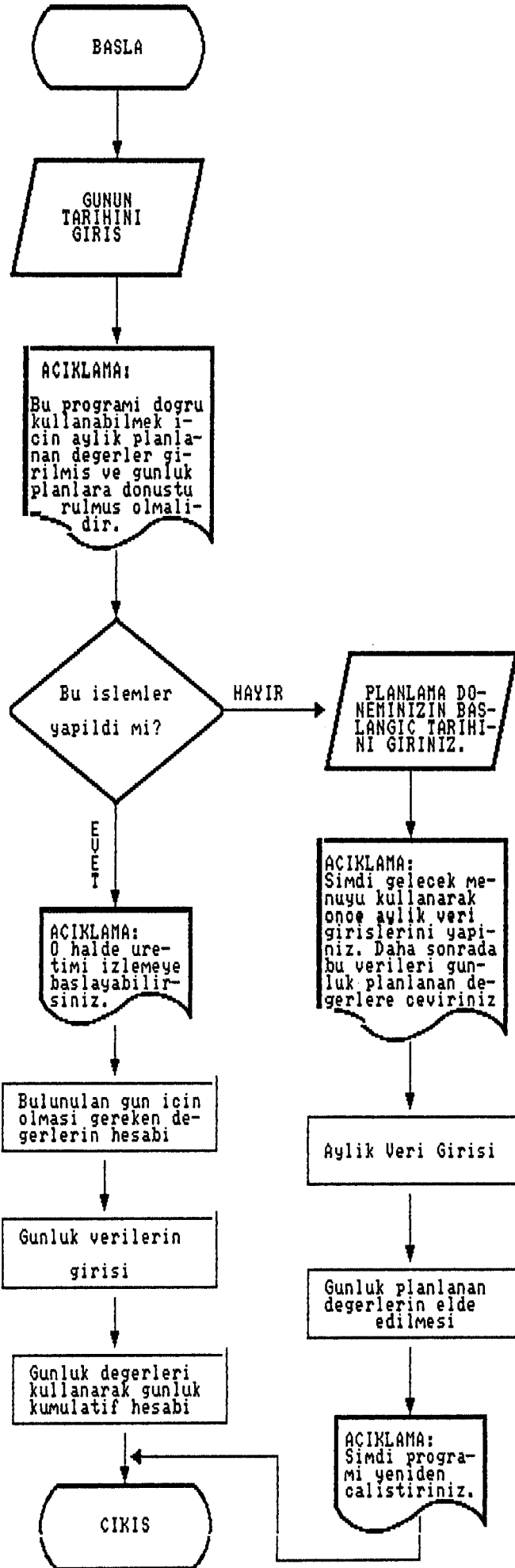
TEZ5.PRO: TEZ4.PRO'da açıklanan, herhangi bir gün için olabilir farklı durumlar, bu modülde tanımlanmıştır. Bu durumlar dört ayrı grupta toplanabilir. Herhangi bir üretim günü, planlardan sapma varsa bu dört temel gruptan birine uyar. Bu gruplar aşağıdaki gibi açıklanabilir:

- 1- Birikimli planlarda sapma varken, günlük üretim planlara uygundur,
- 2- Birikimli planlarda sapma yok, fakat günlük üretimde vardır,
- 3- Birikimli planlarda sapma görülmüştür, fakat birikimlilerdeki sapma o günkü üretimdeki sapmadan kaynaklanmaktadır.
- 4- Birikimli planlarda sapma var, günlük üretimde sapma var fakat birikimlilerdeki sapma o günkü üretimdeki sapma ile açıklanamamaktadır. Bu durumda önceki günlerdeki açık da henüz kapatılamamış demektir.

GLOBTEZ.PRO: İki veya daha fazla modülde kullanılan veri tabanları ve değişkenlerin yalnızca bir kere tanımlandığı programdır. Dolayısıyla bu program tüm modüller tarafından ortaklaşa kullanılır.

Şimdi de sistemin nasıl çalıştığı açıklanacaktır. Bazı işlemlerin yapılabilmesi için önceden yapılmış olması gereken birtakım hesaplamalar vardır. Bu gibi durumlarda program bu işlemlerin yapılıp yapılmadığını soracak ve programın akışını ona göre yönlendirecektir. Tüm işlemlerin öncesinde bir kez aylık verilerin girilmesi ve bu değerlerin günlük planlanan değerlere dönüştürülmesi gerekmektedir.

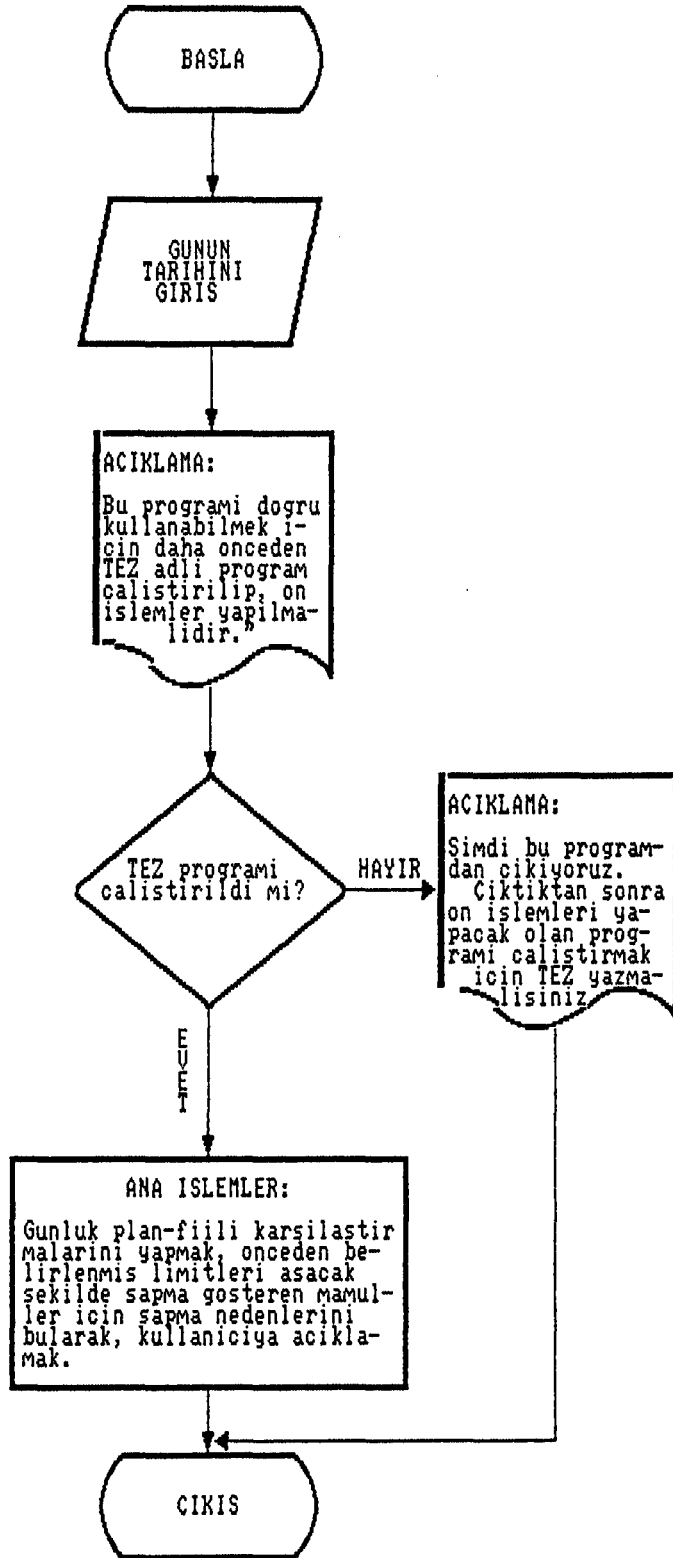
Hazırlanan iki programdan önce birincisi çalıştırılmalıdır. İkinci program çalıştırıldığında, daha önce ilk programın çalıştırılmış olması gerektiğini belirten bir uyarı ekrana gelecek böylece yanlışlıkla ikinci program çalıştırılmış bile olsa hiç bir işlem yapmadan geri dönmek mümkün olacaktır. Şekil 3.3. Tez adlı programının akış şemasını göstermektedir.



Şekil.3.3. TEZ adlı programın akış şeması

Buna göre önce aylık planlanan deęerlerin girilmesi ve gUnluk programlanan deęerlere dÖnÜştürülmesi işlemlerinin yapılıp yapılmadıęı kontrol edilmelidir. Bu işlemler yapılmıřsa program, o güne gelindięinde olması gereken birikimli deęerleri hesaplayacak, daha sonra gUnluk veri giriřinin yapılmasını isteyecektir. Girilen gUnluk deęerler hem bulunulan gUn için hem de o güne gelindięinde gerçekteşen birikimli adetler řeklinde tutulur. Bu işlem tamamlandıktan sonra programdan çıkıp ikinci program çalıştırılacaktır. Asıl deęerlendirme işlemi burada yapılacaktır. Gerekli ön işlemler yapılmamıřsa program bunların yapılabileceęi menüleri ekrana getirecek ve kullanıcıyı yönlendirecektir. Bu işlemler tamamlandıktan sonra program yeniden çalıştırılacaktır.

řekil 3.4. ise yukarıda bahsedilen ve gUnluk deęerlendirmenin yapılacaęı Tez2 adlı programının akıř řemasını göstermektedir.



Şekil.3.4. TEZ2 adlı programın akış şeması

Bu program daha öncede açıklandığı gibi önce birinci programın çalıştırılıp çalıştırılmadığını soracak çalıştırılmışsa işlemlere devam edecektir. Birinci programda elde edilmiş olan bütün değerleri kullanarak tanımlanmış kurallara göre üretimi değerlendirecek ve sonuçları verecektir. Ön işlemler yapılmamışsa, ekranda bu programdan çıkılacağını ve birinci programın çalıştırılması gerektiğini belirten bir mesaj görülecek ve programdan çıkılacaktır.

3.4.5. Test etme ve değerlendirme

Bu aşamada amaç, sistemin gerçekten amacına uygun olarak çalışıp çalışmadığını test etmek ve bu test etmenin sonuçlarını değerlendirmektir. Bunun için gerçek süreçten alınmış, ardışık bir kaç güne ait bilgiler sisteme girilmiş ve sistem çalıştırılmıştır. Sonuç olarak, sistemin değerlendirmesi ile elde edilen sonuçların beklendiği gibi olduğu görülmüştür. Aşağıda bu konudaki işlemler ve elde edilen sonuçların değerlendirmesi açıklanmaktadır.

3.4.5.1. Programın gerçek verilerle çalıştırılması

Örnek olarak 6280 kod nolu klozetin beyaz rengi için yapılan işlemler ve sonuçları aşağıdaki gibidir:

Gerekli hesaplamaların yapılması: Bu kesim, iki ayrı alt grupta ele alınacaktır. Birincisi, aylık olarak, belirtilen adette üretimi yapabilmek için, sürecin diğer safhalarında üretilmesi gereken yarı mamul adetlerinin planlanması, ikincisi ise aylık olarak gerek son ürün gerekse safha temelinde elde edilen değerlerin, günlük programlara dönüştürülmesi işlemlerine yöneliktir.

- i) Aylık planlanan üretim değerlerinden hareketle, bu üretimi yapabilmek için her safhada üretilmesi gereken yarı mamul adetlerinin bulunması

Öncelikle üretimi izlenecek olan mamulün hangi tarihe kadar ve kaç adet üretilmek istendiği programa girilmelidir. 1991 yılının Temmuz ayına ait verilerle 6280 kod nolu mamulün beyaz rengi için aylık toplam olarak üretilip depoya girmesi planlanan adet 5322' dir. Daha sonra yapılması gereken işlem 5322 adet mamul elde edebilmek için sürecin başından itibaren her bir aşamada olması gereken değerlerin hesaplanmasıdır.

Dökümü yapıp kurutulmuş mamullerin bir kısmı bu aşamada çeşitli nedenlerle fire olmaktadır. Geriye kalan net dökümün hepsinin sızlanıp fırına girdiği varsayılmaktadır. Sırlama ve fırına girme sırasında olabilecek fireler (kırılma vb.) ihmal edilecek kadar azdır. Bazı mamuller ise kötü sızlandığından veya bozuk sır malzemesi kullanıldığından dolayı iskartaya ayrılabilir. Böyle mamullerin bozukluğu ancak fırından çıktıktan sonra belli olmaktadır. Sırlı haldeyken anlaşılabilir. Bu tür fireler fırın çıkışından sonra çeşitli nedenlerle olabilecek ürün kayıplarıyla birlikte fırın fire olarak adlandırılmaktadır.

6280 nolu mamulün beyaz rengi için planlanan fırın fire değeri %35, döküm fire değeri ise %11' dir. O halde, fırından çıkan mamullerin iskarta olanları ayrıldıktan sonra geriye toplam olarak 5322 adet net mamul kalmalıdır. Buna göre,

$$(\text{fırından çıkan brüt adet}) \times 0.65 = 5322$$

olmalıdır. Buradan,

$$5322 / 0.65 = 8188$$

adet mamul sızlanıp fırına girmelidir.

Sırlama aşamasına, dökülüp döküm fireleri ayrıldıktan sonra elde kalan net dökümler geldiğine göre,

$$(\text{Brüt döküm adedi}) \times (1 - \text{Döküm fire}) = \text{Net döküm adedi}$$

=Sırlanacak adet

$$\Rightarrow \text{Brut döküm adedi} = (\text{Net döküm adedi}) / (1 - \text{Döküm fire})$$

$$\Rightarrow 8188 / 0.89 = 9200$$

olarak bulunmaktadır.

Sonuç olarak, Temmuz ayında 5322 adet 6280 beyaz üretimi için; toplam 9200 adet brut döküm yapılmalı, bu dökümlerden 8188 adet net döküm elde edilmelidir. Daha sonraki sırlama, fırın ve kalite ayırım aşamalarından sonra ise toplam 5322 adetlik net üretime ulaşabilmek için fırın fire değeri %35'i aşmamalıdır.

ii) Aylık planlanan değerlerin, günlük programlara dönüştürülmesi

Aylık olarak tüm safhalarda olması gereken değerler planlandıktan sonra, her safhanın bu toplam değerleri günlere dağıtılmaktadır. Bu işlemi yaparken her bir değer, ilgili safha için elde bulunan gün sayısına bölünür. İlgilenilen örnek mamul için üretim periyodu 2 Temmuz-31 Temmuz tarihleri arasındadır. Her bölümün çalıştığı gün sayısı farklıdır. Bazı bölümler haftada 5 gün, bazı bölümler 6 gün, bazı bölümler ise 7 gün çalışmaktadırlar. Belirtilen zaman dilimi içerisinde dökümhane toplam 22 gün, kalite ayırım bölümü de yine toplam 22 gün çalışacaktır (Fırından çıkan mamuller kalite ayırım işlemine tabi tutulmadan depoya giremeyeceği için, günlük depoya girmesi gereken adet planlanırken, kalite ayırım çalışma günü esas alınmaktadır).

Buna göre,

Bir döküm gününde $9200/22=418$ adet brut döküm yapıлып,

$8188/22=372$ adet net döküm elde edilmeli, dolayısıyla

372 adet mamul sırlanıp fırına girmelidir.

Uretimi tamamlanıp depoya girmesi gereken mamul adedi ise,

$$5322/22=242$$

adet/gun olmalıdır.

Bu şekilde günlük programlanan deęerler tüm mamuller ve her safha için tek tek hesaplanır. Günlük programlanan deęerler aslında iki grup halinde elde edilmektedir. Birincisi yukarıda açıklandığı gibi o gün gerçekleşen üretimde olması gereken deęerler, dięeri ise o güne gelindiğinde birikimli olarak olması gereken deęerlerdir. Program her iki grup için planlanmış ve programlanmış deęeri hesaplamaktadır.

Bir üretim gününde dökülen mamuller o gün sırlanıp fırına girmekte fakat fırında bir gün kalmaktadırlar. Bu nedenle, yarı mamullerin fırından çıkıp bitmiş ürün olarak depoya girişleri, döküldüklerinin ertesi günü yapılmaktadır. 3 Temmuz 1991 tarihindeki üretime bakıldığında, bu açıklamalara göre üretimin başladığı 2 Temmuz tarihinde henüz depoya mamul girişi olmayacaktır.

Çizelge.3.1.(a). 6280 kod numaralı mamulun beyaz rengi için gerçek üretim sürecinden alınmış örnek verileri göstermektedir. Gerçek deęerlerin yanısıra olması planlanan deęerler de yine aynı çizelgede görülmektedir.

Çizelge.3.1.(a) 6280 kod numaralı mamulün beyaz rengi için gerçek veriler

	2 Temmuz		3 Temmuz	
GUNLUK DEĞERLER				
	Plan	Gerçek	Plan	Gerçek
Brut döküm adedi	418	367	418	148
Döküm fire oranı	0.11	0.019	0.11	0.006
Net döküm adedi	372	360	372	147
Sırlanan adet	372	360	372	147
Fırına giren adet	372	360	372	147
Fırından çıkan adet	0	0	242	360
Fırın fire oranı	0	0	0.35	0.50
Net üretim adedi	0	0	242	180

Çizelge.3.1.(b) ise aynı örneğe ilişkin birikimli değerleri göstermektedir. Herhangi bir gün için sadece günlük değerler değil aynı zamanda o güne gelindiğinde olması gereken birikimli değerler de üretimi izlemek amacıyla elde edilmektedir.

Çizelge.3.1.(b) 6280 kod numaralı mamulün beyaz rengi için gerçek veriler

	2 Temmuz		3 Temmuz	
BIRIKIMLI DEĞERLER				
	Plan	Gerçek	Plan	Gerçek
Brut döküm adedi	418	367	836	515
Net döküm adedi	372	360	744	732
Sırlanan adet	372	360	744	507
Fırına giren adet	372	360	744	507
Fırından çıkan adet	0	0	372	360
Net üretim adedi	0	0	242	180

3.4.5.2. Değerlendirme

İşletmede günlük üretim değerlendirilirken öncelikle günlük ve birikimli depo girişi değerlerine bakılır. 3 Temmuz 1991 tarihi, üretimin ikinci günüdür. Bu tarih aynı zamanda depoya da bitmiş mamullerin girmeye başlayacağı ilk gündür.

Mevcut durumda uzman kişi yukarıdaki değerleri çeşitli raporlardan elde edecektir. 3 Temmuzda depoya 242 adet mamul girmesi gerekirken bu değer 180 olarak gerçekleşmiştir. Bu tarih için bu değer, depoya girmesi gereken adet bakımından birikimli değerlerde de aynı oranda bir sapma demektir. Çünkü bugün depoya girmesi gereken mamul adedi birikimli planlarda ve günlük planlarda aynı değere sahiptir. Böyle bir durumda yapılacak işlem birikimlilerdeki sapmanın bugünkü sapma ile açıklanıp açıklanamadığını araştırmaktır. Bir başka ifadeyle eğer bugün, programlara uygun adetlerde mamulün depoya girişi

yapılmış olsaydı, birikimlilerdeki sapma ortadan kalkacak mıydı sorusuna cevap aranır. Uzman kişi birikimlilerdeki sapma bugünden kaynaklanıyorsa nedenlerini arařtırmak için yalnızca bugün üretimi tamamlanan mamullerin üretim süreci ile ilgilenmesi gerektiğini bilir. Bugün için de böyle bir durum vardır.

Bugün depoya giren mamul sayısı planlananın 60 adet gerisindedir. Nedenlerini arařtırırken dünkü üretim süreci deęerleri incelenecektir. Bu deęerlere bakılırsa,

2 Temmuz tarihinde 6280 kod nolu mamulün beyaz rengi için,

brüt döküm adedi=418

net döküm adedi=360

fırına giren mamul adedi=360

olarak gerekleşmiştir.

3 Temmuz tarihi (ertesi gün) için deęerler ise şöyledir:

fırından çıkan mamul adedi=360

depoya giren mamul adedi=180

Sürecin ilk aşaması olan döküm aşamasında günlük 418 adet brüt döküm yapılması gerektiği halde bu deęer 367 olarak gerekleşmiştir. Ertesi günkü eksik depo girişinde bir neden bu olabilir. Döküm firenin planlanandan daha az olmasına rağmen başlangıçtaki brüt döküm eksiginden dolayı net döküm adedi de yine planın gerisinde kalmıştır (plan 372, gerek 360). Fırın fire deęerini bulabilmek için bugünkü raporlara bakmak gerekmektedir. Çünkü ilgili mamul fırından bugün çıktığı için fırın fire deęeri de ancak bugün öğrenilebilmektedir. Sonuç olarak bugün fırından çıkan mamul adedi 360 iken depoya giren mamul adedinin 180 birim olması aradaki fark kadar mamulün sırlama aşamasından sonraki toplam fire olduğu anlamına gelmektedir. Daha önce

bu deęerin fırın fire olarak tanımlandığı belirtilmişti. Burada %50'lik (1-180/360), bir fire deęeri gerekleşmiştir. Oysa fırın fire için planlanan deęer %35 idi.

Bu açıklamalara göre, 6280 nolu mamulun beyaz rengi için bugünkü depo girişinin programlara uygun olmayışının nedenleri olarak, dün dökümün eksik yapılması buna baęlı olarak elde edilen net döküm adedinin azalması ve fırın aşamasında beklenenden fazla fire olması söylenebilir.

Buraya kadar açıklananlar bugün üretimi tamamlanan mamullerle ilgili deęerlendirmelerdir. Ayrıca döküme bugün başlanıp bugün fırına girecek ve yarın üretimi tamamlanacak olan mamullerle ilgili olarak da bugün başlayan üretimlerinin deęerlendirilmesi yapılmaktadır. Buna göre söylenmesi gereken aksaklıklar şunlardır:

Bugünkü gerek döküm adedi de dünkü gibi planın gerisinde kalmış (148<418), bu durum zincirleme olarak net döküm eksigine ve fırına eksik mamul girişine yol açmıştır. Bu nedenle 6280 nolu mamulun beyaz rengi için yarın üretimi tamamlanıp depoya girecek olan mamul adedinde planlardan sapma görülebilir.

Genelde hazırlanmasına çalışılan sistem sadece mevcut durumu izlemeye ve aksaklıkların nedenlerini bulup kullanıcıya açıklamaya yöneliktir. Bu yönüyle herhangi bir şekilde planları deęiştirme, bazı deęerleri güncleme olayı çalışmanın kapsamına dahil değildir. Fakat kullanıcı bu nedenleri bilirse, o aşamalarda daha dikkatli olunması konusunda ilgili üretim bölümlerine uyarılarda bulunarak, benzeri aksaklıkların izleyen günlerde tekrarlamasının önüne geçebilir. Bu uyarı daha önce dikkatleri çekmemiş bir konuda olup örneğin üretim aşamasındaki kişileri fırın firenin neden fazla olduğu sorusuna cevap araştırmaya da yöneltebilir.

Buraya kadar elle sınıanan günlük üretimin aynı verilerle uzman sistem tarafından ne şekilde değerlendirildiğine bakılırsa aşağıdaki bilgisayar çıktıları elde edilecektir.

GUNLUK URETİM DEĞERLENDİRME SONUÇLARI

İno Renk
 --- ----
 0 beyaz

Bu kod için bugün depoya girişi yapılan mamul adedi, planlananın içinde kalmıştır. Üstelik bugüne gelindiğinde kümülatif olarak da etimi tamamlanıp depoya girmiş olması gereken mamul adedi plana göre daha azdır. Fakat kümülatiflerdeki sapmanın yalnızca bugün günlük üretimdeki sapmadan kaynaklandığı anlaşılmıştır. Buna göre bugün günlük üretimde kümülatifleri etkileyecek oranda bir sapmanın oluştuğu ylenebilir.

BUGUN İÇİN EKSİK DEPO GİRİŞİNİN NEDENLERİ

Bugün bitmiş olarak depoya giren mamullerin üretimine dün başlandı-ndan, sapmaların nedenlerini de dünkü fiili üretim değerlerini inceyerek araştırmalıyız. Bu inceleme için 6280 nolu mamulün dün 4 adet döküldüğüne, bunlardan kaç tane net döküm elde edildine ve ger aşamalara bakmalıyız.

Bu değerlendirmelerin sonuçları aşağıdaki gibidir:

fen herhangi bir tuşa basınız...

GUNLUK URETİM DEĞERLENDİRME SONUÇLARI

Bugün 6280 nolu mamulün beyaz renginden üretimi tamamlanarak depoya girmesi gereken adet 242 olmalıydı. Fakat bugün bu değer 180 olarak gerçekleşmiştir.

utfen herhangi bir tuşa basınız...

PROLOG2>tez2

SIRALAMA MODULU

Kodno	Renk
6280	beyaz

utfen herhangi bir tuşa basınız...

Planlanan brüt döküm adedi 418 iken 367 adet döküm yapılmıştır.

Sapma oranı=0.12200956938

Planlanan net döküm adedi 372 iken 360 adet net döküm elde edilmiştir.

Sapma oranı=0.032258064516

Bu kod için fırın fire planlanana aşmıştır Planlanan fırın fire değeri 0.35 olarak belirlenmişti. Oysa fiili fırın fire değeri 0.5 dir.

Sapma oranı=0.3

utfen herhangi bir tuşa basınız...

GUNLUK URETİM DEĞERLENDİRME SONUÇLARI

Adno Renk

30 beyaz

Bugün dökümü yapıлып, yarın tamamlanacak olan mamuller için bugünkü üretim değerlendirmesi ise şöyledir:

Bugün 148 adet brüt döküm yapılmıştır. Oysa bu değer 418 olarak planlanmıştı.

Elde edilen net döküm adedi planlanandan azdır.
Günlük net döküm adedi 372 olmalıydı. Oysa bugünkü net döküm adedi 147 olarak gerçekleşmiştir. Bu durum yarın bu kod için elde edilecek mamul adedinin azalmasına yol açabilir.

Lütfen herhangi bir tuşa basınız...

Bu konuda uzman olmayan birisi benzeri durumlarda nasıl bir değerlendirme yapabilirdi? Bunu açıklamak için şöyle bir örnek alınabilir:

İlgilenilecek veriler Çizelge 3.2.'deki gibi alınarak, bu veriler karşısında 3 Temmuz tarihinde 226 adet mamulun depoya girdiği görülüp bu değer planlanan değerden az olduğu için nedeni araştırılacak olsun.

Çizelge 3.2. Günlük üretim değerlendirmesi için örnek veriler

	2 Temmuz		3 Temmuz	
GÜNLÜK DEĞERLER				
	Plan	Gerçek	Plan	Gerçek
Brüt döküm adedi	418	380	-	-
Döküm fire oranı	0.11	0.006	-	-
Net döküm adedi	372	377	-	-
Sırlanan adet	372	377	-	-
Fırına giren adet	372	377	-	-
Fırından çıkan adet	-	-	372	377
Fırın fire oranı	0.35	-	-	0.40
Net üretim adedi	-	-	242	226

Böyle bir durumda uzman olmayan kişinin rastgele olarak önce fırından kaç adet mamul çıktığına baktığını düşünelim. Bu değer 377' dir ve planlanan değere ulaşmıştır (planlanan değer=372). Bu durumda sürecin daha geride kalan aşamalarına bakmayabilir ve o günkü üretimi değerlendirirken fırından 377 adet mamul çıktığı halde (buraya kadar her şeyin normal gittiğini zannederek) depo girişi 226 adet olduğundan tek sorunun fırın fire fazlalığı olduğunu düşünebilecektir. O günkü aksaklığın gerçekte en önemli

nedeni fırın fire fazlalığıdır. Fakat uzman, süreci değerlendirmeye ilk aşamadan başlayarak eksik brut döküm yapıldığını da yakalar. Bunu, izleyen günler için bir uyarı olarak açıklar ve gerekirse tekrarlanmaması için önlemler alınmasını sağlar. Özetle dökümhanede bir problemin varlığını yakalar. Brut döküm eksigi o gün için tesadufi olarak döküm firenin planlanandan az olması nedeniyle hissedilmemiş ve diğer aşamalara yansımamıştır. Bu ayrıntıyı uzman kaçırmaz, fakat herhangi birisi yakalayamayabilir. Uzman kişi önceki günlerde aynı süreçte yaşadığı tecrübelerini vb. bilgisini kullanarak bu metodolojiyi kafasına yerleştirmiştir ve onu sistemin diğer bireylerinden farklı kılan da bu metodolojidir.

3.4.6. Yaşatma ve bakım planlama

Geliştirilen bu sistem, ilgili işletmeye önerilmiş, fakat uygulanmasına yönelik herhangi bir sorumluluk alınmamıştır. Fakat sistemin ileride kullanılması olasılığını düşünerek, bakım planlaması aşamasına yönelik olarak nelerin yapılabileceğini belirtmek yerinde olacaktır.

Sistemin kullanılması sırasında, her an sistem üzerinde bir değişikliğe, yeniliğe ihtiyaç olabilir. Bu ihtiyaç, gerçek sistemi etkileyen içsel ve/veya dışsal değişimlerden kaynaklanabilir. Geliştirilen sistemin amacı, gerçek sistemi temsil etmek olduğuna göre, gerçek sistemi etkileyebilecek bu değişimlerin geliştirilen sisteme de olabildiğince yansıtılması gereklidir. Bunun gerçekleştirilmesi ise, ancak o konunun uzmanı olan kişinin desteğiyle mümkün olur. Bu nedenle bakım planlaması sırasında ihtiyaç duyulabilecek olan uzman kişinin, halen aynı işletmede görevini sürdürdüğünü belirtmek yerinde olacaktır.

3.5. Geliştirilen Sistemin Değerlendirilmesi

Ele alınan uzman sistemin hangi aşamalardan geçerek, nasıl geliştirildiği açıklanmış ve geliştirilen sistem tanıtılmıştır. Daha sonra gerçekleştirilen örnek çalıştırmada ise programa aktarılmasına çalışılan kavramlar şöyle açıklanabilir:

- Uretim sürecindeki aksaklıkların kaynağını araştırırken belli bir mantıksal akış izleyip, değerlendirmeye sürecin ilk aşamasından başlama alışkanlığı.

Bu mantıksal akışın, seçilen aracı kullanarak uygun değişken ve kural tanımlamaları ile sisteme de kazandırılmasına çalışılmıştır.

- Herhangi bir gün hem birikimli hem de günlük planlarda sapma varsa ve birikimlilerdeki sapma, o günü sapma ile açıklanabiliyorsa problemleri yalnızca o günü üretim değerlerinde aramanın yeterli olacağını bilmek.

Bu konuda uzman olmayan birisi, ilk anda, günlük üretime ilişkin tüm değerleri alıp incelemeyi doğru bulabilir. Fakat gerçekte, bazı durumlarda, eldeki tüm bilgilere ihtiyaç duyulmaksızın, doğru seçilmiş bir kaç değer, bir aksaklığın nedenlerini bulmaya ve yorumlamaya yeterli olabilir. Buradaki durumda da, birikimli değerlerde plan-gerçek sapması görüldü diye, eldeki tüm (her safha için) birikimli plan değerlerini incelemek gerekemeyebilir. Günlük üretimdeki sapma bunu açıklıyor olabilir.

Gerçek süreçte yukarıda açıklananlardan başka durumlarla da karşılaşılabilir. Her birisi için izlenecek yol (uzmanın yaklaşımı) farklı olabilir. İşte, çalışmada da bu şekilde karşılaşılabilen farklı durumların ve uzmanın bunlara karşı gelen yaklaşımlarının sisteme olabildiğince dahil edilmesine çalışılmıştır.

Bu deęerlendirmelerin klasik yazılımlarla yapılması çok gu ya da mümkün olamamaktadır. unku, klasik yazılımlar; uzman sistemlerde, uzmanlık bilgisini temsil eden kurallara karşı gelecek tanımlamaları ve mantıksal akışı saęlamak için yeterli özelliklere sahip değildirler.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, endüstri mühendislerinin temel ilgi alanlarından olan üretim sürecinden hareketle, bir işletmede üretimi izlemek için bir uzman sistem geliştirmek, bunun öncesinde uzman sistemi kendi içinde tanımlayıp, temel özelliklerini araştırmak, ayrıca uzman sistemlerin Endüstri Mühendisliği ve Yöneylem Araştırması ile olan ilişkilerini örneklemek amaçlanmıştır. Bu nedenle çalışmanın sonuçları da bu amaçları dikkate alarak bir kaç grup içinde açıklanabilir.

Öncelikle uzman sistemler konusunun, özellikle klasik yazılımlardan olan farklılıklar yönüyle dikkati çektiği söylenebilir. Bu farklılıklar; analiz ve gerçekleştirilmede bağımlılık, karar verme sürecindeki yeri, yürütmedeki katkı, yapısal özellik ve yaşatma ve bakım olmak üzere beş grupta toplanmışlardır.

Çalışmayı daha genel anlamda Endüstri Mühendisliği, Yöneylem Araştırması ve uzman sistemler etkileşimi açısından değerlendirmek gerektiğinde ise, şu sonuçlara ulaşılmaktadır: Yöneylem Araştırması ve Yapay Zeka (bağlı olarak uzman sistemler) bilim dalları arasında paylaşılan temel amaç, karar verme ve problem çözmeyi destekleyici yöntemler geliştirmektir. Bu işi yaparken biri diğerinin ele alamadığı ya da yeterince gözönünde bulundurmadığı (Yöneylem Araştırması'nın çoğunlukla algoritmik ve niceliksel bir yaklaşım izlemesi, uzman sistemlerin ise sezgisel ve niteliksel bir yaklaşıma daha fazla önem vermesi) özellikleri

çözümüne yansıtılabilmektedir. Endüstri Mühendisliği ve uzman sistemler etkileşimine bakıldığında uç alandaki ilişkiler dikkati çekmektedir. Endüstri Mühendisliği bilgileri ile bir uzman sistem oluşturulup, diğer kullanıcılara endüstri mühendisi gibi danışmanlık hizmeti verecek bir sistem geliştirilebileceği gibi; uzman sistem, bir endüstri mühendisinin ilgilenebileceği herhangi bir konuda tasarlanıp, endüstri mühendisi için danışmanlık hizmeti de verebilir. İkinci olarak, uzman sistemler karmaşık yönetsel problemleri modellemede endüstri mühendislerine yardımcı bir araç olarak kullanılabilirler. Üçüncü ve son etkileşim alanı ise bir öğretici olarak uzman sistemler konusudur. Uzmanın bilgisi önce bilgisayara buradan da uzman olmayan kullanıcıya aktarılmaktadır.

İleride uzman sistemler, endüstri mühendisleri için, belkide, lisans eğitimleri sırasında Yöneylem Araştırması teknikleri ile birlikte ele alınacak bir konu olacaklardır. Bu uç bilim dalının mevcut durumdaki eğilimine bakarak, izleyen yıllarda problemlerin çözümünde birlikte kullanımlarının daha da yaygınlaşması ve bu gelişmelere bağlı olarak, problemlerin çözümündeki etkinliğin de artması beklenebilir.

Geliştirilen uzman sistem ile ilgili bir değerlendirme yapılırsa:

Uygulamanın yapıldığı işletmedeki mevcut sistemde yine bilgisayar aracılığıyla veri girişleri yapıp, türetilen günlük raporlar uzman kişi tarafından yorumlanmaktadır. Sadece çok sayıda sayısal bilgiyi içeren ve konuşma dili ile yazılmış herhangi bir açıklayıcı mesaj, uyarı vb. içermeyen bu raporları ancak o konudaki uzman kişinin yorumlaması mümkün olmaktadır. Raporları yorumlarken uzman, hangi aşamaların birbirini izlediğini, sırasıyla nelere bakması gerektiğini, hangi bilgilere ihtiyacı olduğunu bilir. Bu

değerlendirmeyi nasıl yapması gerektiğini bilmeyen birisi için, günlük üretim raporları fazla bir anlam ifade etmeyecektir. Hazırlanan sistemde bir uzmanın günlük üretimi değerlendirmek için ihtiyaç duyduğu tüm bilgiler, bunları incelerken izlediği yol ve varabileceği sonuçlar programlanmıştır. Böylece, üretimi değerlendirmeye nereden başlanacağı, hangi raporlara bakmanın gerekeceği ve sapmalar karşısında nedenlerinin nerelerde aranıp, nasıl yorumlanacağı konusundaki bilgilere ihtiyaç kalmaz. Tüm bu işlemler, ilgili mantık sürecinin programa aktarılmasıyla kullanıcıdan bağımsız olarak yapılıp, sonuçlar elde edilmektedir. Kullanıcıya düşen görev sadece günlük gerçek değerleri girmektir. Çünkü program günlük programlanan değerlerle karşılaştırmak için bu bilgilere ihtiyaç duyacaktır.

Mevcut sistem ile, bilgisayardan alınan çok sayıda rapor tek tek değerlendirilmekte, bazen bir rapordaki herhangi bir değer yorumu için diğer raporlardaki başka bilgilere de ihtiyaç olmaktadır. Gerçekleştirilen sistem ile, gerekli olan tüm veriler, daha önceden oluşturulup, aylık ve günlük planlanan değerlerle, gerçekleşen değerlerin girildiği veri tabanlarından alınmakta ve daha kısa sürede değerlendirilmektedir. Çünkü artık uzmanın raporları bizzat inceleyip, gerekli verileri arama ve bu veriler üzerinde düşünme işlemleri ortadan kalkmıştır. Sistem hangi verilere ihtiyacı olduğunu bilip, bunları veri tabanlarından bulmakta ve bu verileri bir uzman gibi yorumlayıp değerlendirmesinin sonuçlarını kullanıcıya sunmaktadır. İşletmenin tırettiği günlük üretim raporları ise herhangi bir yorum içermeyip, yalnızca sayısal bilgileri göstermektedir. Bu çalışmada sonuçların günlük konuşma dili ile yazılı olarak ifade edilmesi ile anlaşılabilirliği arttırmak istenmiştir.

Üretimi değerlendirme işleminin sonuçlarının böyle bir sistemden alınması ile insanın yapabileceği hataları azaltmak

da mümkün olacaktır. Raporlarda yanlış değerlere bakmak önlenemez, uzmanın hastalık vb. nedenlerle bilgi ve tecrübelerini verimli kullanamaması gibi sakıncalar ortadan kalkacaktır.

Uzman kişinin benzeri işlemleri yaparken izlediği mantığın bir kez programa aktarılmasıyla, artık aynı sonuçların uzman olmayan kişiler tarafından da sistemden alınması sağlanmıştır. Böylece hem tek bir kişiye bağımlı olma durumu ortadan kalkmış hem de uzman kişinin boşalan zamanını bilgi ve tecrübelerini kullanabileceği başka alanlara kaydırma fırsatı doğmuştur.

Geliştirilen sistem açısından, geleceğe yönelik eğilime değinmek gerekirse, bu konu, kısmi olarak, çalışmanın bakım planlaması bölümünde açıklanmıştır. İçsel ve/veya dışsal etkenlerden dolayı gerçek sistemde gerekli olabilecek değişikliklerin, gerçek sistemi temsil eden uzman sisteme de yansıtılması gereğinden hareketle, uzman sistemin kullanımı sırasında, uzman kişiye ihtiyacın sürekli olduğunu belirtmek yerinde olacaktır. Çünkü bu sistemin gerçekleştirilmesinde uzmanın rolü büyüktür, büyük ölçüde bu kişinin bilgisine başvurulmuştur ve bağılı olarak ileride gerekli görülebilecek değişikliklerin uzman sisteme nasıl yansıtılacağı konusunda yine uzmana ihtiyaç doğacaktır.

Bu çalışma, günlük üretimin planlara uygun olarak gerçekleşip gerçekleşmediğini bir dizi plan-gerçek karşılaştırması yaparak sınıyıp, bir sapma varsa bunun nedenlerini bulmak ve kullanıcıya sunmakla sınırlanmıştır. Fakat sistem böyle kalmamalı, diğer çalışmalarda yapılmalıdır. Ürünün bu sistemin çıktılarını kullanarak üretimi kontrol eden bir sistem tasarlanabilir. Geliştirilen sistem, süreçte nerelerde problemlerin olduğunu bulup, planlardan hangi oranlarda uzaklaşıldığını açıklarken bu bilgileri girdi olarak alan bir üretim kontrol sistemi, bu

açığı kapatacak şekilde planları güncleyebilir. Daha sonra bu yeni değerler, yine çalışmaya konu olan sisteme planlanmış değerler olarak girilip böylece bu iki sistemin bütünleşik olarak çalışması sağlanabilir.

KAYNAKLAR DIZINI

- Azazi S.**, 1990, Yapay Zeka ve Uzman Sistemler, Koç Unisys' den Haberler, 6-8.
- Badıru A. B.**, 1988, Expert Systems and Industrial Engineers: A Practical Guide to a Successful Partnership, Computers and Ind. Engineering, 14, 1, 1-3.
- Baray M., Doğaç A., Can F.**, 1986, 5. Kuşak Bilgisayar Sistemlerine Genel Bakış, 3. Türkiye Bilgisayar Kongresi, 21.
- Biegel J. E., Wink J. L.**, 1989, Expert systems Can Do Job Shop Scheduling: An Exploration and A Proposal, Computers and Ind. Engineering, 17, 1-4, 347-352.
- Daglı C. H.**, 1986, Uzman Sistemlerin Üretim Planlamadaki Yeri, 2. Ulusal Makina Tasarım ve İmalat Kongresi Bildiri Kitabı, 17-23.
- Daglı C. H., Stacey R.**, 1988, A Prototype Expert Systems for Selecting Control Charts, Int. Journal of Prod. Res., 26, 5, 987-996.
- Demirci E., Dinçmen M.**, 1989, Atölye Tipi Üretim Sisteminin Etkileşimli Benzetimi, Yöneylem Araştırması Bildiriler'89 Kitabı, Meteksan A.Ş., Ankara, 227-228,341.
- Doukidis G. I., Ray J. P.**, 1990, A Survey of the AI Techniques within the OR Society, J. Opl. Res. Soc., 41, 5, 363-375.
- Edosomwan J. A.**, 1987, Ten Design Rules for Knowledge Based Expert Systems, Industrial Engineering, 19, 1, 78-80.
- Gupta Y. P., Chin D. C. W.**, 1989, Expert Systems and Their Applications in Production and Operations Management, Computers and Op. Res., 16, 6, 567-582.
- Harmon P., King D.**, 1985, AI In Business ES, John Wiles and Sons Inc., USA, 283.

- Kanet J. J., Adelsberger H. H., 1987, Expert Sytems in Production Scheduling, European Journal of Op. Res., 29, 51-59.**
- Kara I., 1985, Yöneylem Araştırmasının Yöntembilimi, Anadolu Üniversitesi Basımevi, Eskişehir, 117.**
- Korur E., 1991, Bilgisayarlar İnsan Gibi Düşünmeyi Öğrenebilecek mi?, Bilim ve Teknik, 24,285,25-28 (Siebel M.'den çeviri).**
- Koster A., Raafat F., 1990, The Application of a Knowledge Based Expert Support System to Workers' Compensation Insurance, Computers and Ind. Engineering, 18, 2, 133-143.**
- Krepchin I. P., 1987, A New Tool for Planning and Control, Materials Handling, 42, 1-6, 105.**
- Kumara S. R. T., Kashyap R. L., Moodie C. L., 1988, Application of Expert Systems and Pattern Recognition Methodologies to Facilities Layout Planning, Int. Journal of Prod. Res., 26, 5, 905-930.**
- Kusiak A., 1987, An Expert System for Group Technology, Industrial Engineering, 19, 1-12, 56-61.**
- Levine R. I., Drang D. E., Edelson B., 1988, A Comprehensive Guide to AI and ES Using Turbo Pascal, McGraw-Hill Book Co., Singapore, 257.**
- Malpas J., 1987, Prolog A Relational Language and Its Applications, Prentice-Hall, New Jersey, 465.**
- Palmer R. G., 1986, How Expert Systems Can Improve Crop Production, Agricultural Engineering, 67, 6, 28-29.**
- Parrish A. JR., 1989, A Foreward to Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering, 1, 1, 5-7.**
- Rich K., Robinson P. R. , 1988, Using Turbo Prolog, McGraw-Hill Inc., USA, 253.**
- Roth F. H., Waterman D. A., Lenat D. B., 1983, Building Expert Systems, Addison-Wesley Publishing Company Inc., Canada, 444.**

- Schildt H., 1987, Advanced Turbo Prolog, McGraw-Hill, California, 299.
- Scown J.S., 1985, The Artificial Intelligence Experience: An Introduction, Digital Equipment Corporation, USA, 183.
- Sinha D., Ghiaseddin N., Matta K., 1989, Expert Systems for Inventory Control Management, Computers and Ind. Engineering, 17, 1-4, 425-429.
- Son Y. K., 1990, A Decision Support System for Factory Automation: A Case Study, Int. Journal of Prod. Res., 29, 7, 1461-1473.
- Suraj M. A., 1985, Knowledge Based Expert Systems: Their Applications to Production Management, Production and Inventory Management, 26, 4, 109-114.
- Taşkın H., Türker E.S., 1989, Endüstriyel Sistemlerde Yapay Zeka ve Uzman Sistem Uygulamaları, (Endüstri Mühendisliği Dergisi'ne gönderildi), 21.
- Teng S., Black J. T., 1989, Expert Systems for Manufacturing Cell Control, Computers and Ind. Engineering, 17, 1-4, 18-23.
- Timuroglu F., 1988, Yapay Zeka ve Uzman Sistemler, Bilgisayar Dergisi, 27.
- Turban E., 1986, Expert Systems-Another Frontier for Industrial Engineering, Computers and Ind. Engineering, 10, 3, 227-235.
- Turban E., 1990, Decision Support and Expert Systems, MacMillan Publishing Company, USA, 843.
- Ulengin B., 1987, Yapay Zeka ve Uzman Sistemler, XI. Ulusal YA Kongresi Bildiri Kitabı, VIII-31.
- Watts D. S., Eldin H. K., 1987, The Role of the Industrial Engineer in Developing Expert Systems, Computers and Ind. Engineering, 13, 1-4, 15-20.
- Yalabık N., 1990, Us Yapısı ve Yapay Us, Koç Unisys'den Haberler, 3.
- , 1988, Turbo Prolog Reference Guide Version 2.0., Borland International Inc., USA, 435.

- , 1988, Turbo Prolog Users Guide Version 2.0., Borland International Inc., USA, 457.
- , 1986, Turbo Prolog Owners Handbook, Borland International Inc., USA, 435.

E K L E R

Ek.1. Veri Tabanları ve İçerdikleri Değişkenler

1- aylık2(Kodno, AAdet, Gu2, Ay2)

AAdet: Belirtilen kod numarasına sahip mamul için "Ay2" inci ayın "Gu2" inci gününe kadar üretilmesi gereken alçık kalıp adedi

2- gunp1(Kodno, Renk, GunpDepa, GunpSıra, Gunpbfq, GunpDöka, Gunpbdok)

Bu veri tabanındaki değişkenler, belirtilen kod numarası ve renge sahip mamul için günlük üretime ilişkin planlanan değerlerdir.

GunpDepa: Bir günde üretilip depoya girmesi planlanan adet

GunpSıra: Bir günde sırlanması planlanan adet

Gunpbfq: Bir günde fırına girmesi gereken brüt adet

GunpDöka: Bir günde belirtilen kod ve renge sahip ürün için yapılması gereken brüt döküm adedi

3- gunkf1(Kodno, Renk, GunfkDep, GunfkSır, Gunfkbfq, G, Ay1)

Belirtilen kod numarası ve renge sahip mamul için "Ay1" inci ayın "G" inci gününe gelindiğinde kümülatif olarak;

GunfkDep: Üretimi tamamlanıp depoya giren mamul adedi

GunfkSır: Sırlanan mamul adedi

Gunfkbfq: Fırına giren mamul adedi

4- gunkf2(Kodno, GunfkYm, GunfkDök, Gunfkbdok, GunfkAlçı, G, Ay1)

Belirtilen kod numarasına sahip yarı mamul için "Ay1" inci ayın "G" inci gününe gelindiğinde kümülatif olarak;

GunfkYm: Üretilen yarı mamul adedi

GunfkDök: Dökümü yapılan mamul adedi

Gunfkbdok: Yapılan brüt döküm adedi

GunfkAlçı: Üretilen alçık kalıp adedi

Ek.1. (devam)

5- dokumplan(Kodno, Döka, NetDök, Gu2, Ay2)

Belirtilen kod numarası ve renge sahip mamulün beyaz rengi için "Ay2" inci ayın "Gu2" inci gününe kadar;

Döka: Aylık planlanan brüt döküm adedi

NetDök: Aylık planlanan net döküm adedi

6- dokplan(Kodno, GunpBD, GunpND)

Belirtilen kod numarasına sahip mamulün beyaz rengi için;

GunpBD: Günlük toplam olarak dökülmesi gereken brüt döküm adedi

GunpND: Günlük toplam olarak elde edilmesi gereken net döküm adedi

7- eldg(Kodno, Renk, FarkTop)

FarkTop: Belirtilen kod numarası ve renge sahip mamulün üretimi için elde bulunan süre (gün)

8- galçp(Kodno, GunpAlçia, FarkTop)

Belirtilen kod numarasına sahip yarı mamul için;

GunpAlçia: Günlük üretilmesi planlanan alçikalıp adedi

FarkTop: Planlanan toplam alçikalıp üretimini yapmak için elde bulunan süre (gün)

9- bugün(Gu1, Ay1, Gun)

Bulunulan günün tarihini içerir.

Örnek: 5 Şubat Salı
| | |
Gu1 Ay1 Gun

Ek.1. (devam)

10- plan(Gu1,Ay1,Gun)

Planlama döneminin başlangıç tarihini içerir.

11- egalçı(Kodno,FarkTop)

FarkTop: Belirtilen kod numarasına sahip yarı mamulün planlanan adette alçık kalıp üretimini yapabilmek için elde bulunan süre (gün)

12- gunf1(Kodno, Renk, GunfDepa, GunfSıra, Gunfbfg, Gu1, Ay1)

Belirtilen kod numarası ve renge sahip mamul için "Ay1" inci ayın "Gu1" inci gününde;

GunfDepa: Üretilip depoya giren adet

GunfSıra: Sırlanan adet

Gunfbfg: Fırından çıkan adet

13- gunf2(Kodno, YmAdet, GunfDöka, Gunfbdok, AlçıkF, FDökFire, Gu1, Ay1)

Belirtilen kod numarasına sahip yarı mamul için "Ay1" inci ayın "Gu1" inci gününde;

YmAdet: Üretilen yarı mamul adedi

GunfDöka: Elde edilen net döküm adedi

Gunfbdok: Yapılan brut döküm adedi

AlçıkF: Üretilen alçık kalıp adedi

FDökFire: Gerçekleşen döküm fire

Ek.1. (devam)

14- gunkp1 (Kodno, Renk, GunpkDepa, GunpkSıra, Gunpkbfg, Gu1, Ay1)

Belirtilen kod numarası ve renge sahip mamul için "Ay1" inci ayın "Gu1" inci gününe gelindiğinde kumulatif olarak;

GunpkDepa: Üretimi tamamlanıp depoya girmesi planlanan mamul adedi

GunpkSıra: Sırlanması planlanan mamul adedi

Gunpkbfg: Fırına girmesi planlanan mamul adedi

15- gunkpdök (Kono, GunpbkDök, GunpkDök, Gu1, Ay1)

Belirtilen kod numarasına sahip mamulün tüm renklerinin "Ay1" inci ayın "Gu1" inci gününe gelindiğinde toplam olarak;

GunpbkDök: Dökülmesi planlanan adet

GunpkDök: Elde edilmesi planlanan net döküm adedi

16- gunkpalçı (Kodno, GunpkAlçı, Gu1, Ay1)

GunpkAlçı: Belirtilen kod numarasına sahip mamul için "Ay1" inci ayın "Gu1" inci gününe gelindiğinde kumulatif olarak üretilmesi planlanan alçık kalıp adedi

17- sıra (Kodno, Renk, Kod, N)

Belirtilen kod numarası ve renge sahip mamul için;

Kod: Günlük üretim değerlendirmesi sonunda hangi sapma grubuna düştüğünü gösteren kod numarası

N: Üretimin planlara uygun olarak gerçekleşmemesi halinde verilen sıra numarası.

Ek.2. Programa Ait Menüler

ROLOG2>tez

GUNLUK URETİMİ İZLEME PROGRAMI

Bugünün tarihini giriniz.....:
(Örnek:5 Şubat için Gun=5,Ay=2 şeklinde)

Gun.....:2 Ay.....:7

Hangi gündesiniz ?... salı

ROLOG2>tez

Açıklama Penceresi

Bu programı doğru bir şekilde kullanabilmeniz için;
öncelikle aşağıdaki iki adımın sırayla yapılması gerekmektedir:

- Aylık planlanan değerlerin girilmesi,
- Bu değerlerin günlük planlara dönüştürülmesi,

Bu işlemler daha önce yapıldı mı?(e/h) (e)

ÖNEMLİ NOT:Yukarıdaki iki adımın planlama döneminin en başında,
yalnızca bir kez çalıştırılması gerekmektedir.

Ek.2. (devam)

ROLOG2>tez

Açıklama Penceresi

Planlama döneminizin başlangıç tarihini giriniz
(Örn. 5 Şubat salı için Gun1:5 Ay:2 Gun2:salı şeklinde)

Gun1 :2 Ay :7 Gun2 :salı

ROLOG2>tez

Açıklama Penceresi

Lütfen önce sırasıyla aşağıdaki işlemleri yapınız:

Bunun için şimdi gelecek menüden 1 ve 2 seçeneklerini sırayla çalıştırmalısınız.

- Aylık veri girişi,
- Aylık bilgilerden günlük planlanmış değerlerin elde edilmesi.

Bu işlemler bittikten sonra programı yeniden çalıştırınız.

Lütfen herhangi bir tuşa basınız...

Ek.2. (devam)

ROLOG2>tez

Aylık Girişler

Kod numarasını giriniz.....:6280
Renk kodunu giriniz.....:beyaz

Planlama doneminde

Depoya girmesi planlanan adedi giriniz...:5322
Son tarihi giriniz.....:
(Örnek: 5 Şubat için Gun=5, Ay=2 gibi.)

Gu2.....:31 Ay2.....:7

Doğru girdiniz mi? (e/h) (e)
devam E/H ... (e)

ROLOG2>tez

Aylık Girişler

Kod numarasını giriniz.....:6280
Aylık planlanan alçı kalıp üretim adedini giriniz...:900

Son üretim tarihini giriniz...

Gu2.....:31 Ay2.....7

Doğru girdiniz mi? (e/h) (h)

Ek.2. (devam)

ROLOG2>tez2

Açıklama Penceresi

Bu programı doğru bir şekilde kullanabilmeniz için;

- daha önceden 'tez' adlı program çalıştırılmış olmalıdır.
('Tez' programı ile günlük üretimi izlemenin öncesinde yapılması gereken ön işlemler yapılmaktadır.)

tez.exe yi çalıştırdınız mı?(e/h) (e)

ROLOG2>tez2

Açıklama Penceresi

şimdi bu programdan çıkıyoruz.
Çıktıktan sonra tez.exe'yi çalıştırmak için :

yalnızca 'tez' yazmanız yeterlidir

lütfen herhangi bir tuşa basınız...

Ek.2. (devam)

ROLOG2>tez2

SIRALAMA MODULU

Sonucuları nereden almak istersiniz?

Ekran(1)

Printer(2)

Seciminiz()

Ek.3. Geliştirilen Bilgisayar Programı

* Bu modülde aylık planlanan değerler girilip, günlük planlanan değerlere dönüştürülmektedir.*

```
project "tez"  
include "globtez.pro"
```

```
predicates
```

```
AnaGiriş(integer)  
BolumHes(symbol, integer, symbol, symbol, integer, integer, integer, integer, integer, integer, integer, integer, integer, integer)  
BolHesAlçı(symbol, integer, integer, symbol, integer, integer)  
kontrol(symbol, symbol, integer, integer, integer, integer, integer, integer, symbol, integer, integer, integer, integer)  
kontrol2(symbol, integer, symbol, integer, integer, integer, integer, integer)  
devam  
deva  
equal(symbol, symbol)  
consult_galçp  
consult_gunp1  
consult_gunf1  
consult_dokplan  
consult_gunkfalc  
consult_gunf2  
consult_gunkp1  
consult_gunkf1  
consult_gunkf2  
consult_aylık1  
consult_eldg  
consult_plan  
consult_bugün  
consult_egalçı  
Hangıgun(symbol, integer, integer, integer, integer, integer, integer)  
El(integer, integer, integer, integer, symbol, integer)
```

```
clauses
```

```
Hangıgun(Gun, Sır, Fırın, Alçık, Dok, Kay):-  
    Gun="pazartesi", Sır=1, Fırın=1, Alçık=1, Dok=1, Kay=1, !;  
    Gun="salı", Sır=1, Fırın=1, Alçık=1, Dok=1, Kay=1, !;  
    Gun="çarşamba", Sır=1, Fırın=1, Alçık=1, Dok=1, Kay=1, !;  
    Gun="perşembe", Sır=1, Fırın=1, Alçık=1, Dok=1, Kay=1, !;  
    Gun="cuma", Sır=1, Fırın=1, Alçık=1, Dok=1, Kay=1, !;  
    Gun="cumartesi", Sır=1, Fırın=1, Alçık=0, Dok=0, Kay=0, !;  
    Gun="pazar", Sır=0, Fırın=1, Alçık=0, Dok=0, Kay=0.
```

```
equal(X, X).
```

```
Gunsay(1, 31).  
Gunsay(2, 28).  
Gunsay(3, 31).  
Gunsay(4, 30).  
Gunsay(5, 31).  
Gunsay(6, 30).
```

Ek.3. (devam)

Gunsay(7,31).
Gunsay(8,31).
Gunsay(9,30).
Gunsay(10,31).
Gunsay(11,30).
Gunsay(12,31).

```
islem_secim(1):-  
makewindow(1,2,3,"AYLIK PLANLANAN GİRİŞLER",5,5,15,70),nl,nl,nl,  
write("      Açıklamalar.....(0)"),nl,  
write("      Veri Girişi(1).....(1)"),nl,  
write("      Veri Girişi(2).....(2)"),nl,  
write("      Çıkış.....(3)"),nl,nl,nl,nl,  
write("      Seçiminizi Giriniz.....( )"),  
cursor(10,58),readint(Secim),cursor(10,59),write(" "),removewindow,  
AnaGiriş(Secim).
```

```
islem_secim(2):- consult_gunpl,consult_dokplan,  
makewindow(1,2,3,"Planlanan Günlük Değerlerin  
Hesabı",2,2,20,78),nl,nl,nl,  
consult_eldg,consult_plan,consult_aylık1,gunplan_secim(1),  
makewindow(1,2,3,"GÜNLÜK PLANLANAN DEĞERLERİN  
HESABI",2,2,20,78),nl,nl,nl,  
write("şimdi programdan çıkıp tez.exe'yi yeniden çalıştırınız."),pause.
```

```
gunplan_secim(1):-  
makewindow(1,2,3,"GÜNLÜK PLANLANAN DEĞERLERİN  
HESABI",2,2,20,78),nl,nl,nl,  
write("      Aylık planlanan degerler günlük degerlere  
çevriliyor..."),  
plan(k(Gu1,Ay1,Gun)),  
aylık1(k(Kodno,Renk,BrDok,NetDok,NetSır,BrFırGiriş,DAdet,Gu2,Ay2)),  
kontrol(Kodno,Renk,BrDok,NetDok,NetSır,BrFırGiriş,DAdet,Gun,Gu1,Ay1,Gu  
2,Ay2).
```

```
gunplan_secim(11):-!,consult_plan,consult_aylık2,consult_galçp,pause,  
gunplan_secim(22).
```

```
gunplan_secim(22):-plan(k(Gu1,Ay1,Gun)),retract(aylık2(k(Kodno,AAdet,Gu2,  
Ay2))),  
kontrol2(Kodno,AAdet,Gun,Gu1,Ay1,Gu2,Ay2).
```

```
kontrol2(Kodno,AAdet,Gun,Gu1,Ay1,Gu2,Ay2):- Kodno<>"1111",consult_egalç1,  
egunalç1(Gu1,Gu2,Ay2,Ay1,Kodno,FarkTop),  
BolHesAlç1(Gun,0,AAdet,Kodno,FarkTop,0),  
gunplan_secim(22);  
egunalç1(Gu1,Gu2,Ay2,Ay1,Kodno,FarkTop),  
BolHesAlç1(Gun,0,AAdet,Kodno,FarkTop,0).
```

```
egunalç1(Gu1,Gu2,Ay2,Ay1,Kodno,FarkTop):-Ay2-Ay1=0,FarkTop=Gu2-Gu1,  
asserta(egalç1(k(Kodno,FarkTop))),!; Gunsay(Ay1,Say11),  
FarkTop=Say11-Gu1+Gu2,asserta(egalç1(k(Kodno,FarkTop))),  
save("egalç1.dba",egalç1).
```

Ek. 3. (devam)

```
kontrol(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gun, Gu1, Ay1, Gu2, Ay2):-
```

```
    M=Gu1+1,
    Kodno="1111", Elg(M, Gu2, Ay1, Ay2, Kodno, Renk, FarkTop),
    BolumHes(Gun, 0, Kodno, Renk, FarkTop, DAdet, BrDok, NetDok, NetSır, BrFırGiriş, 0, 0, 0),
    retract aylık1(k(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gu2, Ay2)),
    gunplan_secim(11).
```

```
kontrol(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gun, Gu1, Ay1, Gu2, Ay2):-
```

```
    Elg(Gu1, Gu2, Ay1, Ay2, Kodno, Renk, FarkTop), Yenıgun(Gun, Gunc),
    BolumHes(Gunc, 0, Kodno, Renk, FarkTop, DAdet, BrDok, NetDok, NetSır, BrFırGiriş, 0, 0, 0),
    retract aylık1(k(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gu2, Ay2)),
    gunplan_secim(1).
```

```
Elg(Gu1, Gu2, Ay1, Ay2, Kodno, Renk, FarkTop):- Ay2-Ay1=0, FarkTop=Gu2-Gu1,
    assertz(eldg(k(Kodno, Renk, FarkTop))), save("eldg.db", eldg), !;
Gunsay(Ay1, Sayı1), FarkTop=Sayı1-Gu1+Gu2,
    assertz(eldg(k(Kodno, Renk, FarkTop))), save("eldg.db", eldg).
```

```
El(Gu1, Gu2, Ay1, Ay2, Kodno, FarkTop):-Ay2-Ay1=0, FarkTop=Gu2-Gu1;
    Gunsay(Ay1, Sayı1), FarkTop=Sayı1-Gu1+Gu2.
```

```
AnaGiriş(1):-consult_aylık1, consult_gunf1, consult_gunkp1, consult_gunkfalç,
    ,
    consult_gunkf1, consult_bugün, AnaGiriş(11).
```

```
AnaGiriş(11):-
```

```
makewindow(1,2,3,"Aylık Girişler",2,2,20,78),nl,
cursor(5,10),write("Kod numarasını giriniz.....:"),
cursor(6,10),write("Renk kodunu giriniz.....:"),
cursor(8,13),write("Planlama doneminde"),
cursor(10,10),write("Depoya girmesi planlanan adedi giriniz..:"),
cursor(11,10),write("Son tarihi giriniz.....:"),
cursor(12,10),write("Örnek: 5 Şubat için Gun=5, Ay=2 gibi."),
cursor(14,14),write("Gu2.....:"),cursor(14,30),write("Ay2.....:"),
cursor(5,51), readln (Kodno),
cursor(6,51), readln (Renk),
cursor(10,51), readint (DAdet),
cursor(14,24), readint (Gu2),
cursor(14,30),write("Ay2.....:"),
cursor(14,40), readint (Ay2),
Pfireler(Kodno,Dök,Fırın),
BrFırGiriş=DAdet/(1-Fırın),BrDok=BrFırGiriş/(1-Dök),NetSır=BrFırGiriş,
NetDok=BrDok*(1-Dök),cursor(17,45),
write("Doğru girdiniz mi? (e/h) ( )"),cursor(17,71),readln(Cev),
cursor(16,72),write(" ") ,Cev=e,
assertz(aylık1(k(Kodno, Renk, BrDok, NetDok, NetSır, BrFırGiriş, DAdet, Gu2, Ay2)
)),
save("aylık1.db", aylık1), bugün(k(Gu1, Ay1, Gun)),
assertz(gunkf1(k(Kodno, Renk, 1, 1, 1, Gu1, Ay1))),
assertz(gunkp1(k(Kodno, Renk, 1, 1, 1, Gu1, Ay1))),
```

Ek.3. (devam)

```
assertz(gunkfalc(k(Kodno,1,Gu1,Ay1))),
assertz(gunf1(k(Kodno,Renk,1,1,1,Gu1,Ay1))),
save("gunf1.dba",gunf1),
save("gunkf1.dba",gunkf1),save("gunkfalc.dba",gunkfalc),
save("gunkp1.dba",gunkp1),cursor(17,45),devam;AnaGiriş(11).
```

```
AnaGiriş(2):-consult_plan,consult_aylık2,consult_gunkf2,consult_gunf2,
consult_bugUn,AnaGiriş(22).
```

```
AnaGiriş(22):-makewindow(1,2,3,"Aylık Girişler",2,2,20,78),
cursor(8,9),write("Kod numarasını
giriniz.....:"),
cursor(9,9),write("Aylık planlanan alçı kalıp üretim adedini
giriniz...:"),
cursor(8,62),readln(Kodno),
cursor(9,62),readint(AAdet),
cursor(11,9),
write("Son üretim tarihini giriniz..."),nl,cursor(13,22),
write("Gu2.....:"),cursor(13,46),write("Ay2....."),
cursor(13,32),readint(Gu2),
cursor(13,46),write("Ay2....."),
cursor(13,54),readint(Ay2),cursor(15,30),
write("Doğru girdiniz mi? (e/h) ( )"),cursor(15,56),readln(Cev),
cursor(15,57),write(" "),Cev=e,
assertz(aylık2(k(Kodno,AAdet,Gu2,Ay2))),save("aylık2.dba",aylık2),
plan(k(Gu1,Ay1,Gun)),write("plandan değer alındı"),pause,
El(Gu1,Gu2,Ay1,Ay2,Kodno,FarkTop),write("El hesaplandı"),pause,
bugUn(k(Gu1,Ay1,Gun)),assertz(gunkf2(k(Kodno,1,1,1,1,Gu1,Ay1))),
assertz(gunf2(k(Kodno,1,1,1,1,Gu1,Ay1))),
save("gunf2.dba",gunf2),save("gunkf2.dba",gunkf2),cursor(17,30),deva;AnaGiriş(22).
```

```
AnaGiriş(3):-menu.
```

```
AnaGiriş(0):-write("Bu modul, günlük fiili verilerin girişi için
tasarlanmıştır."),
write("Bazı bilgilerde renk detayına gerek olmadığından (süreçte henüz
sırlama safhasına gelinmediğinden) sadece kod bazında veri"),
write("girişi istenmiştir. Ürneğin 6287 kod nolu mamulden o gün kaç adet
döküldüğü gibi."),
write("Bazı bilgiler ise hem kod hem de renk detayındadır."),
write("6287 kod nolu mamulun beyaz renginden o gün üretilip depoya giren
adet gibi"),nl,nl,
write("Buna göre kod ve renk detayındaki bilgilerin girişi için 1
seçeneğini,"),
write("Sadece kod detayındaki bilgilerin girişi için 2 seçeneğini
seçmelisiniz."),
write("Sonuc olarak her iki grup veri girişi de yapılmalıdır."),pause.
```

```
Pfireler(Kodno,Dök,Fırın):-
Kodno="6280",Dök=0.11,Fırın=0.35;
Kodno="6282",Dök=0.15,Fırın=0.41;
Kodno="6287",Dök=0.15,Fırın=0.40;
Kodno="6288",Dök=0.15,Fırın=0.30;
Kodno="6552",Dök=0.12,Fırın=0.21;
Kodno="6553",Dök=0.14,Fırın=0.41;
Kodno="1111",Dök=0.08,Fırın=0.30.
```

Ek.3. (devam)

```
devam:-write("devam E/H ... ( )"),cursor(17,59),
readln(Harf),
cursor(17,60),write(" "),Harf=h,islem_secim(1),!;AnaGiris(11).

deva:-write("devam E/H ... ( )"),cursor(17,44),
readln(Harf), Harf=h,islem_secim(1);AnaGiris(22).

BolumHes(Gun,Sayac,Kodno,Renk,FarkTop,Dadet,BrDok,NetSır,BrFırGiriş,Topa,Topd,Topkay):-
    Sayac1=Sayac+1,Sayac1<=FarkTop,
    Hangıgun(Gun,_,_,Alçık,Dok,Kay),
    AlçTop=Topa+Alçık,DokTop=Topd+Dok,
    KayTop=Topkay+Kay,
    Yenıgun(Gun,Gunc),
    BolumHes(Gunc,Sayac1,Kodno,Renk,FarkTop,Dadet,BrDok,NetSır,
BrFırGiriş,AlçTop,DokTop,KayTop);
    write("Topkay="),write(Topkay),nl,
    write("Topd="),write(Topd),pause,nl,
    GunpDepa=Dadet/Topkay,
    Gunpbfğ=BrFırGiriş/Topkay,
    GunpSıra=NetSır/Topkay,
    Gunpbdok=BrDok/Topd,
    GunpDöka=NetDok/Topd,
    assertz(gunp1(k(Kodno,Renk,GunpDepa,GunpSıra,Gunpbfğ,GunpDöka,Gunpbdok))),
    assertz(dokplan(k(Kodno,GunpDöka,Gunpbdok))),
    save("gunp1.dba",gunp1),save("dokplan.dba",dokplan).

BolHesAlçı(Gun,Sayac,AAdet,Kodno,FarkTop,Topa):-Hangıgun(Gun,_,_,Alçık,_,_),nl,
Alçtop=Topa+Alçık, Yenıgun(Gun,Gunc),
Sayac1=Sayac+1,Sayac1<=FarkTop,
BolHesAlçı(Gunc,Sayac1,AAdet,Kodno,FarkTop,Alçtop);
!,GunAlç1a=AAdet/Topa,assertz(galçp(k(Kodno,GunAlç1a,FarkTop))),
save("galçp.dba",galçp).

consult_gunkf1:-existfile("gunkf1.dba"),!,consult("gunkf1.dba",gunkf1).
consult_gunkf1.

consult_gunkf2:-existfile("gunkf2.dba"),!,consult("gunkf2.dba",gunkf2).
consult_gunkf2.

consult_galçp:-
    existfile("galçp.dba"),!,
    consult("galçp.dba",galçp).
consult_galçp.

consult_aylık1:-existfile("aylık1.dba"),!,consult("aylık1.dba",aylık1).
consult_aylık1.

consult_aylık2:-existfile("aylık2.dba"),!,consult("aylık2.dba",aylık2).
consult_aylık2.

consult_bugün:-existfile("bugün.dba"),!,consult("bugün.dba",bugün).
consult_bugün.

consult_eldg:-existfile("eldg.dba"),!,consult("eldg.dba",eldg).
consult_eldg.
```

Ek.3. (devam)

```
consult_plan:-
    existfile("plan.dba"),!,
    consult("plan.dba",plan).
consult_plan.

consult_gunp1:-existfile("gunp1.dba"),!,consult("gunp1.dba",gunp1).
consult_gunp1.

consult_egalçi:-existfile("egalçi.dba"),!,consult("egalçi.dba",egalçi).
consult_egalçi.

consult_gunf1:-existfile("gunf1.dba"),!,consult("gunf1.dba",gunf1).
consult_gunf1.

consult_gunf2:-existfile("gunf2.dba"),!,consult("gunf2.dba",gunf2).
consult_gunf2.

consult_dokplan:-existfile("dokplan.dba"),!,consult("dokplan.dba",dokplan
).
consult_dokplan.

consult_gunkp1:-
    existfile("gunkp1.dba"),!,
    consult("gunkp1.dba",gunkp1).
consult_gunkp1.

consult_gunkfalc:-existfile("gunkfalc.dba"),!,consult("gunkfalc.dba",gunk
falc).
consult_gunkfalc.
```

Ek.3. (devam)

* Bu modul bazı sabit tanımlamaları içermektedir.*

```
project "tez"  
include "globtez.pro"
```

```
domains
```

```
kayıt=k(symbol,symbol,integer,integer)
```

```
clauses
```

```
Gunsay(1,31).
```

```
Gunsay(2,28).
```

```
Gunsay(3,31).
```

```
Gunsay(4,30).
```

```
Gunsay(5,31).
```

```
Gunsay(6,30).
```

```
Gunsay(7,31).
```

```
Gunsay(8,31).
```

```
Gunsay(9,30).
```

```
Gunsay(10,31).
```

```
Gunsay(11,30).
```

```
Gunsay(12,31).
```

```
Eg(Gu,Gu1,Ay,Ay1,Kodno,Renk,Top):-Ay1-Ay=0,Top=Gu1-Gu,!;  
    Gunsay(Ay,Sayı),Top=Sayı-Gu+Gu1.
```

```
YeniGun(EskGun,Gun):-EskGun="pazartesi",Gun="salı",!;  
    EskGun="salı",Gun="çarşamba",!;  
    EskGun="çarşamba",Gun="perşembe",!;  
    EskGun="perşembe",Gun="cuma",!;  
    EskGun="cuma",Gun="cumartesi",!;  
    EskGun="cumartesi",Gun="pazar",!;  
    EskGun="pazar",Gun="pazartesi".
```


Ek.3. (devam)

* Bu modulde günlük fiili değerlerin girişi yapılmaktadır. *

```
project "tez"
include "globtez.pro"
```

```
predicates
```

```
dev(integer, integer, integer)
```

```
consult_gunkf1
```

```
consult_gunkf2
```

```
consult_gunf1
```

```
consult_gunf2
```

```
consult_bugun
```

```
consult_galc
```

```
consult_gunkfalc
```

```
clauses
```

```
GunlukFiiliVeriGirişi(1,Gu1,Ay1):-consult_gunf1,consult_gunkf1,
    GunlukFiiliVeriGirişi(11,Gu1,Ay1).
```

```
GunlukFiiliVeriGirişi(11,Gu1,Ay1):-
```

```
makewindow(1,2,3,"GUNLUK FİİLİ GİRİŞLER(1)",2,2,20,78),nl,
```

```
cursor(7,13),write("    Kod numarasını giriniz .....:"),
```

```
cursor(8,13),write("    Renk kodunu giriniz .....:"),
```

```
cursor(9,13),write("    Günlük depoya giren adet .....:"),
```

```
cursor(10,13),write("    Günlük sırlanan adet .....:"),
```

```
cursor(11,13),write("    Günlük fırından çıkan adet .....:"),
```

```
cursor(7,59), readln (Kodno),
```

```
cursor(8,59), readln (Renk),
```

```
cursor(9,59), readint (GunfDepa),
```

```
cursor(10,59), readint (GunfSıra),
```

```
cursor(11,59), readint (Gunfbfç),
```

```
cursor(14,26),
```

```
write("Doğru girdiniz mi? e/h ( )"),cursor(14,50),readln(Cev),
```

```
cursor(14,51),write(" "),Cev=e,
```

```
G=Gu1-1,
```

```
assertz(gunf1(k(Kodno, Renk, GunfDepa, GunfSıra, Gunfbfç, Gu1, Ay1))),
```

```
gunkf1(k(Kodno, Renk, GunfkDep, GunfkSır, Gunkbfç, G, Ay1)),
```

```
GunfkDepY=GunfkDep+GunfDepa,
```

```
GunfkSırY=GunfkSır+GunfSıra,
```

```
GunfkFırY=Gunkbfç+Gunfbfç,
```

```
retract(gunkf1(k(Kodno, Renk, _, _, _, G, Ay1))),
```

```
save("gunf1.dba",gunf1),
```

```
assertz(gunkf1(k(Kodno, Renk, GunfkDepY, GunfkSırY, GunfkFırY, Gu1, Ay1))),
```

```
save("gunkf1.dba",gunkf1),dev(Gu1,Ay1,11);GunlukFiiliVeriGirişi(11,Gu1,Ay1).
```

```
GunlukFiiliVeriGirişi(2,Gu1,Ay1):-consult_gunf2,consult_gunkf2,consult_gunfalc,
```

```
consult_galc, GunlukFiiliVeriGirişi(22,Gu1,Ay1).
```

```
GunlukFiiliVeriGirişi(22,Gu1,Ay1):-makewindow(1,2,3,"GUNLUK FİİLİ GİRİŞLER(2)",2,2,20,78),nl,
```

```
cursor(6,9), write("Kod numarasını giriniz.....:"),
```

```
cursor(7,9), write("Brut dökm) adedi giriniz.....:"),
```

```
cursor(7,9), write("Brut dökm) adedi giriniz.....:"),
```

Ek.3. (devam)

```
cursor(8,9), write("Net döküm adedini
giriniz.....:");
cursor(9,9), write("Yarı mamul adedini
giriniz.....:");
cursor(10,9),write("Uretilen alçı kalıp adedini
giriniz.....:");

cursor(6,60),readln(Kodno),
cursor(7,60),readint(Gunfbdok),
cursor(8,60),readint(GunfDöka),
cursor(9,60),readint(YmAdet),
cursor(10,60),readint(AlçıkF),cursor(14,26),
write("Dogru girdiniz mi? e/h (
)"),cursor(14,50),readln(Cev),cursor(14,51),
write(")"),Cev=e,
G=Gu1-1,write("G="),write(G),
FDökFire=1-(GunfDöka/Gunfbdok),
assertz(gunf2(k(Kodno,YmAdet,GunfDöka,Gunfbdok,FDökFire,Gu1,Ay1))),
write("gunf2 ye kayıt eklendi"),pause,
assertz(galc(k(Kodno,AlçıkF,Gu1,Ay1))),
write("galc a kayıt eklendi"),pause,
save("gunf2.dba",gunf2),
save("galc.dba",galc),
gunkf2(k(Kodno,GunfkYm,GunfkDök,Gunfkbdok,GunfkAlçı,G,Ay1)),
write("gunkf2 den kayıt alındı"),pause,
gunkfalc(k(Kodno,GunfkAlçı,G,Ay1)),write("gunkfalc dan kayıt
alındı"),pause,
GunfkAlçıY=GunfkAlçı+AlçıkF,GunfkDökY=GunfkDök+GunfDöka,
GunfkYmY=GunfkYm+YmAdet,GunfkbdokY=Gunfkbdok+Gunfbdok,
retract(gunkf2(k(Kodno,_,_,_,G,Ay1))),
retract(gunkfalc(k(Kodno,GunfkAlçı,G,Ay1))),
assertz(gunkf2(k(Kodno,GunfkYmY,GunfkDökY,GunfkbdokY,GunfkAlçıY,Gu1,Ay1))
),
assertz(gunkfalc(k(Kodno,GunfkAlçıY,Gu1,Ay1))),
save("gunkf2.dba",gunkf2),save("gunkfalc.dba",gunkfalc),
dev(Gu1,Ay1,22);GunlukFiiliVeriGirişi(22,Gu1,Ay1).

GunlukFiiliVeriGirişi(0,_,_):-
makewindow(1,2,3,"AÇIKLAMA PENCERESİ",2,2,20,78),nl,GunlukGirişler.

GunlukFiiliVeriGirişi(4,_,_):-menu.

GunlukGirişler:-makewindow(1,2,3,"GUNLUK VERİ GİRİŞİ",2,2,20,78),nl,
makewindow(1,2,3,"GUNLUK VERİ GİRİŞİ",2,2,20,78),nl,nl,nl,nl,
write("
Açıklamalar.....(0)"),nl,
write("          Birinci Grup Verilerin
Girişi.....(1)"),nl,
write("          İkinci Grup Verilerin
Girişi.....(2)"),nl,
write("
Çıkış.....(3)"),nl,nl,nl,nl,
write("          Seçiminizi Giriniz.....( )"),nl,nl,
cursor(12,49),
```

Ek.3. (devam)

```
readint (Secim),
clearwindow,consult_bugUn,bugUn(k(Gu1,Ay1,_)),removewindow,
GunlukFiiliVeriGirişi(Secim,Gu1,Ay1),menu.

pause:-nl,write("lutfen herhangi bir tuşa basınız..."), readchar (_,nl).
dev(Gu1,Ay1,Sayı):-cursor(15,26),write("devam e/h ...( )"),cursor(15,40),
readln(Harf),
cursor(15,41),write(""),Harf=h,removewindow,GunlukGirişler,!;
GunlukFiiliVeriGirişi(Sayı,Gu1,Ay1).

consult_gunf1:-existfile("gunf1.db"),!,consult("gunf1.db",gunf1).
consult_gunf1.

consult_gunf2:-existfile("gunf2.db"),!,consult("gunf2.db",gunf2).
consult_gunf2.

consult_gunkf1:-existfile("gunkf1.db"),!,consult("gunkf1.db",gunkf1).
consult_gunkf1.

consult_gunkf2:-existfile("gunkf2.db"),!,consult("gunkf2.db",gunkf2).
consult_gunkf2.

consult_bugUn:-existfile("bugUn.db"),!,consult("bugUn.db",bugUn).
consult_bugUn.

consult_galc:-existfile("galc.db"),!,consult("galc.db",galc).
consult_galc.

consult_gunkfalc:-existfile("gunkfalc.db"),!,consult("gunkfalc.db",gun
kfalc).
consult_gunkfalc.
```

Ek.3. (devam)

* Bu modülde günlük üretimin değerlendirilmesine başlanır. İlk çalıştırıldığında ekrana daha önceden yapılması gereken işlemlerin yapılıp yapılmadığını sorgulamak amacıyla bir mesaj gelir. B u işlemler yapıldıysa program çalışır. Yapılmadıysa programdan çıkarılır. *

```
project "tez2"  
include "globtez.pro"
```

domains

```
gunkpkayıtı=k(symbol, symbol, integer, integer, integer, integer, integer)
```

```
a2=k(symbol, integer, integer, integer, integer)
```

```
kayıt=k(integer, integer, integer, integer, integer)
```

predicates

Hesaplar

```
Devam(real, real, integer, integer)
```

```
git(symbol, symbol, integer, integer, integer, integer, integer, integer,  
)
```

```
SırayaKoy(integer)
```

```
uyarı(integer, symbol, symbol)
```

basla

```
Gi(real, integer)
```

```
consult_gunp1
```

```
consult_eldg
```

```
consult_bugün
```

```
consult_gunf1
```

```
consult_gunf2
```

```
consult_gunkf1
```

```
consult_gunkf2
```

```
consult_gunkp1
```

```
consult_galçp
```

```
consult_plan
```

```
gün(symbol, symbol)
```

```
consult_sıra
```

```
consult_gunkpdök
```

```
consult_gunkpalçı
```

```
consult_galc
```

Analşlemler

Acıkla

```
uyarı22(integer, integer, integer, symbol, symbol, integer, integer)
```

```
Go(integer)
```

```
Başla(symbol)
```

```
kacıncı_ısgunu(kayıt)
```

```
sonişlem(symbol, symbol, integer, integer)
```

```
E(integer, integer, integer, integer, symbol, symbol, integer)
```

```
ayarla(integer)
```

```
tekrarayarla(integer)
```

paus

```
BrutDok(integer, integer)
```

```
ND(integer, integer)
```

```
Dk(real, real)
```

```
Alkalıp(integer, integer)
```

Ek.3. (devam)

clauses

```
kacinci_1sgunu(k(1,7,1,1,1)).
kacinci_1sgunu(k(2,7,2,2,2)).
kacinci_1sgunu(k(3,7,3,3,3)).
kacinci_1sgunu(k(4,7,4,4,4)).
kacinci_1sgunu(k(5,7,5,5,5)).
kacinci_1sgunu(k(6,7,5,5,5)).
kacinci_1sgunu(k(7,7,5,5,5)).
kacinci_1sgunu(k(8,7,6,6,6)).
kacinci_1sgunu(k(9,7,7,7,7)).
kacinci_1sgunu(k(10,7,8,8,8)).
kacinci_1sgunu(k(11,7,9,9,9)).
kacinci_1sgunu(k(12,7,10,10,10)).
kacinci_1sgunu(k(13,7,10,10,10)).
kacinci_1sgunu(k(14,7,10,10,10)).
kacinci_1sgunu(k(15,7,11,11,11)).
kacinci_1sgunu(k(16,7,12,12,12)).
kacinci_1sgunu(k(17,7,13,13,13)).
kacinci_1sgunu(k(18,7,14,14,14)).
kacinci_1sgunu(k(19,7,15,15,15)).
kacinci_1sgunu(k(20,7,15,15,15)).
kacinci_1sgunu(k(21,7,15,15,15)).
kacinci_1sgunu(k(22,7,16,16,16)).
kacinci_1sgunu(k(23,7,17,17,17)).
kacinci_1sgunu(k(24,7,18,18,18)).
kacinci_1sgunu(k(25,7,19,19,19)).
kacinci_1sgunu(k(26,7,20,20,20)).
kacinci_1sgunu(k(27,7,20,20,20)).
kacinci_1sgunu(k(28,7,20,20,20)).
kacinci_1sgunu(k(29,7,21,21,21)).
kacinci_1sgunu(k(30,7,22,22,22)).
kacinci_1sgunu(k(31,7,22,22,22)).
```

Pfireler(Kodno,Dök,Fırın):-

```
Kodno="6280",Dök=0.11,Fırın=0.35;
Kodno="6282",Dök=0.15,Fırın=0.41;
Kodno="6287",Dök=0.15,Fırın=0.40;
Kodno="6288",Dök=0.15,Fırın=0.30;
Kodno="6552",Dök=0.12,Fırın=0.21;
Kodno="6553",Dök=0.14,Fırın=0.41;
Kodno="1111",Dök=0.08,Fırın=0.30.
```

```
E(Gu,Gui,Ay,Ay1,Kodno,Renk,Top):-Ay1-Ay=0,Top=Gui-Gu,!;
    Gunsay(Ay,Sayı),Top=Sayı-Gu+Gui.
```

basla:-

```
makewindow(1,2,3,"ANA MENU",5,5,19,70),nl,nl,nl,
write("        Ana İşlemler.....(1)"),nl,
write("        Açıklamalar.....(2)"),nl,
write("        Çıkış.....(3)"),nl,nl,
write("        Seçiminizi Giriniz.....(
)"),nl,nl,cursor(7,49),
readint (Secim),removewindow,Go(Secim).
```

Ek.3. (devam)

```
Go(Secim):-Secim=1,Analşlemler;
           Secim=2,Acıkla;
           Secim=3,exit.
```

```
Acıkla:-!,makewindow(1,2,3,"SIRALAMA MODULU",2,2,20,78),nl,nl,nl,
write("      Bu modul, gUnluk Üretimi deęerlendirmek için
hazırlanmıştır. Asıl "),nl,
write(" işlemler bu program tarafından yapılacaktır. Eđer daha önce tez
adlı"),nl,
write(" exe programını çalıştırmış ve gerekli ön hazırlıkları
yapmışsanız bu "),nl,
write(" menüden 1 seçeneğini seçerek programı çalıştırmaya
başlayabilirsiniz."),paus,
basla.
```

```
ayarla(2):-writedevice(printer).
ayarla(1).
```

```
tekrarayarla(2):-writedevice(screen).
tekrarayarla(1).
```

```
Analşlemler:-!,makewindow(1,2,3,"SIRALAMA MODULU",2,2,20,78),nl,nl,nl,
write("      Sonuçları nereden almak istersiniz?"),nl,nl,nl,
write("      Ekran ..... (1)"),nl,
write("      Printer ..... (2)"),nl,nl,
write("      Seciminiz ..... ( )"),cursor(9,43),
readint(Secim), cursor(9,44),write(""),ayarla(Secim),
```

```
consult_gunkf1,consult_gunkp1,consult_gunp1,consult_gunf1,consult_sıra,
SırayaKoy(0),tekrarayarla(Secim).
```

```
SırayaKoy(B):-consult_bugün,bugün(k(Gu1,Ay1,_)),
retract(gunf1(k(Kodno,Renk,GunfDepa,_,_ ,Gu1,Ay1))),
retract(gunp1(k(Kodno,Renk,GunpDepa,_,_ ,_))),
retract(gunkp1(k(Kodno,Renk,GunpkDepa,_,_ ,Gu1,Ay1))),
retract(gunkf1(k(Kodno,Renk,GunfkDepa,_,_ ,Gu1,Ay1))),
git(Kodno,Renk,B,GunfkDepa,GunpkDepa,GunfDepa,GunpDepa,Gu1,Ay1).
```

```
git(Kodno,Renk,B,GunfkDepa,GunpkDepa,GunfDepa,GunpDepa,Gu1,Ay1):-
Kodno="1111",
assertz(sıra(k(Kodno,Renk,3,10))),save("sıra.dba",sıra),
consult_sıra,
Hesaplar.
```

```
git(Kodno,Renk,B,GunfkDepa,GunpkDepa,GunfDepa,GunpDepa,Gu1,Ay1):-
SO=1-(GunfkDepa/GunpkDepa),SG=1-(GunfDepa/GunpDepa),
K=GunpDepa-GunfDepa, A=1-((GunfkDepa+K)/GunpkDepa),Gi(A,AKod),
Devam(SO,SG,AKod,Kod),Kod<>0,SıraNo(SO,SıraN1),T=B+1,
assertz(sıra(k(Kodno,Renk,Kod,SıraN1))),write("sıraya deęer kondu"),
save("sıra.dba",sıra),SırayaKoy(T).
```

```
git(Kodno,Renk,B,GunfkDepa,GunpkDepa,GunfDepa,GunpDepa,Gu1,Ay1):-
SırayaKoy(B).
```

Ek.3. (devam)

```
Gi(A,Akod):-A<0.11,AKod=1;
           AKod=0.
```

```
Devam(S0,SG,AKod,Kod):-S0>=0,SG<=0,Kod=1;
                        S0<=0,SG>=0,Kod=2;
                        S0>=0,SG>=0,AKod=1,Kod=3;
                        S0>=0,SG>=0,AKod=0,Kod=4;
                        Kod=0.
```

```
Hesaplar:-!,sira(k(Kodno1,Renk1,Kod,N)),
           nl,nl,nl,nl,
           write("      Kodno      "),write("      Renk      "),nl,
           write("      -----      "),write("      -----      "),nl,
           write("      "),write(Kodno1),write("
"),write(Renk1),nl,nl,paus,
           retract(sira(k(Kodno1,Renk1,Kod,N))),
           save("sira.dba",sira),
           sonislem(Kodno1,Renk1,Kod,N).
```

```
sonislem(Kodno1,Renk1,Kod,N):-Kodno1="1111",
                               write("butun islemler bitti"), paus,exit.
```

```
sonislem(Kodno1,Renk1,Kod,N):-uyari(Kod,Kodno1,Renk1), paus,Hesaplar.
```

```
BrutDok(Gunpbdok,Gunfbdok):-Gunpbdok>Gunfbdok,
                             write(" BugUn "),write(Gunfbdok),write(" adet brüt döküm
yapılmıştır. Oysa bu deger "),
                             write(Gunpbdok),write(" olarak"),nl,
                             write(" planlanmıştı."),nl;true.
```

```
ND(GunpDöka,GunfDöka):-GunpDöka>GunfDöka,
                        write(" Elde edilen net döküm adedi planlanandan azdır."),nl,
                        write(" Günlük net döküm adedi "),write(GunpDöka),write("
olmalıydı. Oysa bugünkü net döküm"),nl,
                        write(" adedi "),write(GunfDöka),write(" olarak gerçekleşmiştir.
Bu durum yarın bu kod için "),nl,
                        write(" elde edilecek mamul adedinin azalmasına yol
açabilir."),nl
                        ;true.
```

```
Dk(FD,Dök):-FD>Dök,
             write(" Dokum fire planlanani asmıştır."),nl,
             write(" Planlanan döküm fire degeri "),write(Dök),write(" olduğu
halde bu-"),nl,
             write(" gün için bu deger "),write(FD),write("olarak
gerçekleşmiştir.");true.
```

```
Alkalip(GunpAlçia,AlçıkF):-GunpAlçia>AlçıkF,
                           write(" Bugunku alcıkalıp uretimi planın gerisinde
kalmıştır."),nl,
                           write(" Planlanan alçıkalıp degeri "),write(GunpAlçia),write("
olarak be-"),nl,
```

Ek.3. (devam)

```
write(" lirlenmişti. Fakat bugün "),write(AlçıkF),write(" adet
alçık kalıp"),nl,
write(" Üretilmiştir.");true.
```

```
uyarı22(22,Gu1,Ay1,Kodno,Renk,GunfDepa,Gunfbfg):-
makewindow(1,2,3,"GUNLUK ÜRETİM DEĞERLENDİRME
SONUÇLARI",0,0,25,80),nl,
write(Kodno),write(" "),write(Renk),nl,
write("-----"),write(" "),write("-----"),nl,nl,
write(" Bugunku depo girisi planlara uygun olarak
gerçekleşmemiştir. Fa-"),nl,
write(" kat kumulatif planlardan sapma söz konusu değildir. Bu durumda
gunluk"),nl,
write(" üretimdeki sapmanın kumulatif planları etkileyecek boyutta
olmadığı"),nl,
write(" soylenebilir"),nl,nl,gun(Kodno,Renk),
consult_gunp1,consult_gunf2,consult_galc,
consult_galçp,consult_gunf1,gunf1(k(Kodno,Renk,GunfDepa,_,Gunfbfç,Gu1
,Ay1)),
retract(gunp1(k(Kodno,Renk,GunpDepa,_,_,GunpDöka,Gunpbdok))),
G=Gu1-1,galc(k(Kodno,AlçıkF,Gu1,Ay1)),
gunf2(k(Kodno,_,GunfDöka,Gunfbdok,FDökFire,G,Ay1)),
gunf2(k(Kodno,_,GunfDök,Gunfdok,FDökFir,Gu1,Ay1)),
retract(galçp(k(Kodno,GunpAlç1a,_))),
Pfireler(Kodno,Dök,Fırın),
FFırınFire=1-(GunfDepa/Gunfbfç),
FırFireSapma=1-(Fırın/FFırınFire),
BrDökS=1-(Gunfbdok/Gunpbdok),
NDökS=1-(GunfDöka/GunpDöka), Alç1S=1-(AlçıkF/GunpAlç1a),
DepS=1-(GunfDepa/GunpDepa), DökFireSapma=1-(Dök/FDökFire),
islem2(2,Kodno,Renk,Fırın,FFırınFire,Dök,FDökFire,GunfDepa,GunpDepa,G
unfDöka,GunpDöka,AlçıkF,GunpAlç1a,Gunfbdok,Gunpbdok,BrDökS,NDökS,Alç1S,Dö
kFireSapma,FırFireSapma),nl,
```

```
write(" Bugun dokumu yapıлып, yarın tamamlanacak olan mamuller
için bugün-"),
write(" ku üretim değerlendirmesi ise şöyledir:"),nl,nl,
gunp1(k(Kodno,Renk,GunpDepa,_,_,GunpDöka,Gunpbdok)),
BrutDök(Gunpbdok,Gunfdok),nl,ND(GunpDöka,GunfDök),nl,
Dk(FDökFir,Dök),nl,Alkalıp(GunpAlç1a,AlçıkF).
```

```
uyarı(1,Kodno,Renk):-
makewindow(1,2,3,"GUNLUK ÜRETİM DEĞERLENDİRME
SONUÇLARI",0,0,25,80),nl,
write(Kodno),write(" "),write(Renk),nl,
write("-----"),write(" "),write("-----"),nl,nl,
write(" Bu kod için bugunku depo girisi planlara uygun olarak
gerçekleşti-"),nl,
write(" mulatif planlarda sapma vardır. Bu durum önceki günlerde
meydana ge-"),nl,
write(" len kumulatif sapmanın hala devam ettiğini
göstermektedir."),nl,paus,
consult_bugun,bugun(k(Gu1,Ay1,_)),consult_gunf2,consult_galc,
consult_gunkf1,consult_gunkp1,consult_gunkf2,
```


Ek.3. (devam)

```
consult_gunkpdök,consult_gunkpalçı,consult_gunp1,
gunkf1(k(Kodno, Renk, GunfkDep,_, GfkBrFırGir, Gu1, Ay1)),
gunkp1(k(Kodno, Renk, GunpkDepa,_, Gunpkbfg, Gu1, Ay1)),
gunkf2(k(Kodno,_, GunfkDök, Gunfkbdok, GunfkAlçı, Gu1, Ay1)),
gunkpdök(k(Kodno, GunpbkDök, GunpkDök, Gu1, Ay1)),write("gunkpdök
yapıldı"),paus,
gunkpalçı(k(Kodno, GunpkAlçı, Gu1, Ay1)),
DepKS=1-(GunfkDep/GunpkDepa), BrDokKS=1-(Gunfkbdok/GunpbkDök),
NDokKS=1-(GunfkDök/GunpkDök), AlçıKS=1-(GunfkAlçı/GunpkAlçı),
KDökFireFiili=1-(GunfkDök/Gunfkbdok),
FırınCıkSonFirF=1-(GfkBrFırGir/GunfkDep),
FırınCıkSonFirP=1-(Gunpkbfg/GunpkDepa),
FırınCıkSonFire=FırınCıkSonFirF-FırınCıkSonFirP,
```

```
islem1(1, Kodno, Renk, BrDokKS, NDokKS, AlçıKS, KDökFireFiili, FırınCıkSonFire),
write("Kodno="),write(Kodno),nl,nl,nl,
write(" Bugün dokumu yapıp, yarın tamamlanacak olan mamuller
icin bugün-"),nl,
write(" ku üretim degerlendirmesi ise soyledir:"),nl,nl,
gunp1(k(Kodno, Renk, GunpDepa,_,_, GunpDöka, Gunpbdok)),
gunf2(k(Kodno,_, GunfDök, Gunfdok, FDökFire, Gu1, Ay1)),
```

```
galc(k(Kodno, AlçıkF, Gu1, Ay1)),consult_galçp,galçp(k(Kodno, GunpAlçia,_,_)),
FD=1-(GunfDök/Gunfdok),PFireler(Kodno, Dök,_,_),
BrutDok(Gunpbdok, Gunfdok),nl,ND(GunpDöka, GunfDök),nl,
Dk(FD, Dök),nl,Alkalıp(GunpAlçia, AlçıkF),paus.
```

```
uyarı(2, Kodno, Renk):-consult_bugün,bugün(k(Gu1, Ay1, _)),
consult_gunf1,
retract(gunf1(k(Kodno, Renk, GunfDepa,_,_, Gunfbfg, Gu1, Ay1))),
uyarı22(22, Gu1, Ay1, Kodno, Renk, GunfDepa, Gunfbfg),!.
```

```
uyarı(3, Kodno, Renk):-
makewindow(1,2,3, "GUNLUK URETİM DEGERLENDİRME
SONUÇLARI",0,0,25,80),nl,
write(" GENEL DEGERLENDİRME
"),nl,nl,nl,
write("Kodno"),write(" "),write("Renk"),nl,
write("-----"),write(" "),write("-----"),nl,
write(Kodno),write(" "),write(Renk),nl,nl,
consult_gunf1,consult_gunp1,consult_gunf2,
consult_galçp,consult_galc,consult_bugün,
uyarı(33, Kodno, Renk).
```

```
uyarı(33, Kodno, Renk):-
retract(gunp1(k(Kodno, Renk, GunpDepa,_,_, GunpDöka, Gunpbdok))),
bugün(k(Gu1, Ay1, Gun)),
G=Gu1-1,gunf1(k(Kodno, Renk, GunfDepa,_,_, Gunfbfg, Gu1, Ay1)),
write(" Bu kod için bugün depoya girişi yapılan mamul adedi,
planlananın"),nl,
write(" gerisinde kalmıştır. Ustelik bugüne gelindiğinde kümülatif
olarak da"),nl,
write(" üretimi tamamlanıp depoya girmiş olması gereken mamul adedi
plana göre-"),nl,
```

Ek.3. (devam)

```
write(" re daha azdır. Fakat kumulatiflerdeki sapmanın yalnızca bugün
gunluk"),nl,
write(" üretimdeki sapmadan kaynaklandığı anlaşılmıştır. Buna göre
bugün gün-"),nl,
write(" luk üretimde kumulatifleri etkileyecek oranda bir sapmanın
oluştugu"),nl,
write(" söylenebilir."),nl,nl,gün(Kodno,Renk),
galc(k(Kodno,AlçıkF,Gu1,Ay1)),
retract(gunf2(k(Kodno,_,GunfDöka,Gunfbdok,FDökFire,G,Ay1))),
retract(galçp(k(Kodno,GunpAlçia,_))),
Pfireler(Kodno,Dök,Fırın),FFırınFire=1-(GunfDepa/Gunfbfç),
FırFireSapma=1-(Fırın/FFırınFire),
BrDökS=1-(Gunfbdok/Gunpbdok),
NDökS=1-(GunfDöka/GunpDöka), Alçıs=1-(AlçıkF/GunpAlçia),
DepS=1-(GunfDepa/GunpDepa), DökFireSapma=1-(Dök/FDökFire),
makewindow(1,2,3,"GUNLUK ÜRETİM DEĞERLENDİRME
SONUÇLARI",0,0,25,80),nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,
write("      Bugün "),write(Kodno),write(" nolu mamulun
"),write(Renk),
write(" renginden üretimi tamamlanarak depoya "),nl,
write("      girmesi gereken adet "),write(GunpDepa),write("
olmalıydı. Fakat bugün bu değer "),
write(GunfDepa),write(" olarak "),nl,write("
gerçekleşmiştir."),
nl,nl,nl,nl,nl,nl,paus,
removewindow,removewindow,
islem3(3,Kodno,Renk,Fırın,FFırınFire,Dök,FDökFire,GunfDepa,GunpDepa,G
unfDöka,GunpDöka,AlçıkF,GunpAlçia,Gunfbdok,Gunpbdok,BrDökS,NDökS,Alçıs,Dö
kFireSapma,FırFireSapma),paus,
removewindow,
makewindow(1,2,3,"GUNLUK ÜRETİM DEĞERLENDİRME
SONUÇLARI",0,0,25,80),nl,nl,
write("Kodno"),write(" "),write("Renk"),nl,
write("-----"),write(" "),write("-----"),nl,
write(Kodno),write(" "),write(Renk),nl,nl,
write("      Bugün dökümü yapılıp, yarın tamamlanacak olan mamuller
için bugün-"),nl,
write("      ku üretim değerlendirmesi ise şöyledir:"),nl,nl,
retract(gunf2(k(Kodno,_,GunfDök,Gunfdok,FDökFir,Gu1,Ay1))),nl,
galc(k(Kodno,AlçıkF,Gu1,Ay1)),nl,
BrutDök(Gunpbdok,Gunfdok),nl,ND(GunpDöka,GunfDök),nl,
Dk(FDökFir,Dök),nl,Alkalıp(GunpAlçia,AlçıkF).

uyarı(4,Kodno,Renk):-
makewindow(1,2,3,"GUNLUK ÜRETİM DEĞERLENDİRME
SONUÇLARI",0,0,25,80),nl,
write("Kodno"),write(" "),write("Renk"),nl,
write("-----"),write(" "),write("-----"),nl,nl,
write(Kodno),write(" "),write(Renk),nl,nl,
write("      Bugünkü üretim sonucu hem günlük hem de kumulatif
planlardan sapma"),nl,
write(" görülmüştür. Bugünkü sapma kumulatiflerdeki sapmayı
açıklamaya yetmemek-"),nl,
write(" tedir. Bu durum, Üretimde önceki günlerde meydana gelen
açığın bugünün"),nl,
```

Ek.3. (devam)

```
write(" sonuna gelindiğinde hala kapatılmadığını, ayrıca bugünde
limitleri a-"),nl,
write(" şacak düzeyde sapma olduğunu göstermektedir."),nl,
write(" Bu durumun nedenlerini araştırmak için hem bugün Uretimi
tamamlanan"),nl,
write(" mamullerin Uretim süreci hem de kümülatif değerler
incelenmelidir."),paus,
consult_bugün,bugün(k(Gu1,Ay1,_)),
```

```
consult_gunkf1,consult_gunkp1,consult_gunf1,consult_gunp1,consult_gunf2,
consult_galçp,consult_galc,consult_gunkf2,consult_gunkpdök,
consult_gunkpalçı,
gunkf1(k(Kodno, Renk, GunfkDep, _, GfkBrFırGir, Gu1, Ay1)),
gunkf2(k(Kodno, _, GunfkDök, Gunfkbdok, GunfkAlçı, Gu1, Ay1)),
gunkp1(k(Kodno, Renk, GunpkDepa, _, Gunpkbfg, Gu1, Ay1)),
gunf1(k(Kodno, Renk, GunfDepa, _, Gunfbfç, Gu1, Ay1)),
gunp1(k(Kodno, Renk, GunpDepa, _, _, Gunpdöka, Gunpbdok)),
G=Gu1-1,
gunf2(k(Kodno, _, Gunfdöka, Gunfbdok, FDökFire, G, Ay1)),
gunf2(k(Kodno, _, GunfD, Gunfb, FDF, Gu1, Ay1)),
galc(k(Kodno, AlçıkF, Gu1, Ay1)),
galçp(k(Kodno, Gunpalçıa, _)),
gunkpdök(k(Kodno, GunpbkDök, GunpkDök, Gu1, Ay1)),
gunkpalçı(k(Kodno, GunpkAlçı, Gu1, Ay1)),
Pfireler(Kodno, Dök, Fırın),
DepKS=1-(GunfkDep/GunpkDepa), BrDokKS=1-(Gunfkbdok/GunpbkDök),
NDokKS=1-(GunfkDök/GunpkDök),
AlçıKS=1-(GunfkAlçı/GunpkAlçı), DepS=1-(GunfDepa/GunpDepa),
write("Buraya gerekli dokum bilgileri eklenecek"),nl,
NDokS=1-(Gunfdöka/Gunpdöka), AlçıS=1-(AlçıkF/Gunpalçıa),
BrDokS=1-(Gunfbdok/Gunpbdok),
FFırınFire=1-(GunfDepa/Gunfbfç), FFırınFir=FFırınFire+0.01,
FırınFireSapma=1-(Fırın/FFırınFir),
DökFireSapma=1-(Dök/FDökFire),
KDökFireFiili=1-(GunfkDök/Gunfkbdok),
FırınCıkSonFirF=1-(GfkBrFırGir/GunfkDep),
FırınCıkSonFirP=1-(Gunpkbfg/GunpkDepa),
FırınCıkSonFire=FırınCıkSonFirF-FırınCıkSonFirP,
BrDS=1-(Gunfb/Gunpbdok),
NDS=1-(GunfD/Gunpdöka),
islem4(4,Kodno, Renk, Gunpdöka, Gunpbdok, Gunfdöka, Gunfbdok, Dök, FDökFire,
Fırın, FFırınFire, BrDS, NDS, FDF, DepKS, BrDokKS, BrDokS, AlçıKS, DepS, NDokKS, NDo
kS, AlçıS, FırınFireSapma, DökFireSapma, KDökFireFiili, FırınCıkSonFire).
```

```
gün(Kodno, Renk):-
write("
BUGÜN İÇİN EKSİK DEPO GİRİŞİNİN NEDENLERİ
"),nl,nl,
write(" Bugün bitmiş olarak depoya giren mamullerin üretimine dün
başlandı-"),nl,
write(" gından, sapmaların nedenlerini de dünkü fiili üretim
değerlerini ince-"),nl,
write(" leyerek araştırmalıyız. Bu inceleme için
"),write(Kodno),write(" nolu mamulün dün"),nl,
write(" kaç adet döküldüğüne, bunlardan kaç tane net döküm elde
edildine ve "),nl,
```

Ek.3. (devam)

```
write(" diđer ařamalara bakmalıyız. "),nl,  
write(" Bu deđerlendirmelerin sonuęları ařaęıdaki  
gibidir:"),nl,paus.
```

```
SıraNo(S0,SıraN):-  
    S0>0.41, SıraN=1;  
    S0>0.31, S0<0.40,SıraN=2;  
    S0>0.21, S0<0.30,SıraN=3;  
    S0>0.11, S0<0.20,SıraN=4;  
    SıraN=5.
```

```
paus:-nl,write("lutfen herhangi bir tuřa basınız..."), readchar (_,)nl.  
consult_gunf1:-existfile("gunf1.dba"),!,consult("gunf1.dba",gunf1).  
consult_gunf1.
```

```
consult_gunkp1:-  
    existfile("gunkp1.dba"),!,  
    consult("gunkp1.dba",gunkp1).  
consult_gunkp1.
```

```
consult_gunkpalęı:-  
    existfile("gunkpalęı.dba"),!,  
    consult("gunkpalęı.dba",gunkpalęı).  
consult_gunkpalęı.
```

```
consult_gunkpdök:-  
    existfile("gunkpdök.dba"),!,  
    consult("gunkpdök.dba",gunkpdök).  
consult_gunkpdök.
```

```
consult_gunp1:-existfile("gunp1.dba"),!,consult("gunp1.dba",gunp1).  
consult_gunp1.
```

```
consult_eldg:-existfile("eldg.dba"),!,consult("eldg.dba",eldg).  
consult_eldg.
```

```
consult_sıra:- existfile("sıra.dba"),!,consult("sıra.dba",sıra).  
consult_sıra.
```

```
consult_gunkf1:-existfile("gunkf1.dba"),!,consult("gunkf1.dba",gunkf1).  
consult_gunkf1.
```

```
consult_gunkf2:-existfile("gunkf2.dba"),!,consult("gunkf2.dba",gunkf2).  
consult_gunkf2.
```

```
consult_gunf2:-existfile("gunf2.dba"),!,consult("gunf2.dba",gunf2).  
consult_gunf2.
```

```
consult_bugün:-existfile("bugün.dba"),!,consult("bugün.dba",bugün).  
consult_bugün.
```

```
consult_plan:-  
    existfile("plan.dba"),!,  
    consult("plan.dba",plan).  
consult_plan.
```

```
consult_galęp:- existfile("galęp.dba"),!,consult("galęp.dba",galęp).  
consult_galęp.
```

```
consult_galc:- existfile("galc.dba"),!,consult("galc.dba",galc).  
consult_galc.
```

Ek.3. (devam)

Başla(e):- basla.

```
Başla(h):- makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(7,10),
write("şimdi bu programdan çıkıyoruz."),cursor(8,10),
write("Çıktıktan sonra tez.exe'yi çalıştırmak için :"),cursor(10,15),
write("yalnızca 'tez' yazmanız yeterlidir"),cursor(17,30),
write("lütfen herhangi bir tuşa basınız..."), readchar (_),
removewindow, exit.
```

goal

```
makewindow(1,2,3,"GUNLUK URETİMİ İZLEME PROGRAMI",2,2,20,78),

cursor(7,10),write("BugünUn tarihini giriniz.....:"),
cursor(8,14),write("(Örnek:5 Şubat için Gun=5,Ay=2 şeklinde)"),
cursor(11,14),write("Gun.....:"),cursor(11,33),write("Ay.....:
"),
cursor(13,20),write("Hangi gundesiniz ?..."),
cursor(11,28),readint(Gu1),cursor(11,33),write("Ay.....:"),
cursor(11,46),readint(Ay1),cursor(13,42),readln(Gun),
consult_bugUn,
asserta(bugUn(k(Gu1,Ay1,Gun))),save("bugUn.dba",bugUn),
makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(6,10), write("Bu programı doğru bir şekilde kullanabilmeniz
için;"),
cursor(8,10), write("- daha önceden 'tez' adlı program çalıştırılmış
olmalıdır." ),
cursor(9,10),write("('Tez' programı ile günlük üretimi izlemenin
öncesinde yapılması"),
cursor(10,10),write("gereken ön işlemler yapılmaktadır.)"),
cursor(13,14), write("tez.exe yi çalıştırdınız mı?(e/h) ( )"),
cursor(13,49), readln(Cev), removewindow,Başla(Cev).
```

Ek.3. (devam)

* Bu modülde bir gün içerisinde karşılaşılabilecek farklı durumlar tanımlanmış ve karşı gelen uyarılar yazılmıştır. *

```
project "tez2"  
include "globtez.pro"
```

```
predicates
```

```
consult_gunkf1  
consult_gunkf2  
consult_gunf1  
consult_gunf2  
consult_bugun  
consult_gunkp1  
consult_gunp1
```

```
BrutD(real, integer, integer)
```

```
NetD(real, integer, integer)
```

```
DoFire(real, real, real)
```

```
FFır(real, real, real)
```

```
%BrutDokum(real)
```

```
%KumDökFire(real)
```

```
Alçık(real)
```

```
FırınCık(real)
```

```
%DepoK(real)
```

```
BrutDokK(real)
```

```
NetDokK(real)
```

```
KumDokF(real)
```

```
Brud(real)
```

```
ND(real)
```

```
FireBug(real)
```

```
paus
```

```
clauses
```

```
paus:-nl,write("lütfen herhangi bir tuşa basınız..."), readchar (_,nl).
```

```
islem2(2,Kodno,Renk,Fırın,FFırınFire,Dök,FDökFire,GunfDepa,GunpDepa,GunfDöka,GunpDöka,AlçıkF,GunpAlçık,Gunfbdok,Gunpbdok,BrDokS,NDokS,AlçıkS,DökFireSapma,FırFireSapma):-
```

```
write("Dunku Uretimde,"),nl,
```

```
BrutD(BrDokS,Gunfbdok,Gunpbdok),nl,NetD(NDokS,GunfDöka,GunpDöka),nl,
```

```
DoFire(DökFireSapma,Dök,FDökFire),nl,FFır(FırFireSapma,Fırın,FFırınFire).
```

```
BrutD(BrDokS,Gunfbdok,Gunpbdok):-BrDokS>0, nl,
```

```
write(" Planlanan brüt döküm adedi "),write(Gunpbdok),write(" iken"),
```

```
write(Gunfbdok), write(" adet döküm yapılmıştır."),nl,
```

```
write(" Sapma oranı="),write(BrDokS);true.
```

```
NetD(NDokS,GunfDöka,GunpDöka):-NDokS>0,
```

```
write(" Planlanan net döküm adedi "),write(GunpDöka),write(" iken"),
```

```
write(GunfDöka),write(" adet net döküm elde edilmiştir."),nl,
```

```
write(" Sapma oranı="),write(NDokS);true.
```

Ek.3. (devam)

```
DoFire(DökFireSapma,Dök,FDökFire):-DökFireSapma>0,
write("    Döküm fire beklenenden fazla olmuştur."),nl,write("
Planlanan döküm "),nl,
write("    "),write(" fire değeri "),write(Dök),write(" olarak
belirlenmişti. Oysa fiili"),nl,
write("    "),write(" döküm fire değeri
"),write(FDökFire),write("olmuştur."),nl,
write("    Sapma oranı="),write(DökFireSapma);true.
```

```
FFır(FırFireSapma,Fırın,FFırınFire):-FırFireSapma>0,
write("    Bu kod için fırın fire planlanana aşmıştır"),write("
Planlanan fırın fire değeri"),nl,
write("    "),write(Fırın),write(" olarak belirlenmişti. Oysa fiili
fırın fire değeri "),
write(FFırınFire),write(" dir."),nl,
write("    Sapma oranı="),write(FırFireSapma);true.
```

```
islem1(1,Kodno,Renk,BrDokKS,NDokKS,AlçıKS,KDökFireFiili,FırınCıkSonFire):-
makewindow(1,2,3,"KUMULATIF DEGERLERDEKI SAPMALARIN
NEDENLERI",0,0,25,80),nl,nl,nl,
BrutDokK(BrDokKS),nl,nl,NetDokK(NDokKS),nl,nl,KumDokF(KDökFireFiili),nl,
nl,AlçıK(AlçıKS),
nl,nl,FırınCık(FırınCıkSonFire),paus.
```

```
BrutDokK(BrDokKS):-BrDokKS>0,
write("    Kumulatif brut döküm adedi planlananın gerisinde
kalmıştır."),nl,
write("    Sapma oranı="),write(BrDokKS);true.
```

```
NetDokK(NDokKS):-NDokKS>0,
write("    Kumulatif net döküm adedi planlananın gerisinde
kalmıştır."),nl,
write("    Sapma oranı="),write(NDokKS);true.
```

```
KumDokF(KDökFireFiili):-KDökFireFiili>0,
write("    Kumulatif döküm fire beklenenden fazla olmuştur. Net
Uretimin az"),nl,
write("    olmasının bir nedeni de budur."),nl,
write("    Sapma oranı="),write(KDökFireFiili);true.
```

```
AlçıK(AlçıKS):-AlçıKS>0,
write("    Kumulatif alçıkaliıp Uretimi beklenenden daha az
gerçekleşmiştir."),nl,
write("    Sapma oranı="),write(AlçıKS);true.
```

```
FırınCık(FırınCıkSonFire):-FırınCıkSonFire>0,
write("    Fırın fire planlanandan fazla olmuştur."),nl,
write("    Sapma oranı="),write(FırınCıkSonFire);true.
```

Ek.3. (devam)

```
islem3(3,Kodno,Renk,Fırın,FFırınFire,Dök,FDökFire,GunfDepa,GunpDepa,Gu  
nfDöka,GunpDöka,AlçıkF,GunpAlçıkA,Gunfbdok,Gunpbdok,BrDokS,NDokS,AlçıkS,Dök  
FireSapma,FırFireSapma):-
```

```
BrutD(BrDokS,Gunfbdok,Gunpbdok),nl,NetD(NDokS,GunfDöka,GunpDöka),nl,
```

```
DoFire(DökFireSapma,Dök,FDökFire),nl,FFır(FırFireSapma,Fırın,FFırınFire).
```

```
islem4(4,Kodno,Renk,GunpDöka,Gunpbdok,Gunfdöka,Gunfbdok,Dök,FDökFire,  
Fırın,FFırınFire,BrDS,NDS,PDF,DepKS,BrDokKS,BrDokS,AlçıkKS,DepS,NDokKS,NDok  
KS,AlçıkS,FırFireSapma,DökFireSapma,KDökFireFiili,FırınCıkSonFire):-
```

```
%DepoK(DepKS),
```

```
BrutDokK(BrDokKS),nl,nl,NetDokK(NDokKS),nl,nl,AlçıkK(AlçıkKS),nl,nl,
```

```
KumDokF(KDökFireFiili),nl,nl,nl,paus,
```

```
write(" Bugün depoya girişi yapılan mamullerdeki sapmayı bulmak  
için"),nl,
```

```
write(" dünkü üretim bilgileri değerlendirilmelidir. Çünkü bugün  
depoya"),nl,
```

```
write(" giren mamullerin üretimine dün başlamıştır. Bugünkü sonuçlar  
dün"),nl,
```

```
write(" dökülmeye başlanan ve sırlanan mamullere aittir."),nl,
```

```
write(" Elde edilen sonuçlar şöyledir:"),nl,paus,nl,
```

```
BrutD(BrDokS,Gunfbdok,Gunpbdok),nl,NetD(NDokS,GunfDöka,GunpDöka),nl,
```

```
DoFire(DökFireSapma,Dök,FDökFire),nl,FFır(FırFireSapma,Fırın,FFırınFire),
```

```
write(" Ayrıca bugün dökümüne başlanan ve yarın bitmiş ürün haline  
gelecek"),
```

```
write(" mamullerle ilgili bugünkü üretim değerlendirmesi de  
şöyledir:"),nl,nl,
```

```
Brud(BrDS),nl,nl,ND(NDS),nl,nl,FireBug(PDF),nl,nl,
```

```
write(" Bugünkü alçık kalıp üretimi eklenecek"),paus.
```

```
FireBug(PDF):-PDF>0,
```

```
write(" Bugün yapılan dökümlerde döküm fire planlanandan fazla  
olmuştur."),nl,
```

```
write(" Sapma oranı="),write(PDF).
```

```
Brud(BrDS):-BrDS>0,
```

```
write(" Bugün gerektiği kadar brut döküm yapılmamıştır."),nl,
```

```
write(" Sapma oranı="),write(BrDS).
```

```
ND(NDS):-NDS>0,
```

```
write(" Elde edilen net döküm adedi planlananın gerisinde  
kalmıştır."),nl,
```

```
write(" Sapma oranı="),write(NDS).
```

```
consult_gunkf1:-existfile("gunkf1.db"),!,consult("gunkf1.db",gunkf1).  
consult_gunkf1.
```

```
consult_gunkf2:-existfile("gunkf2.db"),!,consult("gunkf2.db",gunkf2).  
consult_gunkf2.
```

```
consult_gunf1:-existfile("gunf1.db"),!,consult("gunf1.db",gunf1).  
consult_gunf1.
```

```
consult_gunf2:-existfile("gunf2.db"),!,consult("gunf2.db",gunf2).  
consult_gunf2.
```


Ek.3. (devam)

```
consult_bugUn:-existfile("bugUn.dba"),!,consult("bugUn.dba",bugUn).  
consult_bugUn.
```

```
consult_gunkp1:- existfile("gunkp1.dba"),!,consult("gunkp1.dba",gunkp1).  
consult_gunkp1.
```

```
consult_gunp1:-existfile("gunp1.dba"),!,consult("gunp1.dba",gunp1).  
consult_gunp1.
```

Ek.3. (devam)

* Bu modül ana menüyü içermektedir. Menüler aracılığıyla aylık planlanan değerlerin girilip, günlük değerlere dönüştürülmesi ve daha sonra günlük fiili değerlerin girilmesi gibi işlemler bu modül tarafından yönetilmektedir. *

```
project "tez"
include "globtez.pro"

domains
gunkpkayıtt1=k(symbol,symbol, integer, integer, integer, integer, integer)
a2=k(symbol, integer, integer, integer, integer)
kayıt=k(integer, integer, integer, integer, integer)

predicates
iş(symbol, integer, integer, integer, integer, real)
consult_aylık1
consult_gunpl
gunpl
consult_eldg
consult_bugün
consult_gunf2
consult_gunf1
consult_gunkf1
consult_gunkf2
consult_gunkp1
consult_galçp
consult_plan
consult_sıra
consult_dokplan
consult_gunkpdök
consult_gunkpalç1
cıkark(symbol)
cıkark2(symbol, symbol)
kacıncı_1sgunu(kayıt)
işlem(symbol, symbol, integer, integer, integer, integer, integer)
Acıkla
Go(integer)
Başla(symbol)
%alç1(symbol, integer, integer, integer, integer, integer)
ilkişlemler
ikincişlemler
gunpl_secim(integer, integer, integer, integer)
hatırlatma

clauses
hatırlatma:-write("alçık kalıp günleri düzelecek, 3.hane ").
kacıncı_1sgunu(k(1,7,1,1,1)).
kacıncı_1sgunu(k(2,7,1,1,1)).
kacıncı_1sgunu(k(3,7,2,2,2)).
kacıncı_1sgunu(k(4,7,3,3,3)).
kacıncı_1sgunu(k(5,7,4,4,4)).
kacıncı_1sgunu(k(6,7,4,4,4)).
kacıncı_1sgunu(k(7,7,4,4,4)).
```

Ek.3. (devam)

```
kacinci_1sgunu(k(8,7,5,5,5)).
kacinci_1sgunu(k(9,7,6,6,6)).
kacinci_1sgunu(k(10,7,7,7,7)).
kacinci_1sgunu(k(11,7,8,8,8)).
kacinci_1sgunu(k(12,7,9,9,9)).
```

```
kacinci_1sgunu(k(13,7,9,9,9)).
kacinci_1sgunu(k(14,7,9,9,9)).
```

```
kacinci_1sgunu(k(15,7,10,10,10)).
kacinci_1sgunu(k(16,7,11,11,11)).
kacinci_1sgunu(k(17,7,12,12,12)).
kacinci_1sgunu(k(18,7,13,13,13)).
kacinci_1sgunu(k(19,7,14,14,14)).
```

```
kacinci_1sgunu(k(20,7,14,14,14)).
kacinci_1sgunu(k(21,7,14,14,14)).
```

```
kacinci_1sgunu(k(22,7,15,15,15)).
kacinci_1sgunu(k(23,7,16,16,16)).
kacinci_1sgunu(k(24,7,17,17,17)).
kacinci_1sgunu(k(25,7,18,18,18)).
kacinci_1sgunu(k(26,7,19,19,19)).
```

```
kacinci_1sgunu(k(27,7,19,19,19)).
kacinci_1sgunu(k(28,7,19,19,19)).
```

```
kacinci_1sgunu(k(29,7,20,20,20)).
kacinci_1sgunu(k(30,7,21,21,21)).
kacinci_1sgunu(k(31,7,22,22,22)).
```

menu:-

```
makewindow(1,2,3,"ANA MENU",5,5,19,70),nl,nl,nl,nl,nl,
write("      Aylık Veri Girişİ.....(1)"),nl,
write("      Planlanan GUnluk Değerlerin Hesabı..(2)"),nl,
write("      GUnluk Veri Girişİ.....(3)"),nl,
write("      Açıklamalar.....(4)"),nl,
write("      Çıkış.....(5)"),nl,nl,
write("      Seçiminizi Giriniz.....(
)"),nl,nl,cursor(11,51),
readint (Secım),removewindow,Go(Secım),menu.
```

```
Go(Secım):-Secım=1,islem_secım(1);
           Secım=2,islem_secım(2);
           Secım=3,GUnlukGirişler;
           Secım=4,Acıkla;
           Secım=5,exit.
```

```
Acıkla:-!,nl,nl,nl,
write("Seçiminize Yardımcı Açıklamalar:"),nl,nl,nl,
write("      1 seçeneğini yalnızca bir defa aylık periyodun başında
planlanan"),
write("      aylık değerleri girmek için kullanınız."),nl,nl,
write("      2 seçeneğini, 1 seçeneğinde girilen verileri kullanarak
gUnluk"),
write("      planlanan değerlerin hesabı için kullanmalısınız.
Sonuç olarak 2"),
```

Ek.3. (devam)

```
write("          seçeneği 1 seçeneğinden sonra kullanılmalıdır."),nl,
write(" Not: Her iki secenek de sadece birer kez kullanılmış olacaktır"),
pause,menu.
```

```
ikinciislemeler:-write("ikinci işlemlere
geldik"),pause,bugün(k(Gu1,Ay1,_)),
consult_galçp,kacıncı_1sgunu(k(Gu1,Ay1,Alç1,_,_)),
```

```
consult_gunkpalç1,!,gunpl_secim(3,Gu1,Ay1,Alç1),write("buradayım"),
pause,gunpl.
```

```
gunpl_secim(4,Gu1,Ay1,Dök):-write("gunpl_secim(4) e geldik"),pause,
dokplan(k(Kodno,Topd1,Topd2)),write("dokplandan kayıt
alındı"),
write(Kodno),
pause,iş(Kodno,Topd1,Topd2,Gu1,Ay1,Dök).
```

```
gunpl_secim(3,Gu1,Ay1,Alç1):-galçp(k(Kodno,Gunpalç1a,FarkTop)),
write("Kodno="),write(Kodno),pause,
Kodno<>"1111",GunpkAlç1=Gunpalç1a*Alç1,
assertz(gunkpalç1(k(Kodno,GunpkAlç1,Gu1,Ay1))),
fail,!.
```

```
gunpl_secim(3,Gu1,Ay1,Alç1):-galçp(k(Kodno,Gunpalç1a,FarkTop)),
write("Kodno="),write(Kodno),pause,
Kodno="1111",GunpkAlç1=10,
assertz(gunkpalç1(k(Kodno,GunpkAlç1,Gu1,Ay1))),
fail,!.
```

```
gunpl_secim(3,Gu1,Ay1,Alç1):-!,save("gunkpalç1.dba",gunkpalç1).
```

```
/*alç1(Kodno,Gunpalç1a,Alç1,FarkTop,Gu1,Ay1),
save("gunkpalç1.dba",gunkpalç1),gunplan_secim(4).*/
```

```
/*alç1(Kodno,Gunpalç1a,Alç1,FarkTop,Gu1,Ay1):-
Kodno<>"1111",
GunpkAlç1=Gunpalç1a*Alç1,
assertz(gunkpalç1(k(Kodno,GunpkAlç1,Gu1,Ay1)));!,
write("Kodno="),write(Kodno),pause,
assertz(gunkpalç1(k(Kodno,0,0,0))).*/
```

```
iş(Kodno,Topd1,Topd2,Gu1,Ay1,Dök):-Kodno<>"1111",
GunpkDök=Topd2*Dök,GunpbkDök=Topd1*Dök,
assertz(gunkpdök(k(Kodno,GunpbkDök,GunpkDök,Gu1,Ay1))),
save("gunkpdök.dba",gunkpdök),
retract(dokplan(k(Kodno,Topd1,Topd2))),
gunpl_secim(4,Gu1,Ay1,Dök);consult_gunkpdök,
assertz(gunkpdök(k(Kodno,1,1,Gu1,Ay1))),save("gunkpdök.dba",gunkpdök).
```

```
gunpl:-
bugün(k(Gu1,Ay1,_)),kacıncı_1sgunu(k(Gu1,Ay1,_),Dök,_),
consult_dokplan,gunpl_secim(4,Gu1,Ay1,Dök).
```

Ek.3. (devam)

ilkişlemler:-

```
gunp1(k(Kodno, Renk, GunpDepa, GunpSıra, Gunpbfğ, GunpDöka, Gunpbdok)),
write("gunp1 den alınan kod="), write(Kodno), pause, nl,
işlem(Kodno, Renk, GunpDepa, GunpSıra, Gunpbfğ, GunpDöka, Gunpbdok).
```

```
işlem(Kodno, Renk, GunpDepa, GunpSıra, Gunpbfğ, GunpDöka, Gunpbdok):-
Kodno<>"1111", consult_bugün,
bugün(k(Gu1, Ay1, _)), kacıncı_ısgunu(k(Gu1, Ay1, _, _, Kay1)),
GunpkDepa=GunpDepa*Kay1, GunpkSıra=GunpSıra*Kay1, Gunpkbfğ=Gunpbfğ*Kay1,
G=Gu1-1,
retract(gunkp1(k(Kodno, Renk, _, _, _, G, Ay1))),
Gunpk=GunpkDepa-GunpDepa,
assertz(gunkp1(k(Kodno, Renk, Gunpk, GunpkSıra, Gunpkbfğ, Gu1, Ay1))),
save("gunkp1.dba", gunkp1),
retract(gunp1(k(Kodno, Renk, _, _, _, _))),
ilkişlemler;consult_bugün, write("bugün consult edildi"), pause,
bugün(k(Gu1, Ay1, _)), kacıncı_ısgunu(k(Gu1, Ay1, _, _, Kay1)),
GunpkDepa=GunpDepa*Kay1, GunpkSıra=GunpSıra*Kay1, Gunpkbfğ=Gunpbfğ*Kay1,
G=Gu1-1,
retract(gunkp1(k(Kodno, Renk, _, _, _, G, Ay1))),
write("gunkp1 den değer alındı"), pause,
assertz(gunkp1(k(Kodno, Renk, GunpkDepa, GunpkSıra, Gunpkbfğ, Gu1, Ay1))),
save("gunkp1.dba", gunkp1),
retract(gunp1(k(Kodno, Renk, GunpDepa, GunpSıra, Gunpbfğ, GunpDöka, Gunpbdok)))
```

```
,
consult_bugün, write("şimdi ikinci işlemlere gidilecek"), pause,
ikinciışlemler.
```

```
pause:-nl, write("lütfen herhangi bir tuşa basınız..."), readchar (_, nl).
```

```
consult_aylık1:-existfile("aylık1.dba"), !, consult("aylık1.dba", aylık1).
consult_aylık1.
```

```
consult_gunkp1:-
    existfile("gunkp1.dba"), !,
    consult("gunkp1.dba", gunkp1).
consult_gunkp1.
```

```
consult_gunf1:-existfile("gunf1.dba"), !, consult("gunf1.dba", gunf1).
consult_gunf1.
```

```
consult_gunkpalçı:-
    existfile("gunkpalçı.dba"), !,
    consult("gunkpalçı.dba", gunkpalçı).
consult_gunkpalçı.
```

```
consult_gunkpdök:-
    existfile("gunkpdök.dba"), !,
    consult("gunkpdök.dba", gunkpdök).
consult_gunkpdök.
```

```
consult_gunp1:-existfile("gunp1.dba"), !, consult("gunp1.dba", gunp1).
consult_gunp1.
```

```
consult_eldg:-existfile("eldg.dba"), !, consult("eldg.dba", eldg).
consult_eldg.
```

```
consult_sıra:- existfile("sıra.dba"), !, consult("sıra.dba", sıra).
consult_sıra.
```

Ek.3. (devam)

```
consult_gunkf1:-existfile("gunkf1.dba"),!,consult("gunkf1.dba",gunkf1).
consult_gunkf1.

consult_gunkf2:-existfile("gunkf2.dba"),!,consult("gunkf2.dba",gunkf2).
consult_gunkf2.

consult_dokplan:-existfile("dokplan.dba"),!,consult("dokplan.dba",dokplan
).
consult_dokplan
.
consult_gunf2:-existfile("gunf2.dba"),!,consult("gunf2.dba",gunf2).
consult_gunf2.

consult_bugUn:-existfile("bugUn.dba"),!,consult("bugUn.dba",bugUn).
consult_bugUn.

consult_plan:-
    existfile("plan.dba"),!,
    consult("plan.dba",plan).
consult_plan.

consult_galçp:- existfile("galçp.dba"),!,consult("galçp.dba",galçp).
consult_galçp.

cıkar(h).
cıkar(e):-makewindow(1,2,3,"",2,2,20,78),
    cursor(8,20),
    write("Uretimi biten mamulun:"),cursor(10,30),
    write("Kod numarasını giriniz ...:"),cursor(11,30),
    write("Rengini giriniz .....:"),cursor(10,57),
    readln(Kodno),cursor(11,57),readln(Renk),
    consult_gunkp1,consult_gunp1,
    retract(gunkp1(k(Kodno,Renk,_,_,_,_,_))),
    save("gunkp1.dba",gunkp1),
    write("gunkp1"),
    retract(gunp1(k(Kodno,Renk,_,_,_,_,_))),
    save("gunp1.dba",gunp1),write("gunp1"),cıkar2(Kodno,Renk).

cıkar2(Kodno,Renk):-write(Kodno,Renk),
    consult_aylık1,write("aylık1 consult edildi"),
    retract(aylık1(k(Kodno,Renk,_,_,_,_,_))),
    save("aylık1.dba",aylık1),write("aylık1"),
    consult_gunkf1,
    retract(gunkf1(k(Kodno,Renk,_,_,_,_,_))),
    write("gunkf1"),save("gunkf1.dba",gunkf1),
    consult_gunf1,write("gunf1 consult edildi"),
    retract(gunf1(k(Kodno,Renk,_,_,_,_,_))),
    write("gunf1"),
    save("gunf1.dba",gunf1),
    consult_eldg,write("eldg consult edildi"),

retract(eldg(k(Kodno,Renk,_))),save("eldg.dba",eldg),write("eldg").

Başla(e):- makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
    cursor(8,15), write("0 halde üretimi izlemeye başlayabilirsiniz..."),
    cursor(10,15),write("Bir önceki gün üretimi tamamlanan kod var mı? e/h (
)"),
```

Ek. 3. (devam)

```
cursor(10,66),readln(Cev),cursor(10,67),write(")"),cıkar(Cev),
removewindow,makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(12,15),write("Gerekli işlemler yapılıyor, lütfen bekleyiniz..."),
consult_gunkpl,consult_gunpl,write("gerekli consultler yapıldı"),pause,
ilkişlemler, cursor(14,15),
write("şimdi gelecek menüyü kullanarak günlük veri girişlerini
yapınız."),
cursor(15,15), write("Daha sonra bu programdan çıkıp tez2.exe adlı
programı çalıştırınız."),
pause,removewindow,GünlükGirişler.
```

```
Başla(h):- makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(5,14),write("Planlama döneminizin başlangıç tarihini giriniz"),
cursor(7,9),write("(Örn. 5 Şubat salı için Gun1:5 Ay:2 Gun2:salı
şeklinde)"),
cursor(10,20),write("Gun1 :"),cursor(10,32),write("Ay :"),cursor(10,44),
write("Gun2 :"),cursor(10,26),readint(Gu1),cursor(10,32),
write("Ay :"),cursor(10,44), write("Gun2 :"), cursor(10,36),readint(Ay1),
cursor(10,44), write("Gun2 :"), cursor(10,50),readln(Gun),
asserta(plan(k(Gu1,Ay1,Gun))), save("plan.dba",plan),removewindow,
makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(5,14),write("Lütfen önce sırasıyla aşağıdaki işlemleri yapınız:"),
cursor(6,14),write("Bunun için şimdi gelecek menüden 1 ve 2 seçeneklerini
sırayla çalıştırmalısınız."),
cursor(10,8),write("- Aylık veri girişi,"),
cursor(11,8),write("- Aylık bilgilerden günlük planlanmış değerlerin elde
edilmesi."),
cursor(13,8),write("Bu işlemler bittikten sonra programı yeniden
çalıştırınız."),nl,nl,
pause,removewindow,menu.
```

goal

```
makewindow(1,2,3,"GÜNLÜK ÜRETİMİ İZLEME PROGRAMI",2,2,20,78),

cursor(7,10),write("Bugünün tarihini giriniz.....:"),
cursor(8,14),write("(Örnek:5 Şubat için Gun=5,Ay=2 şeklinde)"),
cursor(11,14),write("Gun.....:"),cursor(11,33),write("Ay.....:
"),
cursor(13,20),write("Hangi gundesiniz ?..."),
cursor(11,28),readint(Gu1),cursor(11,33),write("Ay.....:"),
cursor(11,46),readint(Ay1),cursor(13,42),readln(Gun),
consult_bugun,
asserta(bugun(k(Gu1,Ay1,Gun))),save("bugun.dba",bugun),
makewindow(1,2,3,"Açıklama Penceresi",2,2,20,78),
cursor(6,10), write("Bu programı doğru bir şekilde kullanabilmeniz
için;"),
cursor(7,10), write("öncelikle aşağıdaki iki adımın sırayla yapılması
gerekmektedir:" ),
cursor(9,14),write("- Aylık planlanan değerlerin
girilmesi,"),cursor(10,14),
write("- Bu değerlerin günlük planlara dönüştürülmesi,"),
cursor(16,6),write("ÖNEMLİ NOT:Yukarıdaki iki adımın planlama döneminin
en başında,"),
```

Ek.3. (devam)

```
cursor(17,17),write("yalnızca bir kez alıřtırılması gerekmektedir."),  
cursor(12,14), write("Bu iřlemler daha nce yapıldı mı?(e/h) ( )"),  
cursor(12,54), readln(Cev), removewindow,Başla(Cev).
```


Ek.3. (devam)

* Bu modul, diğer modüllerde ortak olarak kullanılan değişken ve veri tabanlarına ilişkin tanımlamaların yapıldığı modüldür. *

Global Domains

```
ay,gun,süre,sıran=integer
so=real
name,sayı=symbol
ay1=k(symbol,symbol,integer,integer,integer,integer,integer,integer,integer)
ay2=k(symbol,integer,integer,integer)
gpkayıt1=k(symbol,symbol,integer,integer,integer,integer,integer)
gpkayıt2=k(symbol,integer,integer)
gunsayac=k(integer,integer,integer)
gfkayıt1=k(symbol,symbol,integer,integer,integer,integer,integer)
gfkayıt2=k(symbol,integer,integer,integer,integer,integer,integer,integer)
gfkayıt3=k(symbol,integer,integer,integer,integer)
gkfkayıt1=k(symbol,symbol,integer,integer,integer,integer,integer)
gkfkayıt2=k(symbol,integer,integer,integer,integer,integer,integer)
gdokkayıt=k(symbol,symbol,integer,integer)
gdokkay=k(symbol,symbol,integer,integer)
eldgkayıt=k(symbol,symbol,integer)
gdoktop=k(symbol,integer,integer)
isgkayıt=k(integer,integer,integer,integer,integer,integer,integer)
tar=k(integer,integer,symbol)
kayalçı=k(symbol,integer)
kalçı=k(symbol,integer,integer,integer)
file=myfile
```

Global Predicates

```
nondeterm Pfireler(symbol,real,real,real)-(i,o,o,o),(o,o,o,o)
nondeterm menu
nondeterm GUnlukGirişler
consult_aylık2
consult_dokplan
consult_gunf3
pause
nondeterm Analş(integer,integer,integer)-(i,i,i)
nondeterm islem_secim(integer)-(i)
nondeterm gunplan_secim(integer)-(i)
nondeterm uy(integer,symbol,symbol)-(i,i,i)
Yenıgun(symbol,symbol)-(i,o)
Gunsay(integer,integer)-(i,o)
nondeterm
islem3(integer,symbol,symbol,real,real,real,real,real,real,real)-(i,i,i,i,i,i,i,i,i,i)
nondeterm islem2(integer,symbol,symbol,real,real,real,real,real,real,real,real)-(i,i,i,i,i,i,i,i,i,i)
nondeterm islem1(integer,symbol,symbol,real,real,real,real,real,real,real,real)-(i,i,i,i,i,i,i,i,i,i)
nondeterm islem4(integer,symbol,symbol,real,real,real,real,real,real,real,real,real,real,real,real,real,real)-(i,i,i,i,i,i,i,i,i,i,i,i,i,i,i)
nondeterm GUnlukFiiliVeriGirişı(integer,integer,integer)-(i,i,i)
```

Ek.3. (devam)

nondeterm SıraNo(integer, integer, integer)-(i, i, o)
Hangıgun(symbol, integer, integer, integer, integer, integer) -(i, o, o, o, o, o)
Elg(integer, integer, integer, integer, symbol, symbol, integer)
-(i, i, i, i, i, i, o)
Eg(integer, integer, integer, integer, symbol, symbol, integer)
-(i, i, i, i, i, i, o)
egunalçı(integer, integer, integer, integer, symbol, integer) -(i, i, i, i, i, o)

Global Database

database-aylık1
aylık1(ay1)

database-aylık2
aylık2(ay2)

database-gunp1
gunp1(gpkayıt1)

database-gunkf1
gunkf1(gfkayıt1)

database-gunkf2
gunkf2(gfkayıt2)

database-gdok
gdok(gdokkayıt)

database-dokplan
dokplan(gdoktop)

database-eldg
eldg(eldgkayıt)

database-galçp
galçp(gkpkayıt2)

database-bugün
bugün(tar)

database-plan
plan(tar)

database-egalçı
egalçı(kayalçı)

database-gunf1
gunf1(gfkayıt1)

database-gunf2
gunf2(gfkayıt2)

database-gunf3
gunf3(gfkayıt3)

database-gunkp1
gunkp1(gfkayıt1)

database-gunkpdök
gunkpdök(gfkayıt3)

database-gunkpalçı
gunkpalçı(kalçı)

Ek.3. (devam)

database-sıra
sıra(gdokkay)

* Bu programda modüler programlamaya göre, tek bir modül gibi çalışacak olan programlar tanımlanmaktadır. *

tez1+tez2+tez3+tezamac+

tez4+tez5+